# Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data (SensorKDD'09)

**Held in conjunction with**

**June 28, 2009**
**Paris, France**

## Workshop Chairs


| |
|---|
| Dr. Olufemi A. Omitaomu, Oak Ridge National Laboratory, TN, USA |
| Dr. Auroop R. Ganguly, Oak Ridge National Laboratory, TN, USA |
| Prof. Joao Gama, University of Porto, Portugal |
| Dr. Ranga Raju Vatsavai, Oak Ridge National Laboratory, TN, USA |
| Prof. Nitesh V. Chawla, University of Notre Dame, IN, USA |
| Prof. Mohamed Medhat Gaber, Monash University, Australia |

# Third International Workshop on Knowledge Discovery from Sensor Data (SensorKDD'09)

Wide-area sensor infrastructures, remote sensors, RFIDs, and wireless sensor networks yield massive volumes of disparate, dynamic, and geographically distributed data. As such sensors are becoming ubiquitous, a set of broad requirements is beginning to emerge across high-priority applications including adaptability to climate change, electric grid monitoring, disaster preparedness and management, national or homeland security, and the management of critical infrastructures. The raw data from sensors need to be efficiently managed and transformed to usable information through data fusion, which in turn must be converted to predictive insights via knowledge discovery, ultimately facilitating automated or human-induced tactical decisions or strategic policy based on decision sciences and decision support systems.

The challenges for the knowledge discovery community are expected to be immense. On the one hand, dynamic data streams or events require real-time analysis methodologies and systems, while on the other hand centralized processing through high end computing is also required for generating offline predictive insights, which in turn can facilitate real-time analysis. The online and real-time knowledge discovery imply immediate opportunities as well as intriguing short- and long-term challenges for practitioners and researchers in knowledge discovery. The opportunities would be to develop new data mining approaches and adapt traditional and emerging knowledge discovery methodologies to the requirements of the emerging problems. In addition, emerging societal problems require knowledge discovery solutions that are designed to investigate anomalies, hotspots, changes, extremes and nonlinear processes, and departures from the normal. The theme for the 2009 SensorKDD workshop is around three inter-related global priorities: *climate change, energy assurance, and infrastructural impacts*. While authors were strongly encouraged to submit all papers which may be of potential interest to the SensorKDD community, we also solicited submissions from authors who may be new to this community but have an interest in contributing to this year's theme.

The workshop brings together researchers from academia, government, and the industry working in the following areas and applications:

1. Offline Knowledge Discovery
    a. Predictive analysis from geographically distributed and heterogeneous data
    b. Computationally efficient approaches for mining unusual patterns, specifically, anomalies, extremes, nonlinear processes and change, from massive and disparate space-time data
2. Online Knowledge Discovery
    a. Real-time analysis of dynamic and distributed data, including streaming and event-based data
    b. Mining from continuous streams of time-changing data and mining from ubiquitous data
    c. Efficient algorithms to detect deviations from the normal in real-time
    d. Resource-aware algorithms for distributed mining
3. Decision and Policy Aids
    a. Coordinated offline discovery and online analysis with feedback loops
    b. Combination of knowledge discovery and decision scientific processes
    c. Facilitation of faster and reliable tactical decisions as well as prudent and insightful longer term policies
4. Theory

   a. Distributed data stream models
   b. Theoretical frameworks for distributed stream mining
5. Case Studies
   a. Success stories in national or global priority applications
   b. Real-world problem design and knowledge discovery requirements

## Motivation

The expected ubiquity of sensors in the near future, combined with the critical roles they are expected to play in high priority application solutions, point to an era of unprecedented growth and opportunities. The requirements described earlier imply immediate opportunities as well as intriguing short- and long-term challenges for practitioners and researchers in knowledge discovery. In addition, the knowledge discovery and data mining (KDD) community would be called upon, again and again, as partners with domain experts to solve critical application solutions in business and government, as well as in the domain sciences and engineering.

The main motivation for this workshop stems from the increasing need for a forum to exchange ideas and recent research results, and to facilitate collaboration and dialog between academia, government, and industrial stakeholders. Based on the positive feedback from the previous workshop attendees and our own experiences and interactions with the government agencies such as the United States Department of Energy, United States Department of Homeland Security, United States Department of Defense, and involvement with numerous projects on knowledge discovery from sensor data, we strongly believe in the continuation of this workshop. We also believe that the ACM SIGKDD conference is the right forum to organize this workshop as it brings the KDD community together in this important area to establish a much needed leadership position in research and practice in the near term, as well as in the long term.

## Success of Previous SensorKDD Workshops

The previous two workshops – SensorKDD'07 and SensorKDD'08 – held in conjunction with the 13th and 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining respectively attracted several participants as well as many high quality papers and presentations. The 2007 workshop was attended by more than seventy registered participants. The workshop program includes presentations by authors of six accepted full papers and four invited speakers. The invited speakers were Prof. Pedro Domingos of the University of Washington, Prof. Joydeep Ghosh of the University of Texas, Austin, Prof. Hillol Kargupta of the University of Maryland, Baltimore County, and Dr. Brian Worley of the Oak Ridge National Laboratory (ORNL). There were also poster presentations by authors of six accepted short papers. The extended versions of papers presented at the workshop were developed into a book titled "Knowledge Discovery from Sensor Data," the first book published in this specific discipline. The four top accepted papers were awarded a cash prize of $500 each. The prize money was donated by the Computational Sciences and Engineering Division (CSED) of the Oak Ridge National Laboratory and Information Society Technology Project KDUbiq-WG3 of the European Union. The top papers were also published in a special issue of the Journal of Intelligent Data Analysis. This workshop was partially sponsored by the Geographic Information Science and Technology Group of CSED at ORNL.

The 2008 workshop was attended by more than 60 registered participants. There were presentations by authors of seven accepted full papers and six accepted short papers; the workshop program also include presentations by two invited speakers – Prof. Jiawei Han of the University of Illinois at Urbana-Champaign and Dr. Kendra Moore of the Defense Advanced Research Projects Agency. The extended versions of papers presented at the 2008 workshop are scheduled for publication as Springer's LNCS post-proceedings in 2009. The two top accepted papers were awarded a cash prize of $500 each; the prizes were donated by the Computational Sciences and Engineering Division of the Oak Ridge National Laboratory. This workshop was partially sponsored by the Geographic Information Science and Technology Group of CSED at ORNL.

## SensorKDD'09 Cup

This year we organize the first "SensorKDD Cup", similar in spirit to the KDD Cup, but in line with our 2009 workshop theme. We challenge contestants with intriguing problems along with geographically distributed and dynamic data from sensors and model simulations. The datasets were provided or compiled by leveraging ORNL resources.

## Workshop Sponsors

The SensorKDD'09 workshop is sponsored by the Geographic Information Science and Technology (GIST) Group at Oak Ridge National Laboratory, the Computational Sciences and Engineering (CSE) Division at the Oak Ridge National Laboratory, and Cooperating Objects Network of Excellence (CONET).

## Appreciation

We would like to thank our sponsors for their kind donations. In addition, we thank the SIGKDD'09 organizers, the authors of the submitted papers, the invited speakers, and the members of the Program Committee for their respective and collective efforts to make this workshop possible.

The workshop proceedings were compiled by Dr. Olufemi A. Omitaomu of the Computational Sciences and Engineering Division at Oak Ridge National Laboratory. The workshop proceedings have been co-authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains, and the publisher by accepting the article for publication, acknowledges that the United States Government retains, a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Dr. Olufemi A. Omitaomu (omitaomuoa@ornl.gov; +1-865-241-4310)
Dr. Auroop R. Ganguly (gangulyar@ornl.gov; +1-865-241-1305)
Dr. Ranga Raju Vatsavai (vatsavairr@ornl.gov; +1-865-576-3569)
Oak Ridge National Laboratory, 1 Bethel Valley Rd, MS-6017, Oak Ridge, TN, USA.

Prof. Joao Gama (jgama@fep.up.pt; +351-22-339-2094)
LIAAD-INESC Porto LA, University of Porto, Rua de Ceuta, Porto, Portugal.

Prof. Nitesh V. Chawla (nchawla@cse.nd.edu; +1-574-631-8716)
University of Notre Dame, Dept. of Computer Science & Engineering, IN, USA.

Prof. Mohamed Medhat Gaber (mohamed.gaber@infotech.monash.edu.au; +61-39-903-2332)
Monash University, 900 Dandenong Road, Melbourne, Victoria, 3145, Australia

# Organizers

## Workshop Chairs:

1. **Olufemi A. Omitaomu,** Oak Ridge National Laboratory, TN, USA.
2. **Auroop R. Ganguly,** Oak Ridge National Laboratory, TN, USA.
3. **Joao Gama,** University of Porto, Portugal.
4. **Ranga Raju Vatsavai**, Oak Ridge National Laboratory, TN, USA.
5. **Nitesh V. Chawla**, University of Notre Dame, IN, USA.
6. **Mohamed Medhat Gaber,** Monash University, Australia.

## Program Committee (In alphabetical order of last name):

1. **Adedeji B. Badiru**, Air Force Institute of Technology, Dayton, OH, USA.
2. **Budhendra L. Bhaduri**, Oak Ridge National Laboratory, TN, USA.
3. **Eric Auriol**, CLIMPACT, Paris, France.
4. **Albert Bifet**, University Polytechnica, Catalunya, Spain.
5. **Michaela Black**, University of Ulster, Coleraine, Northern Ireland, UK.
6. **Jose del Campo-Avila**, Universidad de Malaga, Spain.
7. **Andre Carvalho**, University of Sao Paulo, Brazil.
8. **Sanjay Chawla**, University of Sydney, Australia.
9. **Diane Cook**, Washington State University, Pullman, WA, USA.
10. **Alfredo Cuzzocrea**, University of Calabria, Italy.
11. **Christie Ezeife**, University of Windsor, Canada.
12. **David J. Erickson III**, Oak Ridge National Laboratory, TN, USA.
13. **Yi Fang**, Purdue University, West Lafayette, IN, USA.
14. **Francisco Ferrer**, University of Seville, Spain.
15. **James H. Garrett**, Carnegie Mellon University, Pittsburgh, PA, USA.
16. **Joydeep Ghosh**, University of Texas, Austin, TX, USA.
17. **Bryan L. Gorman**, Oak Ridge National Laboratory, TN, USA.
18. **Sara Graves**, University of Alabama, Huntsville, AL, USA.
19. **Ray Hickey**, University of Ulster, Coleraine, Northern Ireland, UK.
20. **Forrest Hoffman**, Oak Ridge National Laboratory, TN, USA.
21. **Luke (Jun) Huan**, University of Kansas, Lawrence, KS, USA.
22. **Volkan Isler**, University of Minnesota, Minneapolis, MN, USA.
23. **Vandana Janeja**, University of Maryland, Baltimore County, MD, USA.
24. **Yu (Cathy) Jiao**, Oak Ridge National Laboratory, TN, USA.
25. **Ralf Klinkenberg**, University of Dortmund, Germany.
26. **Miroslav Kubat**, University Miami, FL, USA.
27. **Vipin Kumar**, University of Minnesota, Minneapolis, MN, USA.
28. **Mark Last**, Ben-Gurion University, Israel.
29. **Chang-Tien Lu**, Virginia Tech., VA, USA.
30. **Elaine Parros Machado de Sousa**, University of Sao Paulo, Brazil.
31. **Sameep Mehta**, IBM Research, India.
32. **Laurent Mignet**, IBM Research, India.
33. **S. Muthu Muthukrishnan**, Rutgers University and AT&T Research, NJ, USA.
34. **George Ostrouchov**, Oak Ridge National Laboratory, TN, USA.

35. **Rahul Ramachandran**, University of Alabama, Huntsville, AL, USA.
36. **Pedro Rodrigues**, University of Porto, Portugal.
37. **Josep Roure**, Carnegie Mellon University, Pittsburgh, PA, USA.
38. **Bernhard Seeger**, University Marburg, Germany.
39. **Cyrus Shahabi**, University of Southern California, USA.
40. **Shashi Shekhar**, University of Minnesota, Minneapolis, MN, USA.
41. **Mallikarjun Shankar**, Oak Ridge National Laboratory, Oak Ridge, TN, USA.
42. **Lucio Soibelman**, Carnegie Mellon University, Pittsburgh, PA, USA.
43. **Alexandre Sorokine**, Oak Ridge National Laboratory, Oak Ridge, TN, USA.
44. **Eduardo J. Spinosa**, University of Sao Paulo, Brazil.
45. **Karsten Steinhaeuser**, University of Notre Dame, IN and Oak Ridge National Laboratory, TN, USA.
46. **Peng Xu**, ExxonMobil Corporate Strategic Research, NJ, USA.
47. **Eiko Yoneki**, University of Cambridge, UK.
48. **Philip S. Yu**, IBM Watson Research Center, Yorktown Heights, NY, USA.
49. **Nithya Vijayakumar**, Cisco Systems, Inc., USA.
50. **Guangzhi Qu**, Oakland University, Rochester, MI, USA.

Workshop website: http://www.ornl.gov/sci/knowledgediscovery/SensorKDD-2009/

# Table of Contents


## Invited Talks …….. 9

A Data Modeling Approach to Climate Change Attribution
*Aurélie Lozano*, IBM T.J. Watson Research Center ...…….. 9

Space Missions and Sensor Networking: Challenging Scenarios
*Alessandro Donati*, European Space Agency, Darmstadt, Germany …… 10

How Optimized Environmental Sensing Helps Address Information Overload on the Web
*Carlos Guestrin*, Carnegie Mellon University ……. 11

## Full Research Papers …….…… . 12

Handling Outliers and Concept Drift in Online Mass Flow Prediction in CFB Boilers
*J. Bakker, M. Pechenizkiy, I. Zliobaite, A. Ivannikov, and T. Karkkainen* …… 13

An Exploration of Climate Data Using Complex Networks
*Karsten Steinhaeuser, Nitesh V. Chawla, and Auroop R. Ganguly*……... 23

A Comparison of SNOTEL and AMSR-E Snow Water Equivalent Datasets in Western U.S. Watersheds
*Cody L. Moser, Oubeidillah Aziz, Glenn A. Tootle, Venkat Lakshmi, and Greg Kerr* …… 32

EDISKCO: Energy Efficient Distributed in-Sensor-Network K-center Clustering with Outliers
*Marwan Hassani, Emmanuel Muller, and Thomas Seidl* …….. 39

Phenological Event Detection from Multitemporal Image Data
*Ranga Raju Vatsavai* …….. 49

Mining in a Mobile Environment
*Sean McRoskey, James Notwell, Nitesh V. Chawla, and Christian Poellabauer* …… 56

## Short Research Papers …….…… .. 61

On the Identification of Intra-seasonal Changes in the Indian Summer Monsoon
*Shivam Tripathi and Rao S. Govindaraju* ……. 62

Reduction of Ground-Based Sensor Sites for Spatio-Temporal Analysis of Aerosols
*Vladan Radosavljevic, Slobodan Vucetic, and Zoran Obradovic* ……. 71

OcVFDT: One-class Very Fast Decision Tree for One-class Classification of Data Streams
*Chen Li, Yang Zhang, and Xue Li* …… 79

A Frequent Pattern Based Framework for Event Detection in Sensor Network Stream Data
*Li Wan, Jianxin Liao, and Xiaomin Zhu* …… 87

Supervised Clustering via Principal Component Analysis in a Retrieval Application
*Esteban Garcia-Cuesta, Ines M. Galvan, and Antonio J. de Castro* ……. 97

A Novel Measure for Validating Clustering Results Applied to Road Traffic
*Yosr Naija and Kaouther Blibech Sinaoui* …… 105

SkyTree: Scalable Skyline Computation for Sensor Data
*Jongwuk Lee and Seung-won Hwang* ……... 114

Clustering of Power Quality Event Data Collected via Monitoring Systems Installed on the Electricity Network
*Mennan Guder, Nihan Kesim Cicekli, Ozgul Salor, and Isik Cadirci* …….. 124

**Challenge Papers ……... 131**

SensorKDD'09 Cup …… 132

Change Detection in Rainfall and Temperature Patterns over India
*Shivam Tripathi and Rao S. Govindaraju* …… 133

Anomaly Detection and Spatio-Temporal Analysis of Global Climate System
*Mahashweta Das and Srinivasan Parthasarathy* …… 142

# INVITED TALKS

## A Data Modeling Approach to Climate Change Attribution

**Aurélie Lozano**
Research Staff Member
IBM T.J. Watson Research Center

**Abstract:**
Climate modeling and analysis of climate change have largely been based on forward simulation with physical models. We propose here a data centric approach to climate study based solely on the actual observed data. This novel approach utilizes a variety of relevant statistical modeling and machine learning techniques such as spatial-temporal causal modeling and extreme value modeling, and suggests multiple future research directions. We will describe preliminary results using data for North America from CRU, NOAA, NASA, NCDC, and CDIAC, as well as certain technical challenges encountered. It is hoped that this alternative perspective will help uncover new insights, improve aspects of simulation models with known uncertainties, and provide a useful complementary approach to climate study.

**Bio-Sketch of the Speaker:**
Aurélie Lozano is a Research Staff Member in the Business Analytics and Mathematical Sciences department at the IBM T.J. Watson Research Center. Her research interests are in machine learning, data mining, and their applications. She received a Ph.D. in Electrical Engineering from Princeton University, where her research focused on statistical learning and wireless sensor networks.

# Space Missions & Sensor Networking: Challenging Scenarios

**Alessandro Donati**
Head of Advanced Mission Concepts and Technologies Office
Human Spaceflight and Exploration Operations Department
European Space Agency - Darmstadt, Germany

**Abstract:**
Sensor networking is a paradigm getting familiar in space missions and services. The talk will provide a panoramic view of examples of challenging missions related to earth environment and to space exploration, where sensor knowledge discovery techniques might become instrumental to fulfill mission objectives. European Space Agency (ESA) missions such as GMES (Global Monitoring for Environment and Security) and the series of possible Mars exploration missions will be presented and put in context with the topic of the workshop.

**Bio-Sketch of the Speaker:**
Alessandro Donati is the lead of the Advanced Mission Concepts and Technology Office at the European Space Operations Centre of the European Space Agency, ESA. His team's interests are artificial intelligence technology applied to mission operations processes such as planning & scheduling, monitoring and diagnosis, resource management. He received an electronic engineering degree from La Sapienza University of Rome.

# How Optimized Environmental Sensing Helps Address Information Overload on the Web

**Carlos Guestrin**
Assistant Professor
Machine Learning and Computer Science Departments
Carnegie Mellon University

**Abstract:**
In this talk, we tackle a fundamental problem that arises when using sensors to monitor the ecological condition of rivers and lakes, the network of pipes that bring water to our taps, or the activities of an elderly individual when sitting on a chair: Where should we place the sensors in order to make effective and robust predictions?

Such sensing problems are typically NP-hard, and in the past, heuristics without theoretical guarantees about the solution quality have often been used. In this talk, we present algorithms which efficiently find provably near-optimal solutions to large, complex sensing problems. Our algorithms are based on the key insight that many important sensing problems exhibit submodularity, an intuitive diminishing returns property: Adding a sensor helps more the fewer sensors we have placed so far. In addition to identifying most informative locations for placing sensors, our algorithms can handle settings, where sensor nodes need to be able to reliably communicate over lossy links, where mobile robots are used for collecting data or where solutions need to be robust against adversaries and sensor failures.

We present results applying our algorithms to several real-world sensing tasks, including environmental monitoring using robotic sensors, activity recognition using a built sensing chair, and a sensor placement competition. We conclude with drawing an interesting connection between sensor placement for water monitoring and addressing the challenges of information overload on the web. As examples of this connection, we address the problem of selecting blogs to read in order to learn about the biggest stories discussed on the web, and personalizing content to turn down the noise in the blogosphere.

**Bio-Sketch of the Speaker:**
Carlos Guestrin is the Finmeccanica Assistant Professor in the Machine Learning and in the Computer Science Departments at Carnegie Mellon University. Previously, he was a senior researcher at the Intel Research Lab in Berkeley. Carlos received his PhD in Computer Science from Stanford University. Carlos' work received awards at a number of conferences and a journal: KDD 2007, IPSN 2005 and 2006, VLDB 2004, NIPS 2003 and 2007, UAI 2005, ICML 2005, JAIR in 2007, and JWRPM in 2009. He is also a recipient of the ONR Young Investigator Award, NSF Career Award, Alfred P. Sloan Fellowship, and IBM Faculty Fellowship. Carlos was named one of the 2008 "Brilliant 10" by Popular Science Magazine and received the IJCAI Computers and Thought Award. He is currently a member of the Information Sciences and Technology (ISAT) advisory group for DARPA.

# FULL RESEARCH PAPERS

# Handling Outliers and Concept Drift in Online Mass Flow Prediction in CFB Boilers

J. Bakker, M. Pechenizkiy, I. Žliobaitė
Dept. of CS, TU Eindhoven
P.O. Box 513, NL-5600MB
the Netherlands
{j.bakker,m.pechenizkiy,i.zliobaite}@tue.nl

A. Ivannikov, T. Kärkkäinen
Dept. of MIT, U. Jyväskylä
P.O. Box 35, FIN-40014
Finland
aivanni@cc.jyu.fi,tka@mit.jyu.fi

## ABSTRACT

In this paper we consider an application of data mining technology to the analysis of time series data from a pilot circulating fluidized bed (CFB) reactor. We focus on the problem of the online mass prediction in CFB boilers. We present a framework based on switching regression models depending on perceived changes in the data. We analyze three alternatives for change detection. Additionally, a noise canceling and a state determination and windowing mechanisms are used for improving the robustness of online prediction. We validate our ideas on real data collected from the pilot CFB boiler.

## Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database applications—*Data mining*

## General Terms

Experimentation, Performance, Reliability

## 1. INTRODUCTION

Continuous and growing increase of fluctuations in electricity consumption brings new challenges for the control systems of boilers. Conventional power generation will face high demands to ensure the security of energy supply because of increasing share of renewable energy sources like wind and solar power in power production. This can lead to frequent load changes which call for novel control concepts in order to minimize emissions and to sustain high efficiency during load changes.

From combustion point of view the main challenges for the existing boilers are caused by a wider fuel selection, increasing share of low quality and bio fuels, and co-combustion. In steady operation, combustion is affected by the disturbances in the feed-rate of the fuel and by the incomplete mixing of the fuel in the bed, which may cause changes in the burning rate, oxygen level and increase CO emissions. This is especially important, when considering the new biomass based fuels, which have increasingly been used to replace coal. These new biofuels are often rather inhomogeneous, which can cause instabilities in the feeding. These fuels are usually also very reactive. Biomass fuels have much higher reactivity compared to coals and the knowledge of the factors affecting the combustion dynamics is important for optimum control. The knowledge of the dynamics of combustion is also important for optimizing load changes [11].

Data-mining approaches can be used to develop better understanding of underlying processes in CFB boiler, or learn a model for optimizing its efficiency [9]. For instance, the selection of process variables to optimize combustion efficiency has been studied in [12]. One typical problem, which we address in this work, is an online prediction of the system parameters. These are required for the control optimization but cannot be measured reliably in a direct way. Mass flow is one of such parameters. Currently, it is calculated offline using relatively long time period averages. Recently in [5] a method for learning regression models on mass flow prediction in CFB boilers was introduced. That approach assumed known points in time where a state transition would occur (e.g. changing from replenishment to consume state).

In this paper we propose a framework for online mass flow prediction that is based on learning and switching regression models, where the switching is regulated by an online change detection mechanism.

The paper is organized as follows. Section 2 presents a wider context of sensor data mining for developing better understanding and control of CFB reactors, and particularly the problem of obtaining mass flow signal from the CFB pilot boiler relating this problem to learning under concept drift. In Section 3 we focus on two major subtasks, learning a predictor and identifying the change points, for which we explore three alternative change detection methods. The results of the experimental study with real data collected from the pilot CFB boiler are discussed in Section 4. We conclude with a summary and discussion of the further work in Section 5.

## 2. RESEARCH PROBLEM

In this section we will first describe the wider context of our research and then will focus on the particular problem of online mass flow prediction.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7...$5.00.

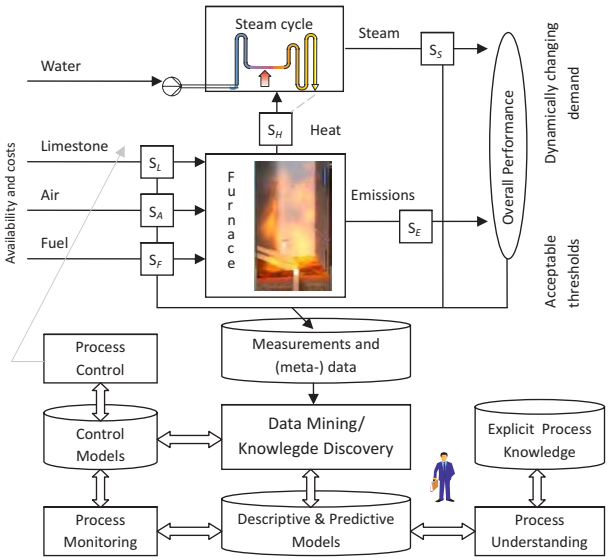



**Figure 1: A simplified view of a CFB boiler operation with the data mining approach.**

## 2.1 Control of CFB processes

The supercritical CFB combustion utilizes coal, biofuels, and multifuels in relatively clean, efficient and sustainable way. However, it needs advanced automation and control systems because of its physical characteristics (relatively small steam volume and absence of a steam drum).

The requirements for control systems are tight, since fuel, air and water mass flows are directly proportional to the power output of the boiler. That is especially relevant in CFB operation where huge amount of solid material exist in the furnace [11]. For large CFB boilers understanding of the process and the process conditions affecting heat transfer, flow dynamics, carbon burnout, hydraulic flows is critical in addition to the mechanical design.

A simplified schematics of a CFB boiler operation is presented in Figure 1. Fuel or a mixture of fuels, air and limestone are the controlled inputs to the furnace. The fuel is used for heat production. The air is added to enhance the combustion process. Limestone reduces the emission of sulfur dioxides ($SO_2$).

The produced heat is used to convert water into steam, which is the final output. The systems performance is monitored using the sensors $S_F$, $S_A$, $S_L$, $S_H$, $S_S$ and $S_E$. The measurements are collected in database repository together with meta-data describing process conditions and used for offline and online analysis. That is where DM techniques come into play. Predictive and descriptive models that can be further utilized to facilitate process monitoring, process understanding, and process control.

## 2.2 Mass flow prediction

The combustion and emission performance of different types of solid fuels and their mixtures are studied at VTT's 50 kW CFB pilot boiler (see Appendix for a schematic view).

Fuel can be fed into the reactor through two separate feeding lines (Figure 2). There is a fuel screw feeder in each line at the bottom of the silo and a mixer, which prevents arching of the fuel in the silo. The fuel silos are mounted on top of the scales. The scales measure a mass flow rate of the solid fuels as a decrease in weight in time. The signal is fluctuating with constant screw feeder rotational speed. It depends on the quality of the fuel changes (e.g. moisture content, particle size), the fuel grades in the silo. Particles might jam in between the screw rotor and the stator causing a peak in the mass signal. fuel addition causes a step change in the mass signal. The mass flow decreases in between fillings in line with the level of fuel in the tank.

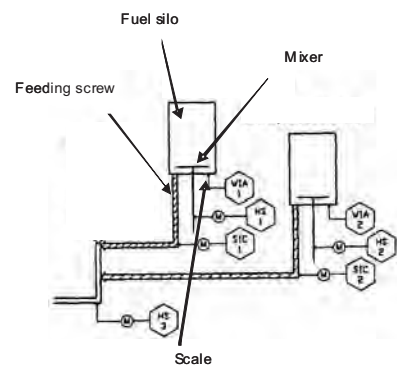



**Figure 2: A fuel feeding system of the CFB-reactor.**

Due to the fluctuation in the signal of the scales no reliable online data can be obtained from the mass flow of fuel to the boiler. The measurement system cannot be easily improved and therefore the mass flow is calculated offline using longer time period averages. The ultimate goal of our study is to predict the *actual* mass flow based on the sensor measurements, that would allow to improve the control system of the pilot CFB boiler.

## 2.3 Online mass flow prediction as learning under concept drift

In case of CFB boiler, like in many other dynamic environments, the data flows continuously and the target concept, i.e. the mass flow in this case could change over time due to the different operational processes (like fuel reloading) or changes in these processes themselves (like change of consumption speed). Therefore, learning algorithms must recognize an abrupt change mass flow (and be able to distinguish it from the outliers) and adjust a model accordingly.

In data mining and machine learning this problem is generally known as concept drift, that is the changes in the (hidden) context inducing more or less radical changes in the target concept [13].

Roughly, adaptation methods can be divided into two categories, blind methods that adapt a learner at regular intervals (e.g. with a sliding window approach) without considering whether changes have really occurred, and informed methods, that only modify learning approach after a change was detected [3]. The later group of methods must be used

in conjunction with a drift detection mechanism which could be implemented in several ways such as estimating statistical difference between *old* and *new* sample, or observing the deterioration of the predictor performance with more recent data point.

In this work we employ the idea of explicit concept drift detection (that corresponds to the context change in the CFB operational settings) and learning of a new model each time after a drift was detected.

There are two important types of change in the mass flow of the boiler; (1) transitions from the process of adding fuel to the process of fuel consumption and vice versa, and (2) gradual changes typically caused by a changing mixture of different fuel types and changes in the speed of fuel consumption of replenishment (the latter type of change can be detected by using the rotation speed signal from the screw). In the context of this paper we will only focus on the sudden changes.

We limit the scope of this study to identification of sudden changes, leaving the discussion of handing gradual changes as a future work.

# 3. OUR APPROACH

In this section we first discuss some peculiarities and domain knowledge of the problem and explain how we learn a predictor for the case when the change points are assumed to be known, and then discuss three approaches for the explicit detection of change points.

## 3.1 Learning the predictor

The data were recorded with 1 Hz sampling rate (an example of mass flow signal collected during a typical experimentation with the CFB boiler is presented in Figure 3). In each test of the experiment the type of fuel and/or the rotation speed of the feeding and mixing screws can be varied.

The three major sources of noise in the measurements are mixing and feeding screws and the occasional jamming of the fuel particle between the stator and rotor in the screw. The rotation of the screws causes vibrations to the system that are reflected by the scales as high frequency fluctuations around the true mass value. In Figure 4 the evolution of the frequency content of the measurements is shown by means of the short-time Fourier transform [8], from which the influence of the screws is evident. Namely, the rotating parts induce oscillations to the measurements of the same frequency as the rotation frequency. The frequency content due to the screws is identified from the figure as contrasting vertical curves.

The jamming of the fuel particle causes an abnormally large upward peak to the measurements that can be seen from Figure 3. The speed of the mass change in the tank at a given time depends not only on the rotation speed of the feeding screw and the speed of the replenishment of the fuel in the tank, but also on the amount of the fuel in the tank. The more fuel is in the tank the more fuel gets in the screw, since the weight of the fuel at the higher levels of the tank compresses (increases the density) the fuel in the lower levels and in the screw. The size and grade of fuel also have an effect on the compression rate of the fuel. Therefore, we assume that the mass flow signal has a nonzero second derivative.

The measured signal at time $t$ can be modeled as the sum of the true mass ($m_t$) and measurement noise ($\Sigma_t$):

$$y_t = m_t + \Sigma_t. \tag{1}$$

In a light of the discussed reasoning from the domain perspective, true mass at time $t$ can be described as:

$$m_t = \frac{a \cdot (t - t_0)^2}{2} + v_0 \cdot (t - t_0) + m_0, \tag{2}$$

where $a$ is acceleration of the mass change, $v_0$ stands for the speed of the mass change at time $t_0$, $m_0$ is the initial mass at time $t_0$.

The noise component at time $t$ can be described as:

$$\begin{aligned} \Sigma_t &= A \cdot \sin(\omega_{feed} \cdot (t - t_0) + \alpha_{feed}) + \\ &+ B \cdot \sin(\omega_{mix} \cdot (t - t_0) + \alpha_{mix}) + e_t, \end{aligned} \tag{3}$$

where $A$ and $B$, $\omega_{feed}$ and $\omega_{mix}$, $\alpha_{feed}$ and $\alpha_{mix}$ are amplitude, frequency and phase of the fluctuations caused by feeding and mixing screws, respectively; $e_t$ denotes the random peaked high amplitude noise caused by the jamming of the fuel particle at time $t$.

One solution to predict the true mass flow signal is to use stochastic gradient descent [2] to fit the model (1) without $e_t$ term to the measured data with the high amplitude peaks skipped from the learning process. This is closely related to fitting the model (2) to the same data in the mean-least-squares sense as the noise fluctuations are symmetric relatively to the true mass signal. Alternatively, a linear regression approach with respect to the second order polynomial can offer a better local stability and faster convergence. As the accurate mass flow measurements are required on-line by a control system the choice of the linear regression method seems more reasonable. The chosen prediction model is as follows:

$$\hat{y}_t = \frac{a \cdot (t - t_0)^2}{2} + v_0 \cdot (t - t_0) + m_0 + \epsilon_t, \tag{4}$$

where $\epsilon_t$ summarizes all the noise components and is treated as random.

To learn a regressor, the Vandermonde matrix [4] $\mathbf{V}$, which elements $v_{i,j}$ are the powers of independent variable $x$, can be used. In our case the independent variable is time $x_i = t_{i-1} - t_0,\ i = 1, \ldots, T$, where $T$ denotes the number of the time samples. If the linear regression is done for a polynomial of order $n$ ($p^n(x) = p_n x^n + p_{n-1} x^{n-1} + \ldots + p_1 x + p_0$), $\mathbf{V}$ is computed from the observed time series of the independent variable as follows:

$$v_{i,j} = x_i^{n-j+1},\ i = 1, \ldots, T,\ j = 1, \ldots, n+1, \tag{5}$$

where $i$ and $j$ run over all time samples and powers, respectively. Provided with $\mathbf{V}$ the problem of polynomial interpolation is solved by solving the system of linear equations

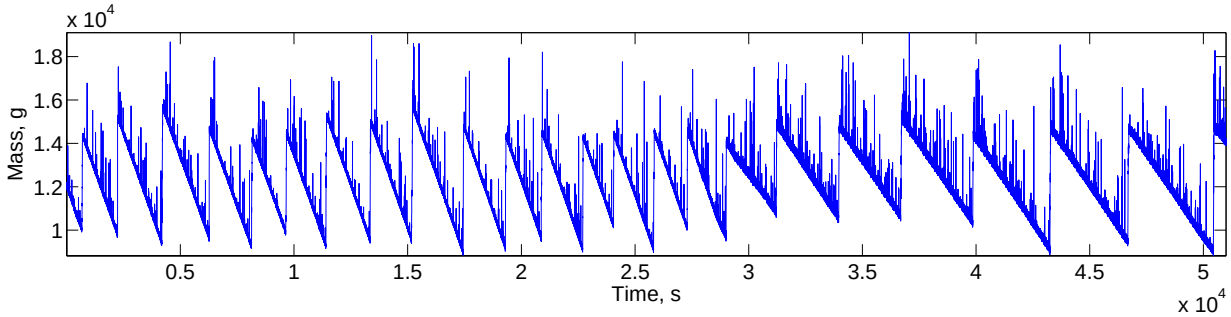



**Figure 3: Measurements of the fuel mass in the tank.**

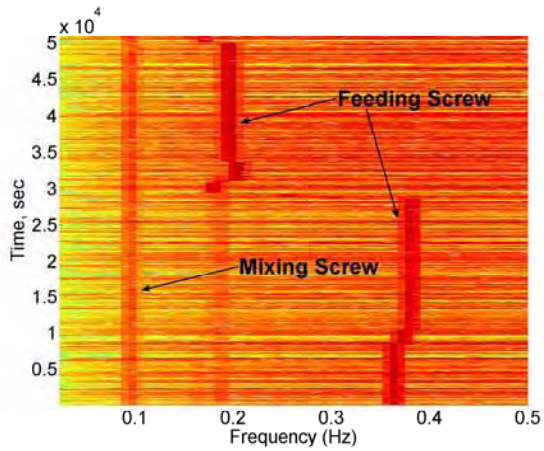



**Figure 4: Spectrogram of the fuel mass measurements computed with Short-Time Fourier Transform. The color denotes the power spectral density.**

$\mathbf{V}\mathbf{p} \cong \mathbf{y}$ with respect to $\mathbf{p}$ in the least square sense:

$$\widehat{\mathbf{p}} = \text{argmin}_{\mathbf{p}} \sum_{i=1}^{T} (\sum_{j=1}^{n+1} V_{i,j} p_{n-j+1} - y_i)^2 \tag{6}$$

Here, $\mathbf{p} = [p_n \; p_{n-1} \; \ldots \; p_1 \; p_0]^{\mathrm{T}}$ denotes the vector of the coefficients of the polynomial, and $\mathbf{y} = [y(x_1) \; y(x_2) \; \ldots y(x_T)]^{\mathrm{T}} = [y_1 \; y_2 \; \ldots y_T]^{\mathrm{T}}$ is the time series of the dependent variable that is indication of the scales. Provided that the $n+1$ columns of the matrix $\mathbf{V}$ are linearly independent, this minimization problem has a unique solution given by solving the normal equation [6]:

$$(\mathbf{V}^{\mathrm{T}}\mathbf{V})\widehat{\mathbf{p}} = \mathbf{V}^{\mathrm{T}}\mathbf{y}. \tag{7}$$

For the modeling of the mass flow we have chosen second order polynomial, where $\widehat{p}_0 = \widehat{m}_0$, $\widehat{p}_1 = \widehat{v}_0$, $\widehat{p}_2 = \widehat{a}$. We distinguish the two types of the periods in the experiment: the consumption (fuel is only consumed) and the consumption with fuel replenishment.

When one period of the CFB operating changes to another (i.e. a new portion of fuel is being added) the process of mass flow prediction starts over again, as the model of the mass flow changes. Thus, the most problematic unstable regions are the transitions intervals, when the parameters of the model change their values.

When a period of the session starts the samples of measurements start to accumulate in the buffer. The data in the buffer are used to fix the parameters of the mass flow prediction model. Only the samples that do not contain high amplitude peak (due to jamming) are placed to the buffer. The buffer is emptied after each change point (that assumed to be known in this setting) and starts to accumulate new data.

The first measurement within the experiment is taken as the prediction of the mass flow signal at that time $\widehat{m}_1 = y_1$ and as the first point that is placed to the buffer. In contrast, the first prediction of the mass within a following period is taken as the last prediction of the mass from the previous period. In addition, the last predicted point from the previous period $\widehat{m}_{i_c}$ is placed to the buffer as the first point for the new period, where $i_c$ denotes the number of the sample when the change of the periods occurs. When a new sample arrives the parameters of the model are estimated based on the points that are in the buffer independently of whether the current point was placed to the buffer or not. The current prediction is computed respectively based on the current model. Depending on the number of data points $T$ in the buffer different models are used:

- if $T = 1$, then $\widehat{m}_i = \widehat{m}_{i-1}$;
- if $1 < T \leq 4$, then $\widehat{p}_2 = 0$ and $\widehat{p}_0$ and $\widehat{p}_1$ of the first order polynomial are estimated from the available data, and $\widehat{m}_i = \widehat{p}^1(x_i)$, where $\widehat{p}^1(x_i)$ is the current prediction by the first order polynomial.
- if $T > 4$, then the second order polynomial model is fitted to the data, and $\widehat{m}_i = \widehat{p}^2(x_i)$.

In practice the operational settings often allow a delay between the data arrival and the evaluation of the signal of interest at this time sample. This means that the estimate of the signal at a given time sample is obtained based on the data that is accumulated also during the future times. This allows the more accurate estimates of the signal to be computed. Note that in our case this will have an effect of

increased accuracy mainly for the beginning of a new experiment, when the amount of the data in the buffer is small yet.

## 3.2 Explicit change detection methods

In this section we consider three different approaches to detect change points and thus facilitate modeling the transitions from one state of the system to another. All three are based on measuring statistical data in a moving window. The first approach is nonparametric and is based on the Mann-Whitney U test. The second approach is based on a parametric test on the performance of the local models. The third approach is based on the statistical analysis of the raw data.

### *3.2.1 Nonparametric change detection*

Let's assume that when the underlying process in the boiler changes, the current model for the prediction of the data will perform worse. The idea behind nonparametric change test is to compare the prediction performances obtained on different subsets of the data. For the classification of whether an observed point is outlier (peak due to a fuel particle jamming) or not, the statistical nonparametric Mann-Whitney U test [7] that compares the two distributions of MSE values can be used. Namely, if a point is to be tested for being an outlier with respect to a window of previous points or, in general, with respect to a set of surrounding points, the following procedure is applied to construct the two samples of MSE values:

- First, a leave-one-out cross validation (LOOCV) is done on a reference set of $N-1$ points, a line is fitted to each of the possible subsets of size $N-2$, and MSE of the fit is computed for fitted $N-2$ points only. Thus, at this stage we obtained a sample of $N-1$ MSE values that characterizes the set of reference points. We call the distribution of these MSE values a reference distribution for the tested point.
- Then, the test distribution composed of $(N-1)(N-2)$ MSE values is computed by replacing each point in each subset of size $N-2$ from previous stage for a test point, computing a linear fit and MSE value for the $N-2$ points used to compute the fit. Therefore, test distribution contains MSE values for a linear fit, when a test point is used. This way we ensure that the fittings and MSE computations are always done using $N-2$ points, but the sizes of reference and test samples are different.

The above considerations are demonstrated in Figure 5 in the second and third steps. In fact, the first step is just a LOOCV, when each point in the initial window of size $N$ is made a test point, and the rest form reference set.

Note that in Figure 5 a less general case is shown compared to a discussion. The illustration considers a set of neighboring points, although in general the points in the set can be drawn from distanced time locations depending on the pursued goal. By changing the size and spread of the reference set, one controls how the decision made using the Mann-Whitney U statistics is tailored/diversified over time.

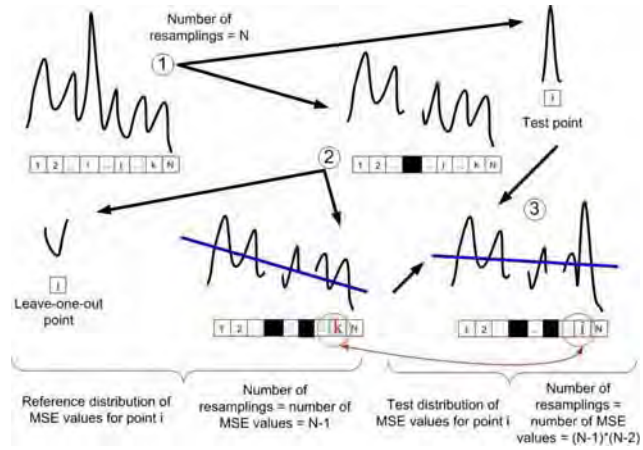



**Figure 5: The resampling procedure for constructing two distributions to be compared by Mann-Whitney U test.**

In other words, varying the window size adjusts the context for decision making to be more local, global, or balanced.

The Mann-Whitney U test applied to the reference and test distributions for a given confidence level $\alpha$ shows whether MSE of the fitted points including test point is systematically different from the case when the test point is excluded. If such tendency is detected by the test, the test point is classified as outlier.

Based on the process of identification of the outlier, we define a criterion for state change detection as the fact of observing several outliers in a row ($o2c$).

In our experiments we used $\alpha = 10^{-9}$ (confidence level of Mann-Whitney U test); $d = 10$ points as delay after which the prediction for the point appeared $d$ points prior to the current point must be output, $r = 16$ points as the maximal size of the reference window (composed of the most recent precedents to the currently considered point) against which the newly arrived point is compared using outlier test, and $r_a = max\{d, r\}$ is the actual reference window size; $f = 150$ points as the maximal size of the window that is used to compute the model parameters to predict the current output, and $f_a = max\{d, f\}$ is the actual size of the fit window; finally $o2c = 5$ is a (currently predetermined) number of detected outliers in a row that is a sufficient condition to assume the state change.

The method operates as follows. Until the first $d+1$ points are accumulated from the start of the new state nothing is output. In the following all considerations are assumed to be referenced to a start of the current state. For simplicity we assume that we start in the fuel consumption state.

When first $d$ points are accumulated, an outlier test is made for each of the $d$ accumulated points. All found outlier points are substituted with an interpolation (by fitting a line to the non-outlier points). After this, when a new point arrives, it is tested for outlier with respect to $r_a$ previous points. If it is

classified as outlier, it is replaced by an extrapolation from the $r_a$ previous points. Then the point 'delayed' by $d$ points from the current one is predicted by the linear fit to the latest sequence of outlier-corrected $f_a$ points. If $o2c$ outliers in a row are noticed, the state change (pointing $o2c$ points back in time) is alarmed and the detected 'false outliers' (that actually belong to the change) are restored (from the backup buffer). The process continues then from the beginning of the newly detected phase.

### 3.2.2 Parametric change detection

The nonparametric approach is relatively time consuming, thus we look for a more efficient change detectors.

When the underlying process in the boiler changes, the current model for the prediction of the data will perform worse. This behavior can be detected by keeping track of two statistical properties of the performance [3]. The first is the error rate that signifies the probability of miss classifying the actual value $y_t$ of the signal. The second is the standard deviation of the error rate. In [3] it is assumed that the classification task can be modeled by a binomial distribution. Since the mass flow is continuous, we have to assume that given a large enough sample or window, the binomial distribution is close to a normal distribution with the same mean and standard deviation. This is used in an online test to see whether the signal has changed.

Since this task is in continuous space, we use the Mean-Squared-Error (MSE) as metric for change detection instead of the error rate. For each time step the window is moved and for point $x_t$ all local reference MSEs are calculated with LOOCV.

We assume that the model prediction performance is stable if the local found reference MSE ($E_t$) satisfies

$$E_t + S_t < E_{min} + \alpha S_{min}, \tag{8}$$

where $E_{min}$ is the minimal found reference MSE, $S_{min}$ the minimal found reference standard deviation, and $\alpha$ a parameter that reflects the level of confidence. If there is a large enough deviation in the signal the algorithm will report a warning. This level is determined by the condition:

$$E_{min} + \alpha S_{min} < E_t + S_t < E_{min} + \beta S_{min}, \tag{9}$$

where $\beta$ is the upper confidence bound signifying a change. Since it might occur that there is a local change in the signal (an outlier) it is not possible to go the change state immediately. If $E_t + S_t > E_{min} + \beta S_{min}$ the algorithm reports a change and it will switch to the initial state of the other process. When this happens $E_{min}$ and $S_{min}$ are reset to the minimal found values in the new regime.

This procedure puts a strict lower bound on the window size. Since we have to assume that the normal distribution is representative for the distribution of reference MSEs, the window size should be at least 30 consecutive points. In principle, a small window size is preferable when trying to detect rapid changes. But a smaller window size will result in a higher variation in the local models. In our experiments we used $\alpha = 2$, $\beta = 3$ (lower and upper bounds for the confidence interval), and the window size of 30 points.

For each 30 accumulated points, the LOOCV is used. For each $N-1$ local model the MSE is calculated and it's standard deviation. The transition conditions are checked and when an outlier is detected it is ignored by the global fit. If, after this, the algorithm detects a change, the boundaries for the transition criteria are reset and the global fit is relearnt for the future points.

### 3.2.3 Change detection using raw data

To reduce the degrees of freedom introduced by MSE based change detectors, we employ a change detection method based on raw data. We choose ADWIN, which was presented in [1]. It is based on the differences between the means of raw data. The method was originally designed for univariate sequential data. The method works as follows: given a sequence of signals it checks whether there are statistically significant differences between the means of each possible split of the sequence. If statistically significant difference is found, the oldest portion of the data backwards from the detected point is dropped and the splitting procedure is repeated until there are no significant differences in any possible split of the sequence. More formally, suppose $m_1$ and $m_2$ are the means of the two subsequences as a result of a split. Then the criterion for a change detection is $|m_1 - m_2| > \epsilon_{cut}$, where

$$\epsilon_{cut} = \sqrt{\frac{1}{2m} \log \frac{4n}{\delta}}, \tag{10}$$

$$m = \frac{1}{\frac{1}{n_1} + \frac{1}{n_2}}, \tag{11}$$

here $n$ is total size of the sequence, while $n_1$ and $n_2$ are sizes of the subsequences respectively. Note that $n = n_1 + n_2$. $\delta \in (0, 1)$ is a hyperparameter of the model. In our experiments we used $\delta = 0.3$, $n = 200$.

## 4. EXPERIMENTAL RESULTS

In this Section we first present the performance results under the assumption of known change points and then demonstrate the accuracy of the considered change detection techniques.

### 4.1 Predictor performance under the assumption of known change points

As the main points of interest are the points, where the change of the period occurs, for this analysis we took an interval of the experiment containing both types of the transition points (see Figure 6).

The computations of mass flow were done for the delay time varying from 0 to 20 samples/seconds with step 1.

It takes about $100 - 150$ seconds from the beginning of each consumption period for the model of the mass signal stabilizes. Since the replenishment period lasts during shorter times ($3 - 40$ seconds), the model may exhibit instability even at the end of the period. However, overall the prediction of the mass during the replenishment period is satisfactory, because changes of the mass are rather steep and prominent against the background of the noise. For comparison, the effect of increased delay time is shown in Figure 7.

**Table 1: Accuracy of detecting sudden changes in feeding ($\phi$) and consumption ($\kappa$) processes, and of detecting outliers ($o$).**

| | Parametric test | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\phi$ | P | N | $\kappa$ | P | N | $o$ | P | N |
| T | 19 | 50898 | T | 4 | 50883 | T | 799 | 49088 |
| F | 74 | 5 | F | 89 | 20 | F | 892 | 217 |
| | Nonparametric test | | | | | | | |
| $\phi$ | P | N | $\kappa$ | P | N | $o$ | P | N |
| T | 24 | 50952 | T | 8 | 50942 | T | 816 | 49434 |
| F | 20 | 0 | F | 20 | 16 | F | 546 | 200 |
| | ADWIN method | | | | | | | |
| $\phi$ | P | N | $\kappa$ | P | N | $o$ | P | N |
| T | 24 | 50960 | T | 0 | 50972 | T | - | - |
| F | 12 | 0 | F | 0 | 24 | F | - | - |

To indicate the rate of convergence of the model with respect to the delay time we computed the mean-square error between the true mass prediction for each pair of the consecutive delays (see Figure 8):

$$MSE(\tau) = \frac{1}{T}\sum_{i=1}^{T}(\widehat{m}_i^{\tau} - \widehat{m}_i^{\tau-1})^2. \qquad (12)$$

It can be clearly seen from the figure that for the small delays ($1-4$ samples) the prediction accuracy improves dramatically, and for the larger delays the improvement slows down.

## 4.2 Performance of change detection methods

The results are summarized in Table 1. For each method we present confusion matrixes of detecting sudden changes in feeding ($\phi$) and consumption ($\kappa$) processes, and of detecting outliers ($o$).

***The parametric test*** succeeds in identifying 19 out of 24 changes from feeding state to consumption state. However, it finds only 4 out of 24 transitions the other way around. The worst part of the results is the number of falsely identified changes(74 for feed-to-consumption and 89 for consumption-to-feed). At these points the algorithm will signal a change where in fact there is none. This results in a very poor fit as demonstrated with an example in Figure 9.

The number of unidentified outliers is also important, since it will greatly affect the performance of the model if this number is high. For the parametric test this number is 217 out of 1016. This is roughly 20% of the total number of outliers. This number is too high for the algorithm to be

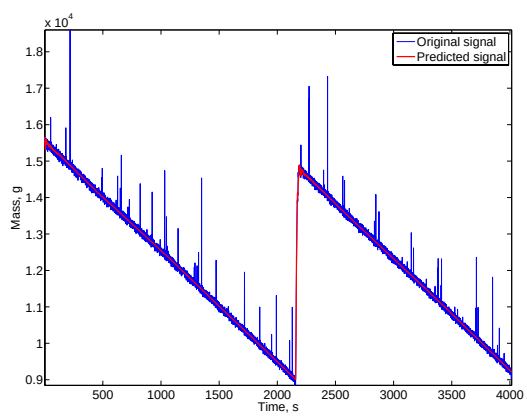



**Figure 6: Online mass flow prediction with the switching regression models and zero delay time.**

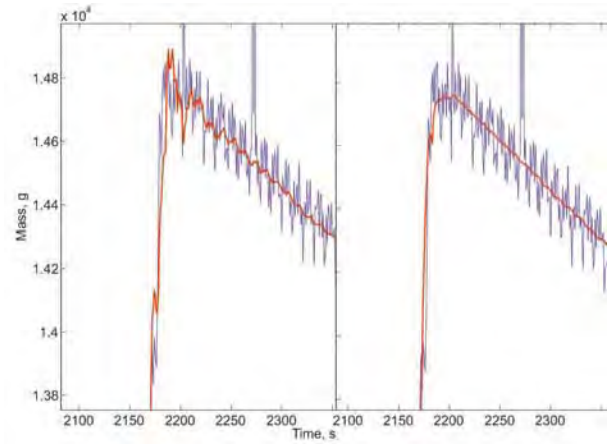



**Figure 7: Zooming to the transition point with zero (left) and 20 (right) samples delay.**

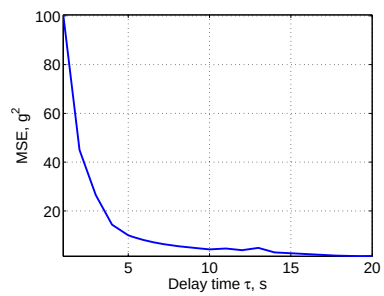



**Figure 8: Mean-square error between the mass signal predictions for each pair of the successive delays.**

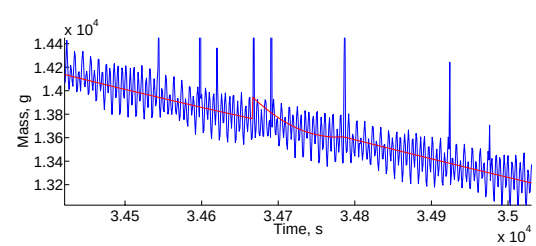



**Figure 9: An example of impact of misclassifying an outlier as a change (red line).**

effective. The relatively bad results might be caused by the fact that the assumption that the local MSEs are normally distributed is not valid.

***The Mann-Whitney U-test*** does not assume a normal distribution and its accuracy is noticeably better. It enables finding all of the consumption-to-feed transitions. However, this test also incorrectly identifies 20 (false) change points for this transition. The other type of change is recognized 8 times out of 24 and the number of false classifications is also high. The number of outliers detected is comparable to that of the parametric test.

The most intolerable and harmful for prediction detection errors are related to early (see Figure 10) and late (see Figure 11) determination of the feeding-to-consumption transition. If the early or late detection happens, not only the predictions of the transition period itself deteriorate, but also a large portion of the following or prior to the detection event consumption interval lacks accurate interpolation, because points from feeding are also used to compute fit for consumption. Short perturbations of signal during the feeding state may lead to early false detection of the transition, because the outliers may be misinterpreted as indication of the change (see Figure 10).

At the same time the recognition of the fact that feeding continues may be problematic, since no points of appropriate number and properties are observed in the following window. The current implementation does not take into account prior signal once state change is detected. This has an effect of resetting the context. This way only one level of verification if the change is true is used: observing relatively large numbers of outliers in a row. But additional level of verification can be advised by continuing back-tracking, while new points arrive. In other words, we may wish to check if the situation is really what it appears to be also with respect to the following part of the signal, and whether the change was real. The late determination of the feeding-to-consumption transition may be a consequence of a too short feeding phase that is perceived by the method as outlier (see Figure 11).

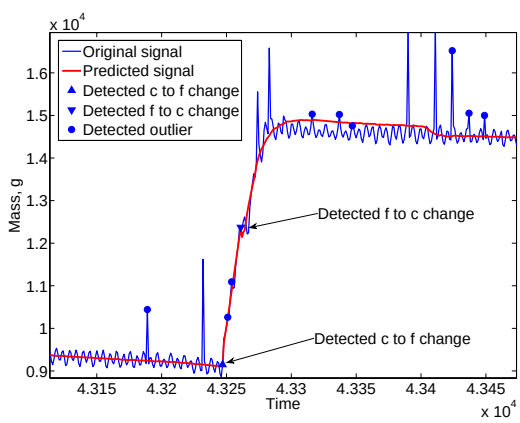



**Figure 10: An example of early detection of the feeding-to-consumption transition.**

The reasons for delayed transition detection considered above

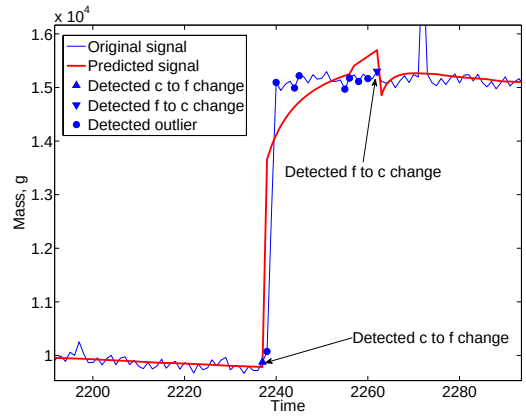



**Figure 11: An example of late detection of the feeding-to-consumption transition.**

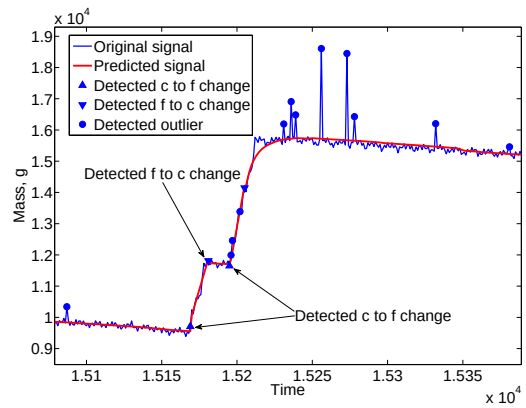



**Figure 12: Detecting plateau inside feeding state.**

are well understood. The errors happen due to a locally tuned criteria for outlier and change detection. However, the more sophisticated windowing schemes can be used. For example, an adaptive window size approach can be employed (however, this requires further study of the data patterns observed in the data). Besides, a simple back-tracking mechanism can be used to verify the already made decision with respect to subsequently arrived points.

It is worth noticing that not always outlier behavior can be unambiguously distinguished from the true behavior of the signal. For example, in Figure 12 the plateau inside the feeding phase was successfully identified. In this case it is clear that feeding was stopped for a while and than continued again. However, if such plateau is small it is generally unknown if this is a false measurement or consequence of real process of mass flow. In our computations, the small plateaus inside feeding phase sometimes can be ignored by a detection procedure or lead to an early detection of the transition to consumption with or without following detection of continuing feeding.

***The ADWIN method*** required on average a lag of 40.7

seconds to detect the start of the feeding stage. Given the fact that the average length of this stage (in our data) is 26.0 seconds the detection in most cases occurred after it was already finished. The shortage of the feed stage explains the fact that there were no detections of the consumption stage using this method.

From the careful visual inspection it was possible to conclude that the method detected all the feed stages and made no false alarms (no outliers were mistreated for the change). The method seems to be slow in reaction but reliable and can be of a particular use for verification of the changes, since there are no "ground truth" available.

Note that although the change is detected with a lag, but the actual signaled change point is quite close to the original (on average 7.5 seconds away from the "true" change points). The numbers in Table 1 are based on the detected change points, not the detection time.

***Computational complexity*** of the considered change detection methods. Assuming that at time $t$ we have a history of $t$ datapoints available, but we delimit the back search to $N$ past data points. Both parametric and nonparametric change detection algorithms employed use LOOCV procedure, which requires multiple scans to build local prediction models to compute MSEs. The parametric model would require $\sim N^2$ iterations to output the detection decision at time $t$, but once the records for MSE are kept, the subsequent detections would require only $\sim N$ iterations per time point. Nonparametric change detection involves comparison between the two distributions, thus it requires $\sim N^3$ iterations per time point. Change detection using raw data requires $N$ splits to test for a change at time $t$.

The methods were implemented in MATLAB and the performance time of all the methods is reasonable for online operation. We have a series consisting of $\sim 50000$ entries, which correspond to seconds. That gives a restriction for implementation to be reasonable for online operation if all the experiment is completed in less than 14 hours. The factual performance in our implementation was $3-4$ times more quick.

## 5. CONCLUSIONS AND FURTHER WORK

Prediction of mass flow in CFB boilers in online settings is an important and challenging problem having clear connections to the problem of learning under the presence of sudden and gradual concept drifts.

In this paper we presented a regression learning framework for fuel mass prediction. Change and outlier detection is the key component in this framework, it regulates switching of the predictors. We studied three alternatives of the explicit detectors.

Our results demonstrated that (1) when the change points are correctly identified our predictors perform reasonably well and a delay of less than 5 seconds allows to predict the mass flow accurately enough to be used as a reliable indicator for the CBF control system, (2) when the change points need to be determined online without any additional input, the nonparametric approach shows much better performance than parametric one in terms of accuracy; however it is much slower and therefore requires more computational power to guarantee in time prediction, (3)The ADWIN method was the very precise in consumption-to-replenishment change detection, while the lag of detection was quite large and therefore this method can be used only as a reference for the "ground truth", but is not applicable for the operational settings of the CFB boiler.

The directions of our further work in mining CFB sensor data include (1) studying in more detail the transition period at the end of the fuel feeding stage, (2) considering an effect of fuel feeding speed, and the effect of using different mixtures of fuels, (3) exploring the potential of adaptive context window size approach and back-tracking mechanisms for improved detection of state changes, and that is not least important (4) external validation of our online mass flow prediction approach being implemented as part of the control system of the pilot CFB boiler in operational settings with different conditions.

We anticipate that local reliability estimation of mass flow prediction [10] may be helpful for the domain experts. Furthermore, our experimental study also suggests that learning from multiple experiments and better utilizing additional relevant information available from various sensors (e.g., rotation speed of the feeding screw) should bring a further improvement of the reliability of online mass flow prediction.

## 6. ACKNOWLEDGMENTS

This research is partly supported by TEKES (Finnish Funding Agency for Technology and Innovations) DYNERGIA project and NWO (The Netherlands Organization for Scientific Research) HaCDAIS project. The authors are thankful to the domain experts, Timo Leino and Mikko Jegoroff from VTT, Technical Research Centre of Finland for their endless support and valuable discussions.

## 7. REFERENCES

[1] A. Bifet and R. Gavaldà. Learning from time-changing data with adaptive windowing. In *SDM*. SIAM, 2007.

[2] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2001.

[3] J. Gama and G. Castillo. Learning with local drift detection. In X. Li, O. R. Zaïane, and Z. Li, editors, *ADMA*, volume 4093 of *Lecture Notes in Computer Science*, pages 42–55. Springer, 2006.

[4] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, 1991.

[5] A. Ivannikov, M. Pechenizkiy, J. Bakker, T. Leino, M. Jegoroff, T. Karkkainen, and S. Ayramo. Online mass flow prediction in cbf boilers. In *Proc. 9th Industrial Conference on Data Mining (ICDMŠ09)*. Springer-Verlag, 2009.

[6] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[7] H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18:50–60, 1947.

[8] S. Nawab and T. Quatieri. Short time fourier

transform. In J. Lim and A. Oppenheim, editors, *Advanced topics in signal processing*, pages 289–337. Prentice Hall, 1988.

[9] M. Pechenizkiy, A. Tourunen, T. Kärkkäinen, A. Ivannikov, and H. Nevalainen. Towards better understanding of circulating fluidized bed boilers: a data mining approach. In *Proceedings ECML/PKDD Workshop on Practical Data Mining*, pages 80–83, 2006.

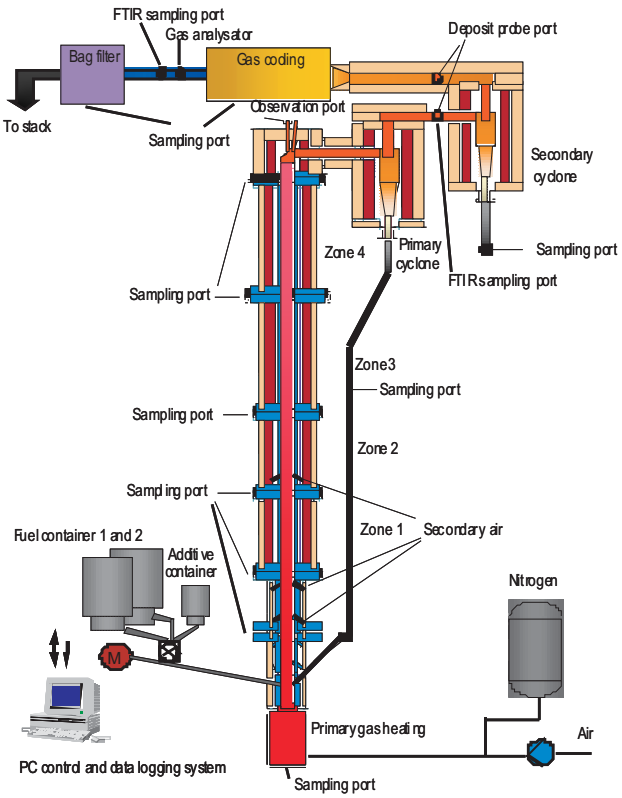
[10] P. P. Rodrigues, J. Gama, and Z. Bosnic. Online reliability estimates for individual predictions in data streams. In *ICDM Workshops*, pages 36–45. IEEE Computer Society, 2008.

[11] J. Saastamoinen. Modelling of dynamics of combustion of biomass in fluidized beds. *Thermal Science*, 8(2):107–126, 2004.

[12] Z. Song and A. Kusiak. Constraint-based control of boiler efficiency: A data-mining approach. *IEEE Trans. Industrial Informatics*, 3(1):73–83, 2007.

[13] G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, 23(1):69–101, 1996.

**Appendix.** The laboratory scale CFB-reactor. The height of the riser of the boiler is 8 m and the inner diameter 167 mm. The reactor is equipped with several separately controlled electrically heated and water/air cooled zones in order to control the process conditions, for example, oxygen level, temperature and load almost independently. Several ports for gas and solid material sampling are located in the freeboard area.



# An Exploration of Climate Data Using Complex Networks

Karsten Steinhaeuser
Dept. of Comp. Sci. & Eng.
University of Notre Dame
GIST Group, CSED
Oak Ridge National Laboratory
ksteinha@nd.edu

Nitesh V. Chawla
Dept. of Comp. Sci. & Eng.
Interdisc. Center for Network Science & Applications
University of Notre Dame
Notre Dame, IN 46556
nchawla@nd.edu

Auroop R. Ganguly
GIST Group, CSED
Oak Ridge National Laboratory
Oak Ridge, TN 37831
gangulyar@ornl.gov

## ABSTRACT

To discover patterns in historical data, climate scientists have applied various clustering methods with the goal of identifying regions that share some common climatological behavior. However, past approaches are limited by the fact that they either consider only a single time period (snapshot) of multivariate data, or they consider only a single variable by using the time series data as multi-dimensional feature vector. In both cases, potentially useful information may be lost. Moreover, clusters in high-dimensional data space can be difficult to interpret, prompting the need for a more effective data representation. We address both of these issues by employing a complex network (graph) to represent climate data, a more intuitive model that can be used for analysis while also having a direct mapping to the physical world for interpretation. A cross correlation function is used to weight network edges, thus respecting the temporal nature of the data, and a community detection algorithm identifies multivariate clusters. Examining networks for consecutive periods allows us to study structural changes over time. We show that communities have a climatological interpretation and that disturbances in structure can be an indicator of climate events (or lack thereof). Finally, we discuss how this model can be applied for the discovery of more complex concepts such as unknown teleconnections or the development of multivariate climate indices and predictive insights.

## Categories and Subject Descriptors

I.5.3 [**Pattern Recognition**]: Clustering; I.5.4 [**Pattern Recognition**]: Applications—*Climate*

## General Terms

Algorithms, Experimentation

## Keywords

clustering, networks, community detection, spatio-temporal data, climate, mining scientific data

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7...$5.00

## 1. INTRODUCTION

Identifying and analyzing patterns in global climate is important, because it helps scientists develop a deeper understanding of the complex processes contributing to observed phenomena. One interesting task is the discovery of climate regions (areas that exhibit similar climatological behavior, see Fig. 1), which has been addressed with various clustering methods. While k-means [5, 7] works well with multivariate data, it is limited to finding clusters of relatively uniform density and largely ignores the temporal nature of the domain. Alternate clustering approaches have been explored to address the space-time aspect of the data, including a weighted k-means kernel with spatial constraints [17] and a shared-nearest neighbor method to discover *climate indices* [24] from sea surface temperature data [18]. However, none of these approaches provides a means for explicitly identifying clusters from multivariate spatio-temporal data.

In this paper, we consider a different perspective on analyzing climate data. Instead of clustering based on univariate similarity or spatial proximity, we model the data as a *climate network* [21]. Physical locations are represented by nodes, and we introduce a *cross correlation-based measure of similarity* to create weighted edges (connections) between them. Edge placement is determined only by the relationship among multiple climate variables and is not subject to any spatial constraints, and a community detection algorithm discovers clusters corresponding to climate regions. This network view captures complex relationships and is able to identify patterns that span both space and time.

The concept of complex networks has been used to derive interesting climate insights. For example, [19] and [23] independently found that changes in network structure give predictive insights about El Niño events. A network of different climate indices was used to explain the major climate shifts of the 20th century as transitions between different equilibria of oscillators representing the earth system [20]. While all of these studies were hypothesis-driven, we believe that similar innovations are also possible with respect to the discovery of climate regions using a data-driven approach.

**Contributions**

- Using a network to model multivariate climate data
- A correlation-based measure for edge weighting
- Community detection to identify climate regions
- An empirical evaluation of our approach on a real-world dataset spanning 60 years

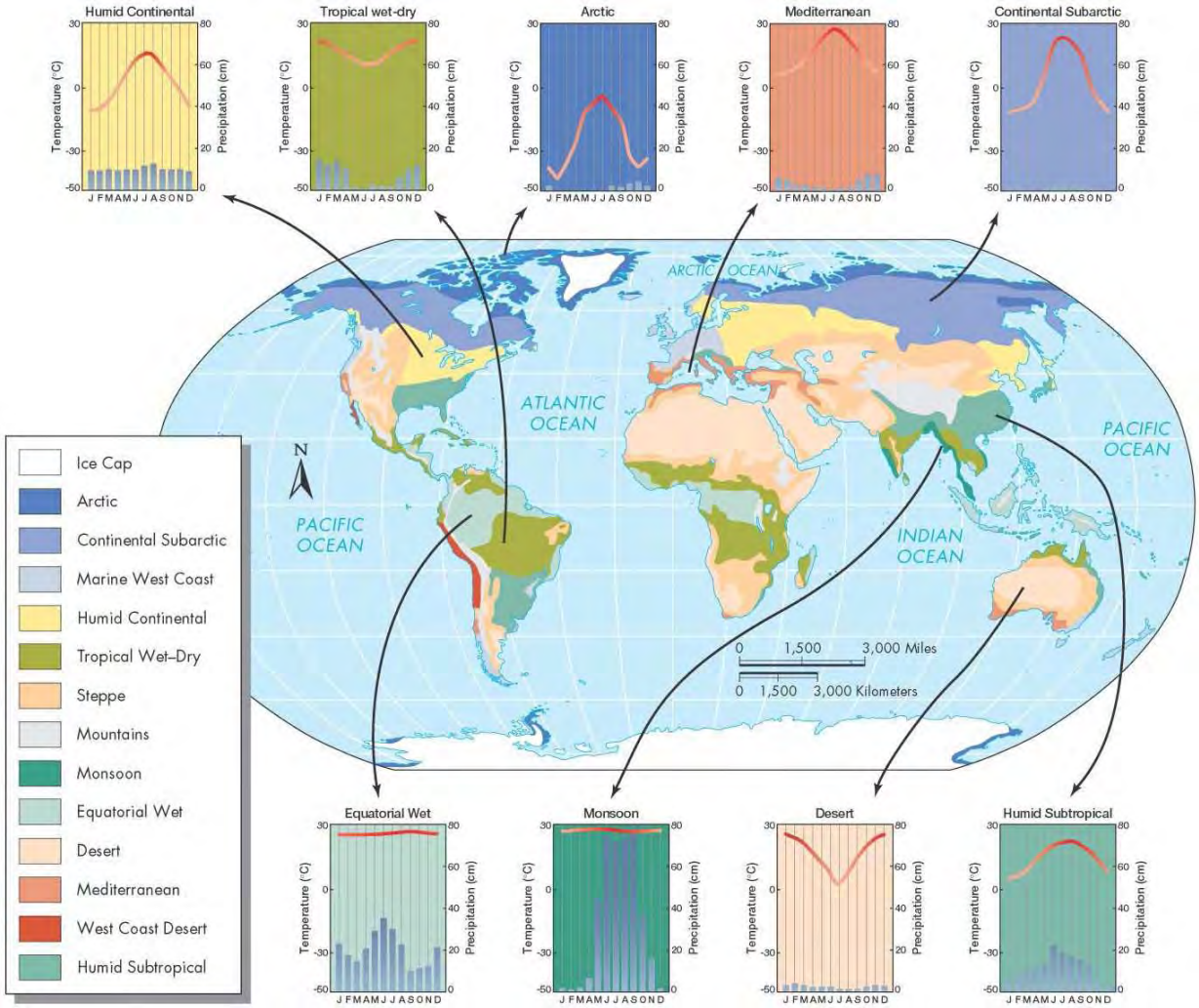



**Figure 1: Climate regions for temperature and precipitation, used with permission [12] (best viewed in color).**

**Organization**
In section 2 we briefly introduce the dataset used in this study. Section 3 describes the methodology in detail, and the empirical evaluation is presented in section 4. We conclude with a discussion placing this work in the broader context of climate science and identifying directions for future work.

## 2. HISTORICAL CLIMATE DATA

The Earth science data for our analysis stems from the NCEP/NCAR Reanalysis project [10], which is publicly accessible for download at [26]. This dataset is constructed by fusing and assimilating measurements from heterogeneous remote and in-situ sensors. Variable selection is an important issue in this context, one we have not yet fully explored. Previous research has relied on domain expertise for an appropriate selection [19, 21, 23]; alternatively, an objective feature selection approach could be used. For the purpose of this study, we selected four variables with the guidance of a domain expert: air temperature, pressure, relative humidity, and precipitable water, available at monthly intervals for a period of 60 years from 1948 to 2007 (720 points).

Temperature and pressure were chosen because they are two key variables in terms of significance, for example in defining climate regions [12] or in determining indices that may act as predictors [18, 19, 23]. Precipitation is another variable of great importance, but it is known to be inaccurate in reanalysis products due to its inherently large space-time variability [9]. Therefore, we use relative humidity and precipitable water as surrogates because these variables are relatively more stable and reliable in the reanalysis data [10].

Measurements are provided for points (grid cells) at a resolution of $2.5^\circ \times 2.5^\circ$ on a latitude-longitude spherical grid. Figure 2 shows a sample time series for each variable at the grid point closest to Paris, France (47.5°N 2.5°E).

## 3. THE NETWORK VIEW

In this section, we describe the major components of our methodology: using cross correlation to define a measure of similarity between locations, constructing the weighted climate network, identifying communities of interest, and studying their behavior over time.

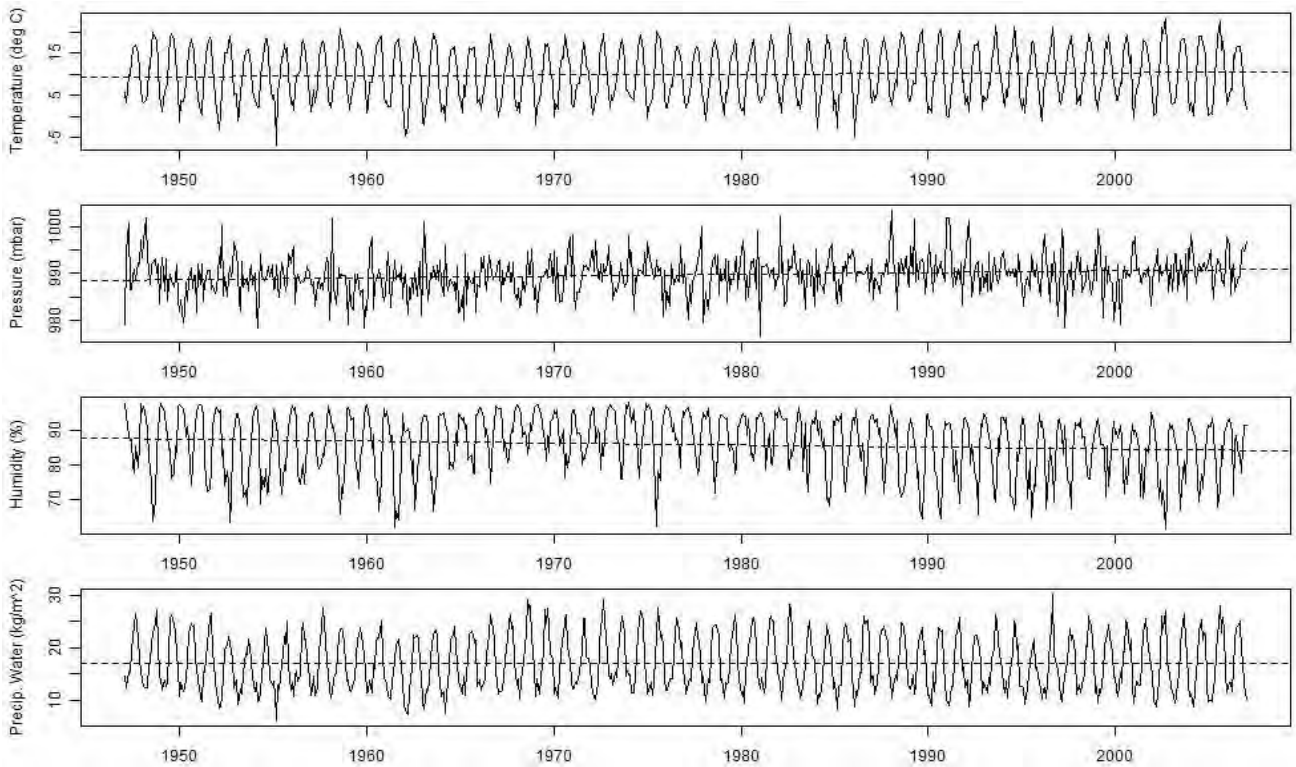



**Figure 2: Climate observations near Paris, France (47.5°N 2.5°E) from 1948 to 2007 with trend lines**

Figure 3 provides an overview of our approach; algorithm details, including pseudocode, are provided in Section 3.5.

## 3.1 A Similarity Measure for Climate Data Based on Cross Correlation

Various methods have been used for clustering climate data, and underlying each approach is some type of measure that defines the distance or, conversely, the *similarity* between two points. Traditional measures such as Euclidean [7] and Mahalanobis [5] distance have been employed in climate applications. But these may not be the most appropriate for high-dimensional and noisy data, or when clusters of varying density are known to exist within the data. To counter these problems, [4] proposes to define similarity locally based on the number of nearest neighbors two points share, which was demonstrated to provide good results with univariate climate data.

In the present application, each data point represents a physical location (grid cell) for which we have four separate time series corresponding to the four climate variables, and none of the aforementioned distance measures can take full advantage of the information contained therein. Therefore, we propose to define a new feature space based on the correlations between the four time series at each point, and similarity between locations is then measured as distance within this space.

Let $AT$, $PR$, $RH$, and $PW$ denote the time series for air temperature, pressure, relative humidity, and precipitable water, respectively, and let $t$ denote the number of data points in each series. Then, for any two series $A$ and $B$ the cross correlation function CCF at delay $d$ is computed as

$$CCF(A,B,d) = \frac{\sum_{i=1}^{t}[(a_i - \bar{a})(b_{i-d} - \bar{b})]}{\sqrt{\sum_{i=1}^{t}(a_i - \bar{a})^2 \sum_{i=1}^{t}(b_{i-d} - \bar{b})^2}} \quad (1)$$

where $a_i$ is the $i^{th}$ value in series $A$ and $\bar{a}$ is the mean of all values in the series. Note that the correlation coefficient ranges from -1 to 1, where 1 indicates perfect agreement and -1 perfect disagreement, while 0 indicates that no correlation is present at all. Since an inverse relationship is equally relevant in this application, we take the absolute value of the cross correlation function. In addition, cognizant that some climate phenomena may occur with some lag (i.e., at different times in different places), we account for this possibility by computing the cross correlation function for delays in the range $-6 < d < +6$ months and take the largest (absolute) value to be the correlation between $A$ and $B$.

Given the above definition, we now compute the correlation between all pairs of variables (time series) at each location, in this case $\binom{4}{2} = 6$ pairs. This results in a new 6-dimensional feature space wherein each grid cell is represented as a point defined by the correlations between climate variables at the corresponding location, as follows:

$$\mathbb{R}^6 = \langle CCF(AT,PR), CCF(AT,RH), CCF(AT,PW), CCF(PR,RH), CCF(PR,PW), CCF(RH,PW)\rangle$$

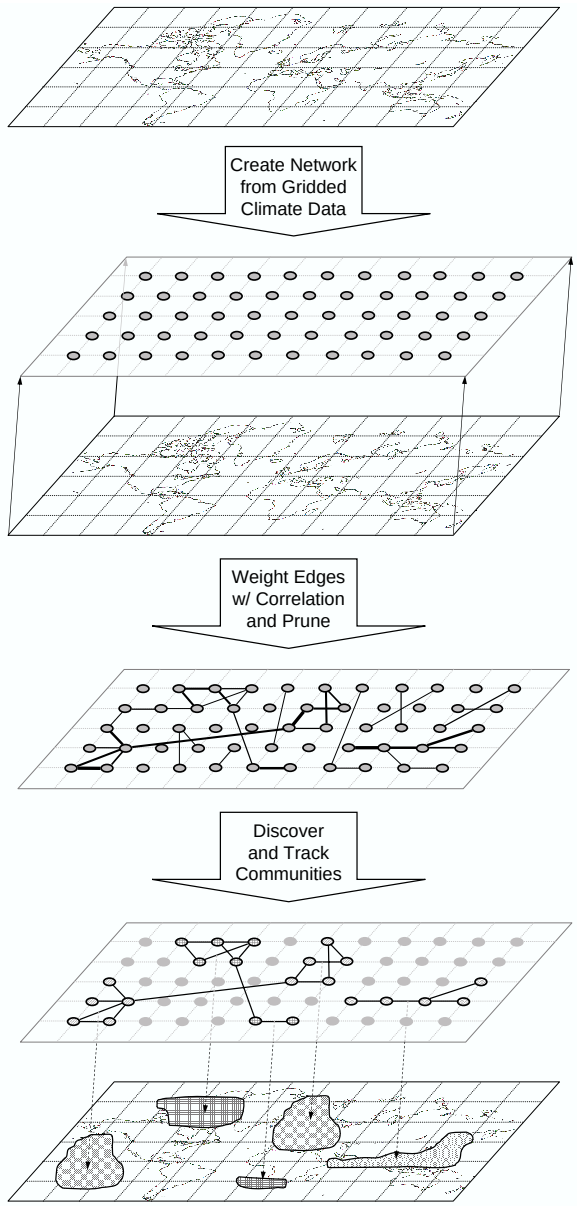



**Figure 3: Step-by-step overview of our workflow.**

Our measure for similarity between two grid cells is then defined as the Euclidean distance in this new $\mathbb{R}^6$ space, so that the *interaction between variables at each location* – as opposed to the behavior of a single variable – defines the strength of the relationship between locations.

## 3.2 From Similarity to Networks

Having defined a similarity measure that maps our four time series corresponding to the four climate variables into Euclidean space, we could apply k-means or a similar clustering method to cluster grid cells into potential climate regions. However, this approach would neither solve the problem of data representation nor would it address the issues mentioned earlier, namely, that the data is noisy, contains clusters of varying densities, and may change with time.

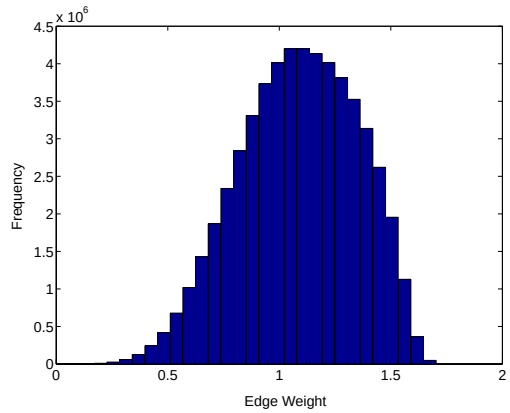



**Figure 4: Distribution of edge weights in the cross correlation-based network for the period 1948-1952.**

Instead, we propose to construct networks from the climate data by dividing the time series into 5-year windows, so that twelve separate networks are available to study changes in structure over time. The network for each window is then constructed as follows. Let each grid cell be a node in the network. For each pair of nodes create an edge and assign it a weight equal to the correlation-based similarity described in section 3.1. This will result in a fully connected network (i.e., one giant clique) consisting of over ten thousand nodes and more than 55 million edges. The concept of a community is naturally absent here, so the next step is to prune away many of the edges in order for structure to emerge.

Of course we want to eliminate edges in a principled manner, and the edge weights allow us to do just that. As illustrated in Figure 4, the histogram of edge weights in the complete network (1948-1952 shown) follows a unimodal distribution – the exact shape is irrelevant here. What *is* important is the presence of only a few edges with very high weight (in the right tail), and it is precisely those strongest edges that define the fundamental structure of the network. Therefore, we prune away 99% of the edges, retaining only the top 1 percentile by weight (this may seem extreme, but it still leaves over a half million edges intact).

After performing this procedure in each window, we obtain twelve climate networks for analysis. In the next section, we briefly describe the community detection process and demonstrate why correlation-based similarity is necessary to identify interesting clusters.

## 3.3 Community Detection in Climate Networks

Given a set of networks constructed from the climate data as described above, we can now use community detection to identify regions of interest. A variety of algorithms with different characteristics have been proposed and applied in a number of settings including social networks [15], protein interaction networks [1], and food webs [2]. Two criteria drove our selection of an appropriate algorithm: (i) due to the relatively high network density it must be computationally efficient, and (ii) it must have the ability to consider weighted networks. Based on these requirements we chose an algorithm called *WalkTrap*, which is grounded in the in-

tuition that random walks in a network are more likely to remain within the same community than to cross community boundaries; for algorithm details see [16]. To our knowledge, this is the first time community detection has been used on networks constructed from spatio-temporal data.

We applied the WalkTrap algorithm to each of the twelve networks using the default parameter of walk length $t = 4$. A sample visualization of the communities $\geq 20$ nodes for the first period is shown in Figure 5(a). Note that they vary widely in both shape and size, and many of them are spatially disjoint. To illustrate the implications of different edge weightings we also constructed a network from air temperature alone, where similarity is defined as the maximum cross correlation ($\pm 6$ months) between the time series for two locations. Figure 5(b) depicts communities $\geq 20$ nodes in this network. While these may be more pleasing to the eye (primarily due to their spatial cohesion), they are also less interesting because they merely show areas where temperatures are similar. In fact, more elementary measures such as annual means, ranges, or the presence/absence of seasons should be sufficient to identify these kinds of patterns. This result is nonetheless encouraging as it demonstrates that the network representation is capable of discerning simple patterns such as univariate climate regions, but our ultimate goal is to discover more complex patterns.

## 3.4 Tracking Communities over Time

Since we are interested in the behavior of communities over time, our last task is to extract only those which can be tracked through several consecutive windows. Community tracking in dynamic networks can itself be a challenging problem [8], but in this application the following method proved sufficient. For each community $C_{t,i}$ labeled $i$ at time step $s$, maximize the quantity

$$\arg\max_{j} |C_{s,i} \cap C_{s+1,j}| \tag{2}$$

s.t.

$$\frac{|C_{s,i} \cap C_{s+1,j}|}{C_{s,i}} > 0.5 \quad and \quad \frac{|C_{s,i} \cap C_{s+1,j}|}{C_{s+1,j}} > 0.5$$

In other words, find the corresponding community labeled $j$ at time $s+1$ with which it shares the most nodes. If less than 50% of the nodes in the community change between time steps, then the community is said to *persist* and we can reasonably assume that there is continuity; otherwise we consider there to be insufficient evidence for tracking and the community is discarded. This process is repeated for all time steps $s = 1, 2, ..., 11$ until all "trackable" communities have been identified.

## 3.5 Summary of Methods and Complexity Analysis

The pseudocode for our methodology is shown in Algorithm 1, divided into its three major stages: computation of cross correlation-based similarities between locations (lines 1-14), systematic pruning of edges (15-23), and community detection and tracking over multiple time periods (24-35). The procedure takes as input a spatio-temporal climate dataset and produces as output the progression of all communities deemed trackable.

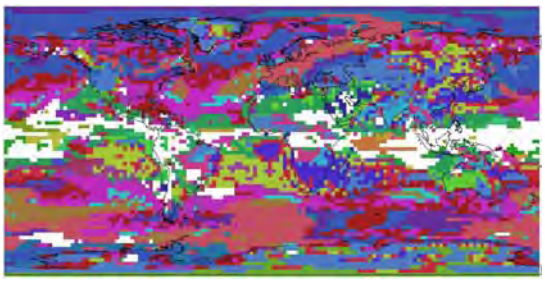
It is apparent from the pseudocode that the computational complexity of the algorithm is quite high. For the first stage, the dominant operation is the nested loop beginning on line 4. Using a simplified notation where $n = lat \times lon$ is the total number of grid cells, this loop has a complexity of $O(n^2v^2t)$, so that the processing requirements increase quadratically with the number of points as well as the number of variables. Given that the complete network contains $O(n^2)$ edges that need to be sorted, the second stage will be $O(n^2 log\ n)$. The third stage of the procedure is bounded by the WalkTrap algorithm, also $O(n^2 log\ n)$ [16]. Therefore, the overall complexity of the end-to-end community detection procedure is $O(n^2v^2t) + O(n^2 log\ n)$.

Note that in practice, a vast majority of the total execution time is spent in the first stage. In fact, the first stage took approximately 1,200 CPU hours (24 hours on 50 machines) to complete, whereas the second and third stages combined only required 2 CPU hours (on a single machine).

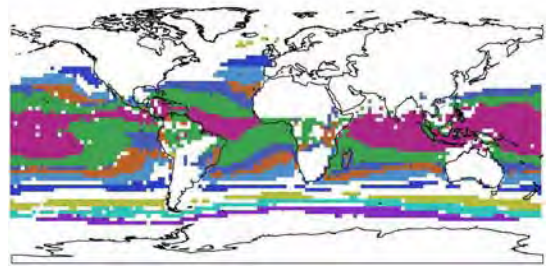
**Algorithm 1** Community Detection in a Climate Network.

**Input:** A dataset $D$ of $lat \times lon$ locations, divided into $k$ time series of length $t$ for each climate variable in $V$ (elements of $D$ are accessed with subscripts $D[x, y, v]$).

1: {Compute Cross Correlation-Based Similarities}
2: **for** each time period $s = 1...k$ **do**
3: initialize graph $G_s = \{\ \}$
4: **for** each location $p$ in $(x_1 = 1...lon, y_1 = 1...lat)$ **do**
5: **for** each location $q$ in $(x_2 = 1...lon, y_2 = 1...lat)$ **do**
6: **for** each $v_1 \in V$ **do**
7: **for** each $v_2 \in V \backslash v_1$ **do**
8: $\mathbb{R}^6_{v_1,v_2} = \underset{-6 \leq d \leq 6}{argmax}\ CCF(D_s[x_1, y_1, v_1], D_s[x_2, y_2, v_2], d)$
9: **end for**
10: **end for**
11: calculate edge weight $w = dist(p, q, \mathbb{R}^6)$
12: add edge $e(p, q, w)$ to $G_s$
13: **end for**
14: **end for**
15: {Network Pruning}
16: sort edges of $G_s$ by weight
17: set pruning threshold $w_{min}$ at $99^{th}$ percentile
18: **for** each edge $e(p, q, w) \in G_s$ **do**
19: **if** $w < w_{min}$ **then**
20: remove $e(p, q, w)$ from $G_s$
21: **end if**
22: **end for**
23: **end for**
24: {Community Detection and Tracking}
25: **for** each time period $s = 1...k$ **do**
26: $C_s = WalkTrap(G_s)$
27: **end for**
28: **for** each time period $s = 1...(k-1)$ **do**
29: **for** each community $i \in C_s$ **do**
30: $overlap = \underset{j}{argmax}\ |C_{s,i} \cap C_{s+1,j}|$
31: **end for**
32: **if** $(overlap/|C_{s,i}| > 0.5)$ and $(overlap/|C_{s+1,j}| > 0.5)$ **then**
33: output $C_{s,i}, Cs + 1, j, overlap$
34: **end if**
35: **end for**



(a) Network weighted with cross correlation-based similarity for the period 1948-1952



(b) Network weighted with correlation of air temperature only for the period 1948-1952

**Figure 5: Comparison of community structure for different weighting methods (best viewed in color).**

# 4. EXPERIMENTAL RESULTS

Here we present several examples of communities identified using the methodology described in Section 3, along with further analysis and potential interpretations based on underlying climate phenomena. For space reasons we limit ourselves to the four communities shown in Figure 7, each covering five consecutive windows (25 years). But before we delve deeper into the discussion, let us first define a measure of density that enables us to determine the relative strength of individual communities at different time steps.

## 4.1 Evaluation Measure: Community Density

The structural properties of clusters, or communities, are an important tool to better characterize them and detect changes over time. One property that is frequently considered is cluster density, and many algorithms implicitly (or even explicitly) measure density as part of the clustering process. Community detection is different in that the density of an individual cluster cannot be measured directly. Therefore, we define an alternate measure to estimate density of communities based on the distribution of edges instead.

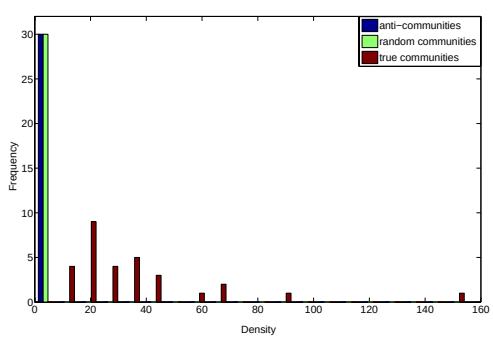



**Figure 6: Density of different types of communities.**

Let $n$ and $e$ be the total number of nodes and edges in the network, respectively; similarly, let $n_i$ denote the number of nodes in community $C_i$ and $e_i$ the number of edges between nodes in this community. The density of community $C_i$ is then defined as the ratio of the number of within-community edges to the expected number of edges based on density of the network as a whole,

$$Density(C_i) = \frac{e_i}{\frac{e}{n(n-1)/2} \times n_i(n_i - 1)/2} \quad (3)$$

We validated this method by comparing the density distribution of 30 "true" communities (identified in our network) with random communities as well as "anti-communities", chosen from different true communities in a round-robin fashion. As shown in Figure 6, the anti-communities have a density less than 1 and the random communities have a density between 1 and 2, while the density of the true communities ranges from 8 to 152. It is generally true that higher density is an indication of more interesting communities, but a domain expert should assist in making this determination.

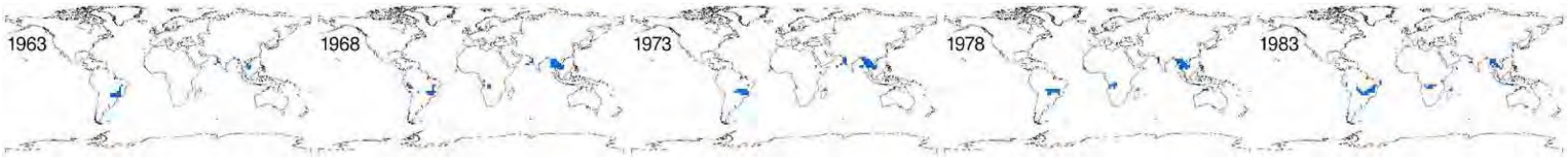
## 4.2 Community 1: South America, Africa, and South-East Asia

During the following discussion, please refer to Figure 1 for information on the major global climate zones. The first community, shown in Figure 7(a), spans the years 1963-1987 and is relatively small, ranging from approximately 25 to 50 nodes. Nonetheless, it consistently covers spatially stable regions on three different continents and is one of the most dense communities overall.

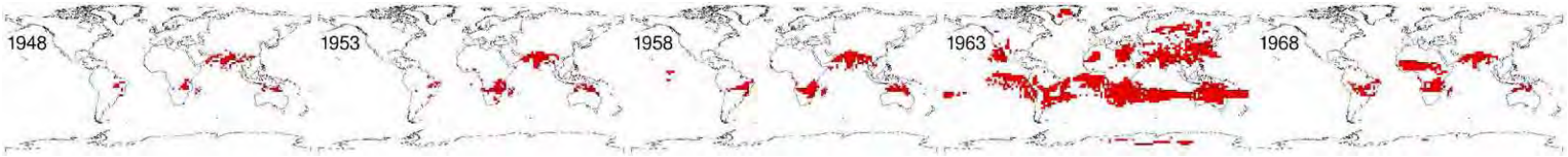
In terms of physical interpretation, the areas included in community 1 belong either to the Tropical Wet-Dry (South America, Africa) or to the Monsoon (South-East Asia) climate zones. Given the subset of variables we considered here, it is likely that the strong inverse relationship (negative correlation) between the hydrological patterns in the two regions – extremely dry conditions versus intense rainfall/monsoons during the summer months – are at least partially responsible for the emergence of this community.

## 4.3 Community 2: South America, Africa, India, and Australia

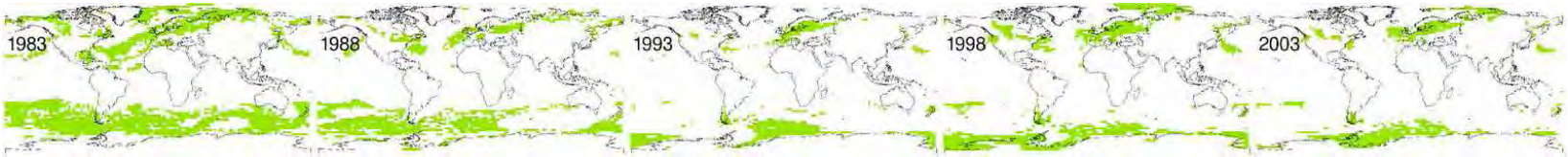
The second community in Figure 7(b) spans the years 1948-1972 and, with the exception of the fourth window (1963), consists of approximately 150 to 220 nodes. Much like the first community, it is composed (primarily) of Tropical Wet-Dry areas and (relatively fewer) Monsoon regions, specifically in Northern India.



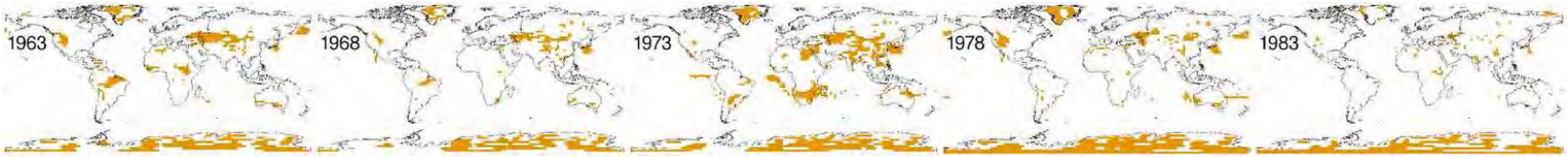

(a) Community 1: South America, Africa, South-East Asia. Density: 78.94 (1963), 44.02 (1968), 41.16 (1973), 86.24 (1978), 58.81 (1983)




(b) Community 2: South America, Africa, India, Australia. Density: 89.84 (1948), 70.38 (1953), 45.30 (1958), 8.85 (1963), 48.90 (1968)




(c) Community 3: Southern Ocean, Canada, Europe. Density: 5.68 (1983), 9.68 (1988), 16.21 (1993), 13.31 (1997), 17.31 (2003)




(d) Community 4: Antarctica, Western US, Greenland, Central Asia. Density: 31.34 (1963), 52.70 (1968), 43.17 (1973), 41.37 (1978), 75.00 (1983)

**Figure 7: Four communities extracted from climate networks, each tracked over five consecutive windows (best viewed in color). Density is given as the ratio of within-community edges actually present relative to the expected number of edges based on the entire network.**

What makes this community interesting is that significant change in structure occurs in 1963, which is visually detectable but also manifests itself as a decrease in density. A closer look at the network reveals that the community merged with a larger community containing a number of locations in a Desert zone, only to separate again in the following period. Since Tropical Wet-Dry differs from the Desert mainly in precipitation, we postulate that the monsoon seasons during this time were unusually weak, thereby becoming more simila and causing the two to temporarily merge. Indeed, there is some indication that the monsoon pattern altered slightly during the 1960s [3, 6]. We cannot attribute the change in structure to this climate phenomenon with certainty, but [20] also linked changes in network structure to climate shifts.

### 4.4 Community 3: Southern Ocean, Canada, and Europe

The third and most recent community, depicted in Figure 7(c), spans the years 1983-2007. It is dominated by the Continental Sub-Arctic climate zone but also contains Mediterranean regions, and decreases in size from approximately 2,200 down to 800 nodes. Once again, the decreasing size is accompanied with a slight increase in density. Given that a majority of the locations in the first time step lie the Southern Ocean, it is possible that the reduction in network size is the result of warming trends observed in these areas. However, a strong relationship remains between the Weddell Sea and much of the European continent as well as parts of North America, prompting the question whether this ocean region might be a source for a climate index and exhibit some predictive capabilities.

### 4.5 Community 4: Antarctica, Western US, Greenland, and Central Asia

This last community, shown in Figure 7(d), also spans the years 1963-1987. It is quite dense, indicating a strong relationship between the locations, but the underlying mechanism is non-obvious in this case. A large number of areas are either covered by ice (Greenland, Antarctica) or located in mountain ranges (Rocky Mountains, Andes, Himalayas), but there are also locations in South America, Central Asia, and Australia that defy both of these categorizations. It is possible that the strong relationship arises from an inverse relationship between temperature and precipitation, but that statement is purely speculative at this point. What we know is that teleconnections *do* exist and, ultimately, such unexplained patterns can help guide the development of new analysis methods.

## 5. DISCUSSION & FUTURE WORK

Here we place this exploratory study in the broader context of climate science. We expand upon known issues relating to data and methodology, point out current capabilities and limitations, and discuss potential extensions.

### 5.1 Data and Variable Selection

The present work examines gridded reanalysis data, the best estimate of a global historical climate record. However, it is worth noting several factors to keep in mind when using this type of data. For one, reanalysis data is composed from multiple heterogeneous sources including satellite, remotely sensed, and in-situ measurements. These raw inputs are combined by fitting a model to the data, which inherently results in some smoothing. In addition, values are interpolated to a regular grid, further reducing variability as well as precision. This was not a concern here as cross correlation measures relatively long-term trends, but it could have serious implications if we were comparing other indicators such as climate extremes, for example.

Likewise, the selection of variables may significantly affect community structure and needs to be considered when interpreting the results. As discussed in Section 2, we chose four variables that are strongly indicative of certain climate phenomena, and we intentionally avoided variables that are known to be problematic (e.g., precipitation) In the future, we plan to investigate the variable selection problem more thoroughly and answer such questions as, How does the community structure change by adding/removing a variable? Can these changes be explained by the presence/absence of that variable? And how to select an optimal subset of variables for a given task?

Lastly, we note that reanalysis products are not the only type of climate data. In some cases actual observations (e.g., thermometer, rain gauge) or at least higher-resolution gridded datasets are available for smaller geographic regions. Climate models represent another viable source of data, and comparing observations with model hindcasts may be one valuable exercise as differences between the two may provide some insights into the model bias.

### 5.2 Similarity Measures

The cross correlation-based similarity between locations proposed here is a rather complex measure, especially in terms of interpreting and relating communities back to the data. In fact, when faced with the problem of clustering climate data there are a host of other measures one might consider first, including averages and ranges, anomalies from long-term means, presence of seasonality, or (auto)correlation in space and time. But much prior work has been done using these measures, especially for univariate data, and the results are well understood. Instead, we went beyond these conventional boundaries to explore an innovative, multivariate measure of correlation. In the future, we intend to investigate other similarity measures in this context. For instance, non-linear relationships are known to exist between climate variables [14], but most correlation measures assume linear dependence. Using the framework presented here, we can substitute a non-linear measure [13] and potentially find very different communities. We envision that a combination of simple (mean, seasonality) and more complex (cross correlation, non-linear dependence) measures could eventually be used to discover a variety of patterns in climate data.

### 5.3 Networks and Community Detection

We want to impress upon the reader once again the specific benefits of our network methodology over traditional clustering approaches such as k-means. First, k-means is known to perform poorly on noisy and high-dimensional data [4]. However, even if the correlation-based similarity were used as a distance measure, k-means would be unable to find certain communities. For example, it may not capture tran-

sitive relationships like "if A is similar to B and B is similar to C, then A is also similar to C" in the same way a network can, and community detection algorithms leverage this information to find more meaningful climate regions. While this paper presents but a few examples, we will further explore the parameter space, different clustering algorithms (e.g., k-means, spectral), varying window sizes and a more extensive evaluation of robustness to changes [11].

### 5.4 Computational Considerations

As discussed in Section 3.5, calculating all-pairs similarity is an expensive operation: computing cross correlation for 10,000 grid cells with twelve 5-year windows of four variables took 1,200 CPU hours. However, each of these dimensions could conceivably grow and drastically increase problem size. There has been a consistent trend towards higher resolution, both for reanalysis data and climate model outputs. The present study used a $2.5^\circ \times 2.5^\circ$ grid, but spacings of $1^\circ$ or less are quickly becoming the norm. Second, reanalysis and model data contain dozens of variables. For this study we relied on a domain expert to select a subset, but using additional variables or, worse yet, exploring a large subspace of them may not be feasible. Finally, it should also be noted that model outputs, and to some extent even observations, are available for much longer time periods.

Given that the complexity of our method is $O(n^2v^2t) + O(n^2 log\ n)$, even relatively small changes in dataset size will have noticeable consequences. For example, if the grid spacing decreases by a factor of two, there will be four times as many cells, which in turn means computational requirements will grow by a factor of 16. Therefore, all the aforementioned issues must be taken into consideration when designing experiments, lest the execution time becomes intractable.

## 6. ACKNOWLEDGMENTS

This research was performed as part of a project titled "Uncertainty Assessment and Reduction for Climate Extremes and Climate Change Impacts", which in turn was funded in FY2009 by the initiative called "Understanding Climate Change Impact: Energy, Carbon, and Water Initiative", within the LDRD Program of the Oak Ridge National Laboratory, managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract DE-AC05-00OR22725. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

## 7. REFERENCES

[1] S. Asur, D. Ucar, S. Parthasarathy. An ensemble framework for clustering protein-protein interaction graphs. *Bioinformatics*, 23(13), 29–40, 2007.

[2] A. Clauset, C. Moore, M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453, 98–101, 2008.

[3] F. Congbin, J. Fletcher. Large signals of climatic variation over the ocean in the asian monsoon region. *Adv. Atmos. Sci.*, 5(4): 389–404, 1988.

[4] L. Ertöz, M. Steinbach, V. Kumar. Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High-Dimensional Data. *SIAM Data Mining*, 47–58, 2003.

[5] R. G. Fovell, M.-Y. C. Fovell. Climate Zones of the Conterminous United States Defined Using Cluster Analysis. *J. Climate*, 6(11): 2103–2135, 1993.

[6] S. Gadgil, J. Srinivasan, R. S. Nanjundiah. On Forecasting the Indian Summer Monsoon. *Curr. Sci. India*, 84(4): 394–403, 2002.

[7] W. W. Hargrove, F. M. Hoffman. Using Multivariate Clustering to Characterize Ecoregion Borders. *Comput. Sci. Eng.*, 1(4): 18–25, 1999.

[8] J. Hopcroft, O. Khan, B. Kulis, et al. Tracking evolving communities in large linked networks. *PNAS*, 101(1): 5249–5253, 2004.

[9] J. E. Janowiak, A. Gruber, C. Kondragunta, et al. A Comparison of NCEP-NCAR Reanalysis Precipitation and GPCP Rain-Gauge Satellite Combined Dataset. *J. Climate*, 11(11): 2960–2979, 1998.

[10] E. Kalnay, M. Kanamitsu, R. Kistler, et al. The NCEP/NCAR 40-Year Reanalysis Project. *B. Am. Meteorol. Soc.*, 77(3): 437–471, 1996.

[11] B. Karrer, E. Levina, M. E. J. Newman. Robustness of community structure in networks. Phys. Rev. E, 77, 046119, 2008.

[12] E. A. Keller. Introduction to Environmental Geology, 3rd ed. Prentice Hall, 2004.

[13] S. Khan, S. Bandyopadhyay, A. R. Ganguly, et al. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Phys. Rev. E*, 76(2), 026209, 2007.

[14] S. Khan, A. R. Ganguly, S. Bandyopadhyay, et al. Nonliear statistics reveals stronger ties between ENSO and the tropical hydrological cycle. *Geophys. Res. Lett.*, 33, L24402, 2006.

[15] M. E. J. Newman, M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 026113, 2004.

[16] P. Pons, M. Latapy. Computing communities in large networks using random walks. *J. Graph Alg. App.*, 10(2): 191–218, 2006.

[17] M. N. M. Sap, A. M. Awan. Finding Spatio-Temporal Patterns in Climate Data Using Clustering. *IEEE Cyberworlds*, 164–171, 2005.

[18] M. Steinbach, P.-N. Tan, V. Kumar, et al. Discovery of Climate Indices Using Clustering. *ACM SIGKDD*, 446–455, 2003.

[19] A. A. Tsonis, K. Swanson. Topology and Predictability of El Niño and La Niña Networks. *Phys. Rev. Lett.*, 228502, 2008.

[20] A. A. Tsonis, K. Swanson, S. Kravtsov. A new dynamical mechanism for major climate shifts. *Geophys. Res. Lett.*, 34, L13705, 2007.

[21] A.A. Tsonis. Introducing Networks in Climate Studies. *Nonlinear Dynamics in Geosciences*, Springer, 2007.

[22] K. Wyrtki. Teleconnections in the Equatorial Pacific Ocean. *Science*, 6(180): 66–68, 1970.

[23] K. Yamasaki, A. Gozolchiani, S. Havlin. Climate Networks around the Globe are Significantly Affected by El Niño. *Phys. Rev. Lett.*, 228501, 2008.

[24] http://cdiac.ornl.gov/climate/indices/indices_table.html

[25] http://www.cgd.ucar.edu/cas/catalog/climind/

[26] http://www.cdc.noaa.gov/data/gridded/data.ncep.reanalysis.html

# A Comparison of SNOTEL and AMSR-E Snow Water Equivalent Datasets in Western U.S. Watersheds

Cody L. Moser
Oubeidillah Aziz
Glenn A. Tootle
University of Tennessee
Dept. of Civil and Environmental Engineering
Knoxville, TN 37996
+1-865-974-7777
cmoser3@utk.edu

Venkat Lakshmi
University of South Carolina
Dept. of Geological Sciences
Columbia, SC 29208
+1-803-777-3552
vlakshmi@geol.sc.edu

Greg Kerr
University of Wyoming Office of Water Programs
Laramie, WY 82070
+1-307-766-6656
rrek@uwyo.edu

## ABSTRACT

It is a consensus among earth scientists that climate change will result in an increased frequency of extreme events (e.g., precipitation, snow). Streamflow forecasts and flood/drought analyses, given this high variability in the climatic driver (snowpack), are vital in the western United States. However, the ability to produce accurate forecasts and analyses is dependent upon the quality (accuracy) of these predictors (snowpack). Current snowpack datasets are based upon in-situ telemetry. Recent satellite deployments offer an alternative remote sensing data source of snowpack. The proposed research will investigate (compare) remote sensing datasets in western U.S. watersheds in which snowpack is the primary driver of streamflow. A comparison is made between snow water equivalent (SWE) data from in-situ snowpack telemetry (SNOTEL) sites and the advanced microwave scanning radiometer – earth observing system (AMSR-E) aboard NASA's Aqua satellite. Principal component techniques and Singular Value Decomposition are applied to determine similarities and differences between the datasets and investigate regional snowpack behaviors. Given the challenges (including costs, operation and maintenance) of deploying SNOTEL stations, the objective of the research is to determine if satellite based remote sensed SWE data provide a comparable option to in-situ datasets. Watersheds investigated include the North Platte River, the Upper Green River, and the Upper Colorado River. The time period analyzed is 2003-2008, due to the recent deployment of the NASA Aqua satellite. Two distinct snow regions were found to behave similarly between both datasets using principal component analysis. Singular Value Decomposition linked both data products with streamflow in the region and found similar behaviors among datasets. However, only 11 of the 84 SNOTEL sites investigated correlated at a significance of 90% or greater with its corresponding AMSR-E cell. Also, when comparing SNOTEL data with the corresponding satellite cell, there was a consistent difference in the magnitude (Snow Water Equivalent) of the datasets. Finally, both datasets were utilized and compared in a statistically based streamflow forecast of several gages.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7…$5.00.

## Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Correlation and regression analysis, multivariate statistics, reliability and life testing, statistical computing.

## General Terms

Management, Measurement, Documentation, Performance, Reliability

## Keywords

SNOTEL, AMSR-E, Snow, Water, Equivalent, SWE

## 1. INTRODUCTION

Snowpack in mountainous regions of the western U.S. provide 50-80% of the water supply in the region [NRCS 2006]. However, the west is being affected by a changed climate more than any other part of the U.S., outside Alaska [NRDC 2008]. Warmer temperatures have major impacts on snowpack in the region. Increased temperatures cause more precipitation in the form of rain, rather than snow. The decrease in snowpack affects many downstream users who expect water in the demand season (April-July). This snowpack acts as a reservoir and is a critical component for summer water supply in arid and semi-arid downstream regions. The hydrologic variability of the west makes it critical to accurately measure the amount of snowpack in headwater regions.

Snow Water Equivalent (SWE) is the amount of water contained within a snowpack. It can be thought of as the depth of water that would result if the entire snowpack was melted instantaneously. Historically, SWE datasets have been collected for over a century. SWE data are important for an array of reasons. Streamflow forecasts along with flood and drought analysis are important uses SWE datasets provide. Accurate measurements of this snowpack are vital in understanding the hydrologic variability in the western United States. Effective management of limited water supplies is a critical component of the sustainability of populations throughout the western United States [Pagano et al. 2004].

The comparison of in-situ SWE with satellite based SWE data is a first step in determining the similarities and differences between the datasets. Three outcomes are possible: (1) the datasets are highly related; (2) the datasets have little similarity or; (3) the

datasets are inconsistent. If satellite-based SWE is highly related to in-situ SWE, steps can be taken to incorporate satellite data in streamflow, flood, and drought forecasts and analyses. Additionally, and if required, new land based climate stations can be effectively positioned, based on conclusive results from satellite data. However, if the datasets have little similarity between them, the reliability and consistency of both datasets comes into question, particularly satellite data. Finally, the datasets may be inconsistent (e.g., highly related in one region and unrelated in another region). This might occur because both datasets have instrument limitations, and the topography as well as the climate variability of the region makes it difficult to accurately measure SWE.

There are advantages and disadvantages associated with each dataset. Cloud and vegetative cover coupled with topography are primary concerns when observing the land surface with satellites. However, cloud and vegetative cover is not an issue with land-based stations. Satellite data provide global coverage, while in-situ stations only provide point coverage. Installation, operation, and maintenance of land-based sensors are costly, but it is also expensive to deploy satellites with advanced instrumentation. Both land-based and satellite-based data are gathered and published in near-real time. This assists water supply forecasters and managers in making timely decisions for efficient allocation of water. Digital land-based sensors that measure SWE were first installed around 1980. Instruments aboard satellites that measure SWE were launched in 2002.

Although the advanced microwave scanning radiometer – earth observing system (AMSR-E) datasets are relatively new (i.e., short period of record), there are numerous applications these datasets can be used for, and a large amount of research has incorporated these data into studies. The value of AMSR-E soil moisture data was shown in [Bindlish et al. 2009], and flood forecasts were improved with success. The work of [Reichle et al. 2007] compared AMSR-E and the Scanning Multichannel Microwave Radiometer (SMMR) soil moisture datasets. Spatial and temporal resolution of sea surface temperatures using AMSR-E data was dramatically improved in [Reynolds et al. 2007]. The research of [Narayan and Lakshmi 2008] fused soil moisture estimates from the AMSR-E instrument with backscatter data and produced a higher spatial resolution of soil moisture variability. Additional research has utilized AMSR-E sea ice data and moderate resolution imaging spectroradiometer (MODIS) data.

The objective and contribution of this research is to compare land based SWE point data and satellite based SWE spatial data within the western United States. This will provide comprehensive and valuable information about the validity of each dataset. While satellite products have been compared to other satellite products, few (or none) have been carried out between satellite observations and land based snow telemetry (SNOTEL) data. This is most likely attributed to the lack of an intense spatial coverage of SNOTEL sites given the typical grid size of satellite data (i.e., there is generally one SNOTEL gauge in a grid of 25 $km^2$). While this is a challenge/limitation, there is an interest in comparing both the spatial and temporal variability and the magnitude of satellite data to in-situ data. Ultimately, one would hypothesize that the use of satellite generated spatial SWE data will result in an improved representation of basin wide hydrology compared to in-situ data. The incorporation of satellite data in hydrologic-based forecasts is a natural 'next step' as displayed in the movement from snow course (manually measured snowpack data) to SNOTEL (snowpack data provided by remote sensors). Research and analysis of global hydrology may be based strictly on satellite data in the near future. This would increase the efficiency as well as decrease the costs of collecting hydro climatic data.

# 2. WATERSHED DESCRIPTIONS

Three critical western U.S. watersheds that receive extensive snowfall are included within the scope of this research. Basins include the North Platte River, Upper Green River, and Upper Colorado River. The headwaters of the North Platte River are located in northern Colorado. The watershed is bound on all sides by mountain ranges: the Rawah and Never Summer ranges to the east, Rabbit Ears range to the south, and Park Range to the west [Daniels 2007]. From northern Colorado, the North Platte River continues north into Wyoming, and then flows east into Nebraska. Understanding the hydrology of the North Platte River headwaters is critically important for water resource planning in the Rocky Mountains and Great Plains regions [Daniels 2007].

The Upper Green River originates in western Wyoming and is the primary tributary of the Colorado River. The headwaters begin in the Wind River Mountains. From Wyoming, the Green River flows south into Utah and Colorado, where it flows into the Colorado River.

The Upper Colorado River Basin originates in the mountains of central Colorado. The Colorado River flows south into Nevada and Arizona. The Colorado River is the major source of water for the driest part of the country. Upwards of 30 million Americans across seven states now depend on it for agricultural, municipal, industrial, and hydroelectric needs – and the basin is among the fastest growing areas in the country [NRDC 2008].

# 3. DATA SOURCES

## 3.1 SNOTEL (NRCS)

Snow Water Equivalent (SWE) was first measured manually using a snow course. A snow course is a permanent site where manual measurements of snow depth and SWE are taken by trained observers. Generally, the courses were about 1,000 feet long and situated in small meadows protected from the wind. Technological advances have allowed the National Resource Conservation Service (NRCS) to use snowpack telemetry, or SNOTEL sites. The NRCS currently operates and maintains over 750 automated SNOTEL stations in 13 western states. The data as well as related reports and forecasts are made available in near real-time. SNOTEL sites are generally located in remote, mountainous watersheds and provide SWE data via a pressure-sensing snow pillow. Eighty-four SNOTEL sites across 4 states are used in this study. The distribution is as follows: Wyoming (36), Colorado (35), Idaho (7), and Utah (6). Snowpack on the first day of April is a good indicator of the water content in the maximum seasonal snowpack and for the water supply for coming months [Woodhouse 2003]; and is used in this study. The location of SNOTEL sites included in this research can be seen in Figure 1.

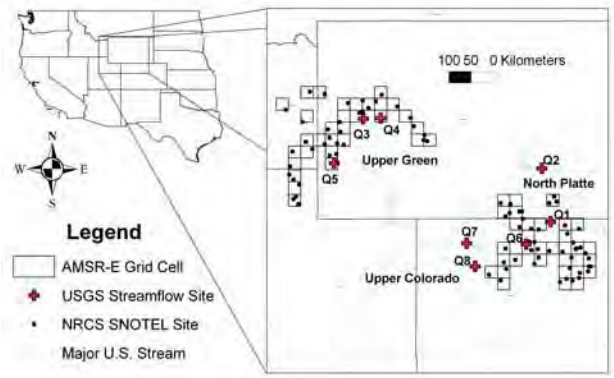



**Figure 1: Location map showing all data used**

## 3.2 AMSR-E (NASA)

The advanced microwave scanning radiometer – earth observing system (AMSR-E) instrument aboard NASA's Aqua satellite was launched in May of 2002. The instrument measures polarized microwave radiation horizontally and vertically. Land, oceanic, and atmospheric parameters are measured for the investigation of global and water energy cycles. Data is mapped globally into 25 $km^2$ Equal-Area Scalable Earth (EASE) Grids. This study breaks the gridded data down further into 0.25 decimal degrees longitude by 0.25 decimal degrees latitude cells. Available snow water equivalent datasets include daily, 5-day maximum and monthly average. The five-day maximum SWE value (27March – 01April) from [Kelly et al. 2004] is used in this research. The five-day maximum was chosen to eliminate the possibility of missing data that results from cloud cover. It should be noted that AMSR-E data experiences numerous validation stages. Datasets used within this work are in the transitional validation stage. The transitional validation stage is the period between beta and validated. Figure 1 shows the AMSR-E grid cells used in this study.

## 3.3 Streamflow (USGS)

Within the U.S., the United States Geological Survey (USGS) collects surface-water data that describe stream levels, streamflow (discharge), reservoir and lake levels, surface-water quality, and rainfall. Automatic recorders and manual measurements collect data. [Slack and Landwehr 1992] identified a Hydro-Climatic Data Network (HCDN) of stream gages as being relatively free of significant human influences and, therefore, appropriate for climate studies. Streamflow measurements from eight of these gages are incorporated in this study - USGS 06620000 (Q1), USGS 06635000 (Q2), USGS 09188500 (Q3), USGS 09196500 (Q4), USGS 09223000 (Q5), USGS 09239500 (Q6), USGS 09251000 (Q7), and USGS 09304500 (Q8). Q1 and Q2 are located in the North Platte River Basin; Q1 on the North Platte River and Q2 on the Medicine Bow River. Q3, Q4, and Q5 are located in the Upper Green River Basin. Q3 is on the Green River, Q4 is on Pine Creek, and Q5 is on Hams Fork. Q6, Q7, and Q8 are within the Upper Colorado River Basin. Q6 and Q7 are on the Yampa River while Q8 is on the White River. The average streamflow in the region for the months of April-July (AMJJ) from 2003 to 2008 is the value of interest that water managers are most interested in, and is used in this study. Table 1 contains physical characteristics for the streamflow stations utilized in this study, and the locations of these stations are shown in Figure 1.

**Table 1: Streamflow gage descriptions**

| Station ID | Drainage Area ($km^2$) | Elevation (m) | Symbol |
|---|---|---|---|
| 06620000 | 3,706 | 2,380 | Q1 |
| 06635000 | 6,055 | 1,955 | Q2 |
| 09188500 | 1,212 | 2,276 | Q3 |
| 09196500 | 196 | 2,270 | Q4 |
| 09223000 | 331 | 2,272 | Q5 |
| 09239500 | 1,471 | 2,040 | Q6 |
| 09251000 | 8,831 | 1,798 | Q7 |
| 09304500 | 1,955 | 1,920 | Q8 |

# 4. METHODS

## 4.1 Principal Components Analysis

Principal Components Analysis (PCA) using a varimax rotation procedure is used to determine regions in which both datasets may behave similarly. PCA is a widely used technique in meteorology and climatology [Baeriswyl and Rebetez 1997]; and can be used to reduce the size of datasets without losing information. PCA is a statistical technique that restructures a set of intercorrelated variables into an equal number of uncorrelated variables. Each new variable (principal component) is a different linear combination of all the original variables.

First, the eigenvalues and eigenvectors of the correlation matrix of the time series and loading matrix are computed. The loading matrix represents the correlation of the original variables (SWE) with the principal components. Based on the results of the loading matrix, the stations that are highly correlated with the particular principal components can be identified. Factor loadings are investigated after a varimax rotation is performed on each dataset. Use of a varimax rotation results in easier interpretation of the principal component factor loadings, and a similar technique was applied in [Timilsena and Piechota 2008]. Factor loadings are on a scale of -1.0 to 1.0; higher values (positive or negative) signify factor representation. Loadings greater than +0.90 and less than -0.90 are retained in this research. The factor loadings that are retained are investigated to identify regions where SNOTEL sites and AMSR-E grid cells may behave similarly.

## 4.2 Singular Value Decomposition

Singular Value Decomposition (SVD) is a powerful statistical tool for identifying coupled relationships between two, spatial-temporal fields [Tootle et al. 2008]. [Bretherton et al. 1992] evaluated several statistical methods and concluded SVD was simple to perform and preferable for general use. [Wallace et al. 1992] determined that SVD isolates the most important modes of variability. While [Bretherton et al. 1992] provides a detailed discussion of the theory of SVD, a brief description of SVD, as applied in the current research is provided.

Initially, a matrix of SWE and a matrix of streamflow are developed. The time dimension of each matrix (i.e., 6 years) must be equal while the spatial component (i.e., number of SNOTEL sites, AMSR-E grid cells, or streamflow stations) can vary in

dimension. Standardized anomalies of the matrices are used. A cross-covariance matrix X is then computed for the two spatial, temporal matrices by multiplying the SWE matrix with the inverse of the streamflow matrix, and then dividing by the number of years. SVD is applied to the cross-covariance matrix and physical information about the relationship between the two fields (snowpack and streamflow) is obtained.

In this study, the data are broken down into 2 geographic regions (northwest and southeast) and SVD between snowpack and streamflow is performed separately in each region. The northwest region contains the Upper Green River Basin, and the North Platte and Upper Colorado River Basins are within the southeast region. The northwest region contains 38 SNOTEL stations and their corresponding AMSR-E grid cells. Three streamflow stations (Q3, Q4, and Q5) are also located in the northwest region. The southeast contains the remaining 48 SNOTEL/AMSR-E sites along with Q1, Q2, Q6, Q7, and Q8. A significance level of 90% is used. Figure 2 shows the relationship between April 1st snowpack and streamflow in the Upper Green Basin. A single SNOTEL station and corresponding AMSR-E grid cell are plotted against streamflow (Q3) from 2003-2008. Similar relationships are found in surrounding regions.

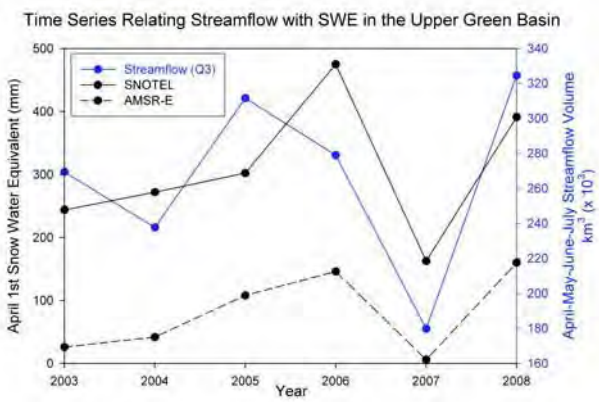



**Figure 2: The relationship between April 1st snowpack and streamflow in the Upper Green Basin**

## 4.3 Application – Streamflow Forecasting

The objective of streamflow forecasting is to accurately predict the volume of water during high demand season (April-July in this region). Accurate forecasts of seasonal streamflow volumes assist a broad array of natural resource decision makers, and Water Supply Outlooks (WSOs) are currently issued jointly by the National Resource Conservation Service (NRCS), the National Weather Service (NWS), and local cooperating agencies [Pagano et al. 2004].

Three USGS streamflow stations are forecasted in this study: Q1 in the North Platte River Basin, Q4 in the Upper Green River Basin, and Q8 in the Upper Colorado River Basin. Two predictor screening steps are performed to find appropriate predictors. First, SNOTEL stations and AMSR-E cells that are 95% significant (n = 6, r = 0.81) with streamflow in the region are determined. Additionally, a search radius is applied. While [Dressler et al. 2006] used a 200 km radius, this study applies a 100 km radius outwards from the streamflow station being forecasted. Therefore, all SNOTEL stations and AMSR-E grid cells that are at least 95% significant and within a 100 km radius of the streamflow station being forecasted are used as initial predictors in each forecast model.

NRCS forecasters rely on a statistical principal component regression technique to predict future streamflow using information about current SWE, precipitation, base flow, and climate indices [Garen 1992]. The NWS is increasingly engaged in dynamic simulation of streamflow by initializing a conceptual hydrologic model with current soil moisture and snowpack conditions [Day 1985]. However, the work presented in this study only incorporates SWE from the two data sources being investigated (SNOTEL & AMSR-E) as predictors for comparison purposes. The most satisfactory and statistically rigorous way to deal with intercorrelation is the use of principal components regression [Garen 1992]; and it is the method used to forecast streamflow in this study. Garen's 1992 method for selecting the principal components to include into the forecast model is followed and applied in this research. The method Garen follows is a three step process: (1) components are added to the model one at a time in sequence, beginning with the one having the largest eigenvalue and progressing in order of decreasing eigenvalue; (2) when the first component with a nonsignificant regression coefficient is found, the components retained are the ones in sequence up to, but not including, the nonsignificant one and; (3) regression coefficients must have the same algebraic sign as their correlations with the dependant variable.

Forecast accuracy of each model (SNOTEL vs. AMSR-E) is evaluated using the $R^2$ – predicted ($R^2_{pred}$) statistic. $R^2_{pred}$ is calculated from the PRESS statistic. PRESS is based upon a leave-one-out cross-validation in which a single year or observation is removed when fitting the model. As a result, the prediction errors are independent of the predicted value at the removed observation (Garen 1992). For selecting a model when the primary interest is in prediction (forecasting), the model with the smaller PRESS (higher $R^2_{pred}$) is preferable [Montgomery et al. 2006]. The idea is that if an area has poor spatial coverage of SNOTEL/weather stations, AMSR-E SWE data collected from satellites may help improve overall forecast accuracy. For the research purposes of this paper (i.e., the comparison of 2 datasets) regions were chosen that had intensive coverage of land-based stations. When an area does not have this luxury, the use of satellite data to forecast streamflow presents an alternative option to identify predictor variables.

# 5. RESULTS

## 5.1 Principal Components Analysis (PCA)

PCA applying a varimax rotation identified two regions in which both datasets behave similarly (see Figure 3). Region 1 is in the Upper Snake/Upper Green River Basins. The first principal component factor (Region 1) retained 17 SNOTEL sites and eight AMSR-E grid cells. Three of the 17 SNOTEL sites retained in factor one showed up outside of the region (i.e., anomalies). There is also one AMSR-E grid cell located outside of Region 1. Region 2 is in the Upper Colorado/North Platte River Basins. The second principal component factor (Region 2) retained seven SNOTEL sites and 17 AMSR-E grid cells. Anomalies were not present in Region 2. However, the spread of AMSR-E grid cells in Region 2 is greater than that of SNOTEL sites. This suggests that AMSR-E SWE data have more inconsistency compared to SNOTEL sites in the Upper Colorado/North Platte region. The datasets agree more in Region 1 than in Region 2 as shown by the

number of AMSR-E grid cells that encompass SNOTEL sites (see Figure 3).

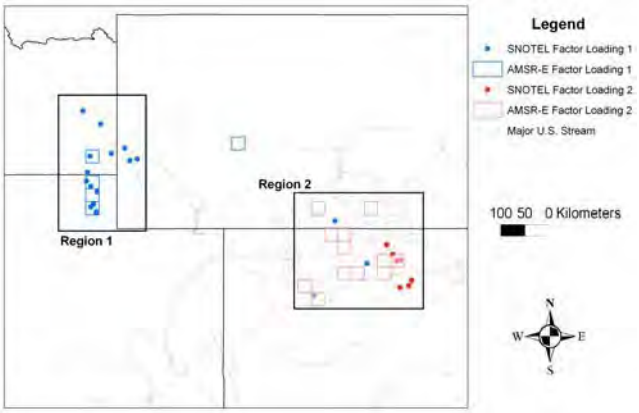



**Figure 3: Map showing the 2 regions PCA identified**

## 5.2 Singular Value Decomposition (SVD)

The SVD results for both SWE datasets with streamflow in each region (northwest and southeast) are shown in Figures 4a (SNOTEL) and 4b (AMSR-E). For the northwest region, SVD found 14 SNOTEL sites that were significant with 2 of the 3 streamflow stations in the region. SVD produced no significant SNOTEL stations with Q5. Only Q3 was found to have a significant relationship with AMSR-E grid cells in the northwest region. Four of the fourteen SNOTEL sites that were significant had a significant corresponding AMSRE-E grid cell with streamflow in the region.

In the southeast region, SNOTEL was found to be significant with all 5 streamflow stations (Q1, Q2, Q6, Q7, and Q8). AMSR-E was found to be significant with all of the streamflow stations except Q1. There were 22 significant SNOTEL sites and 18 significant AMSR-E cells. Of the 18 significant SNOTEL sites, 9 of them had a corresponding AMSR-E grid cell that was also significant with streamflow in the region.

The results from SVD are encouraging and suggest there are similar relationships between both SWE datasets and streamflow in both regions.

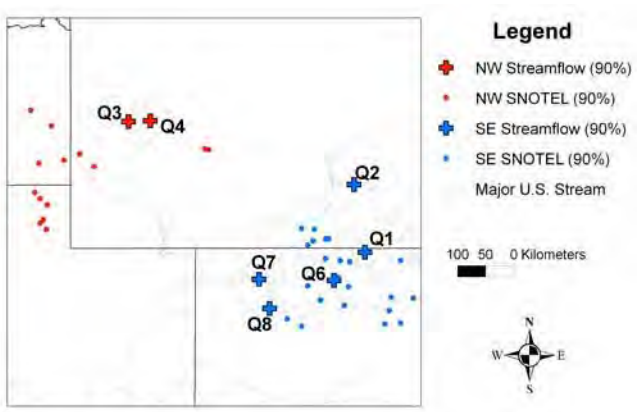



**Figure 4 (a): SVD SNOTEL results**

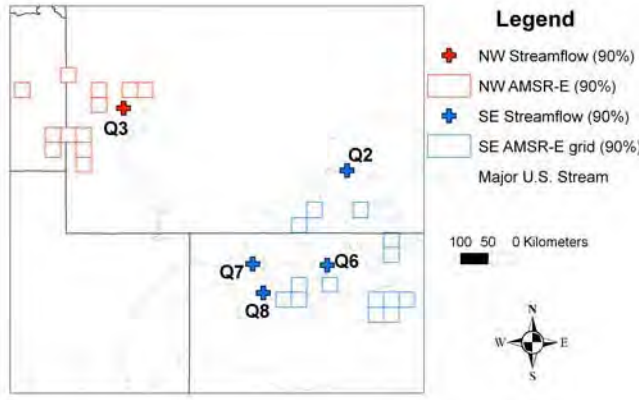



**Figure 4 (b): SVD AMSR-E results**

## 5.3 Streamflow Forecasting

Predictor screening resulted in a comparable number of predictors between SNOTEL sites and AMSR-E grid cells that were included in each forecast model. For Q1 forecasts, there were 6 predictors in the SNOTEL model and 5 in the AMSR-E model. Q4 had 3 SNOTEL predictors and 5 AMSR-E predictors. Q8 models also included 3 SNOTEL sites and 5 AMSR-E grid cells after the predictor screening process.

**Table 2: Forecast statistics**

| | Q1 - North Platte | |
|---|---|---|
| | SNOTEL | AMSR-E |
| $R^2$ | 0.83 | 0.90 |
| $R^2_{adjusted}$ | 0.79 | 0.88 |
| $R^2_{predicted}$ | 0.71 | 0.79 |
| PRESS ($10^3$ $km^3$) | 15.8 | 11.0 |
| Durbin-Watson | 3.0 | 1.5 |
| | Q4 - Upper Green | |
| | SNOTEL | AMSR-E |
| $R^2$ | 0.89 | 0.97 |
| $R^2_{adjusted}$ | 0.86 | 0.97 |
| $R^2_{predicted}$ | 0.41 | 0.90 |
| PRESS ($10^3$ $km^3$) | 0.6 | 0.1 |
| Durbin-Watson | 3.3 | 1.4 |
| | Q8 - Upper Colorado | |
| | SNOTEL | AMSR-E |
| $R^2$ | 0.98 | 0.95 |
| $R^2_{adjusted}$ | 0.98 | 0.93 |
| $R^2_{predicted}$ | 0.96 | 0.91 |
| PRESS ($10^3$ $km^3$) | 1.1 | 2.6 |
| Durbin-Watson | 1.8 | 0.9 |

After performing PCA and following Garen's procedure, one principal component was significant and had the same sign as the streamflow station being forecasted for each model (SNOTEL & AMSR-E). Forecast accuracy was determined using the $R^2$ - predicted statistic. The forecast accuracy for Q1 and Q8 are comparable for each model. For Q1, the SNOTEL model produced an $R^2_{pred}$ of 0.71 while the AMSR-E model's $R^2_{pred}$ is 0.79. For Q8, the SNOTEL model's $R^2_{pred}$ is 0.96, compared to 0.91 for AMSR-E. For Q4, the AMSR-E model produced more accurate forecasts ($R^2_{pred}$ = 0.90) compared to 0.41 for the SNOTEL model. Statistical parameters determined for all forecast models are provided in Table 2. However, because forecasting is not the primary purpose of this paper, the discussion of all statistical parameters is not provided.

# 6. ADDITIONAL ANALYSIS

## 6.1 Correlation

Correlation (r) values between SNOTEL SWE recorded on the ground and AMSR-E SWE recorded by satellite were determined. This was accomplished by finding the AMSR-E grid cell that encompassed each SNOTEL station. A 90% significance level was selected. This step investigates the similarities and differences of each dataset. Furthermore, sites that are found to be significant are investigated in more depth (i.e., elevation) to determine possible reasons for similarities and differences between datasets.

Correlation resulted in only 11 of the 84 (13%) SNOTEL sites to be significant with its corresponding AMSR-E SWE grid cell at 90%. The average elevation of all 84 SNOTEL sites used is 2,737 meters. In the 11 cases that had 90% significance, the average elevation was 2,471 (m), while the remaining 73 SNOTEL sites had an average elevation of 2,777 (m). Also, there were no significant sites located above 3,048 meters (10,000 feet) while 18 were present in the overall sample. These results suggest that elevation has an effect on satellite obtained SWE from the AMSR-E instrument.

## 6.2 Magnitude Differences

The AMSR-E SWE data have a range of 0-480 mm (0-19 inches). In many instances, SNOTEL sites record SWE values that exceed this range within the region. For the April 1st SWE data analyzed in this study, and for at least one year from 2003-2008, 47 of the 84 (56%) SNOTEL sites included in this study experienced SWE values that exceeded the maximum AMSR-E value (>480 mm). However, there were no cases in which the AMSR-E recorded the maximum value of 480 mm. Figure 5 provides a box plot of SWE values for the 11 SNOTEL sites and AMSR-E grid cells that were found to be 90% significant from correlation. It is clear from Figure 4 that there is a significant magnitude difference between SNOTEL and AMSR-E SWE values. Additionally, there is more variability in the SNOTEL dataset.

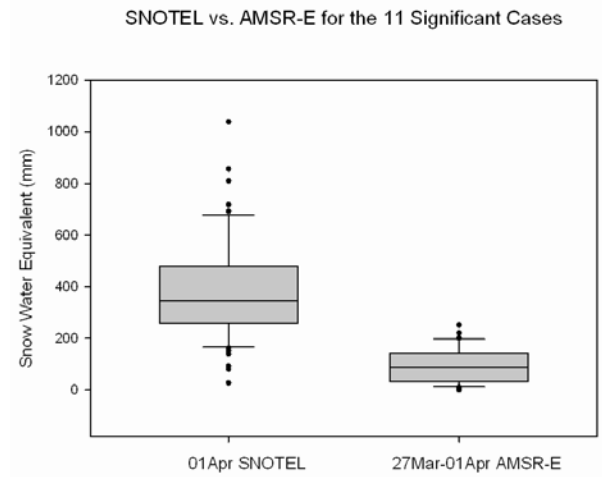



**Figure 5: Box plot showing the magnitude difference between SNOTEL and AMSR-E SWE values**

Magnitude issues limit the use of AMSR-E SWE to a narrow list of applications. In many statistical models, the magnitude of the predictor variables is irrelevant (i.e., magnitude does not affect the results), and this dataset would be sufficient to use for analysis (i.e., streamflow forecasting). However, incorporating AMSR-E into a physical model would result in an inaccurate basin-wide hydrologic representation, and is therefore not recommended without the use of correction factors. The development of correction factors to apply to AMSR-E SWE values is needed because this would provide a basis for incorporating AMSR-E gridded satellite SWE into physically based climate models. The work of [Durand and Margulis 2007] has investigated the correction of first-order errors in SWE estimates from the AMSR-E instrument.

# 7. CONCLUSION

A thorough comparison has been made between SNOTEL and AMSR-E snow water equivalent datasets. Statistical techniques applied included Principal Components Analysis, Singular Value Decomposition, and correlation. PCA utilizing a varimax rotation identified two distinct regions in which both datasets are consistent and compare well with one another. Both SWE datasets were successfully linked to streamflow in the region utilizing SVD. The incorporation of both SWE products was applied to forecast regional streamflow. Forecast results were comparable for 2 of the 3 streamflow stations (North Platte and Upper Colorado); however, the AMSR-E SWE dataset produced a more skillful forecast compared to SNOTEL for the Upper Green station. Additional analysis using correlation found that the AMSR-E dataset is inconsistent with land-based SNOTEL values in the region and there are significant magnitude differences between datasets. SWE from NASA's Aqua satellite was found to be sufficient to use in statistically based forecast models in which magnitude did not affect results. Given the projections of climate change (e.g., increased temperatures and decreased snowpack), the ability of satellites to capture important hydrologic parameters such as snow water equivalent is vital in many applications. Incorporating additional satellite parameters such as soil moisture and vegetative cover, in addition to SWE, may be in the near future for next generation forecasters.

## 8. ACKNOWLEDGMENTS

This research is sponsored by the Wyoming Water Development Commission, the USGS/State of Wyoming Water Research Program, the University of Wyoming Office of Water Programs, and the University of Tennessee. Additional funding provided by the University of Tennessee and the Oak Ridge National Laboratory JDRD program.

## 9. REFERENCES

[1] BAERISWYL, P.A., AND REBETEZ, M. 1997. Regionalization of precipitation in Switzerland by means of principal component analysis. *Theoretical and Applied Climatology* 58, 31–41.

[2] BINDLISH, R., CROW, W.T., AND JACKSON, T.J. 2009. Role of passive microwave remote sensing in improving flood forecasts. *IEEE Geoscience and Remote Sensing Letters* 6(1), 112-116.

[3] BRETHERTON, C.S., SMITH, C., AND WALLACE, J.M. 1992. An intercomparison of methods for finding coupled patterns in climate data. *Journal of Climate* 5, 541–560.

[4] DANIELS, J.M. 2007. Flood hydrology of the North Platte River headwaters in relation to precipitation variability. *Journal of Hydrology* 344(1-2), 70-81.

[5] DAY, G.N. 1985. Extended streamflow forecasting using NWSRFS. *Journal of Water Resources Planning and Management* 111, 157–170.

[6] DURAND, M., AND MARGULIS, S.A. 2007. Correcting first-order errors in snow water equivalent estimates using a multifrequency, multiscale radiometric data assimilation scheme. *Journal of Geophysical Research* 112, D13121, doi:10.1029/2006JD008067.

[7] GAREN, D.C. 1992. Improved techniques in regression-based streamflow volume forecasting. *Journal of Water Resources Planning and Management* 118, 654–670.

[8] KELLY, R.E.J., CHANG, A.T.C., FOSTER, J.L., AND TEDESCO, M. 2004. Updated every five days. AMSR-E/Aqua 5-day 3 Global Snow Water Equivalent EASE-Grids V002, [2003-2008]. Boulder, Colorado USA: National Snow and Ice Data Center. Digital media.

[9] MONTGOMERY, D.C., PECK, E.A., VINING, G.G., AND MYERS, R.H. 2006. *Introduction to linear regression analysis* (4th edition). John Wiley and Sons, Hoboken, NJ, 141-142.

[10] NARAYAN, U., AND LAKSHMI, V. 2008. Characterizing subpixel variability of low resolution radiometer derived soil moisture using high resolution radar data. *Water Resources Research* 44, W06425, doi:10.1029/2006WR005817.

[11] NATURAL RESOURCES CONSERVERATION SERVICE. 2006. Snow surveys and water supply forecasting. *USDA Agriculture Information Bulletin* 536. http://www.wcc.nrcs.usda.gov/factpub/aib536.html.

[12] NATURAL RESOURCES DEFENSE COUNCIL. 2008. Hotter and drier: the West's changed climate. http://www.nrdc.org/globalWarming/west/contents.asp.

[13] PAGANO, T.C., GAREN, D.C., AND SOROOSHIAN, S. 2004. Evaluation of official Western US seasonal water supply outlooks, 1922-2002. *Journal of Hydrometeorology* 5(5), 683–698.

[14] REICHLE, R.H., KOSTER, R.D., LIU, P., MAHANAMA, S.P.P., NJOKU, E.G., AND OWE, M. 2007. Comparison and assimilation of global soil moisture retrievals from the Advanced Microwave Scanning Radiometer for the Earth Observing System (AMSR-E) and the Scanning Multichannel Microwave Radiometer (SMMR). *Journal of Geophysical Research* 112, D09108, doi:10.1029/2006JD008033.

[15] REYNOLDS, R.W., SMITH, T.M., LIU, C., CHELTON, D.B., CASEY, K.S., AND SCHLAX, M.G. 2007. Daily high resolution blended analyses for sea surface temperature. *Journal of Climate* 20(22), 5473-5496

[16] SLACK, J.R., AND LANDWEHR, J.M. 1992. Hydro-Climatic Data Network: A U.S. Geological Survey streamflow data set for the United States for the study of climate variations, 1874–1988. USGS Open-File Report 92-129, 193.

[17] TOOTLE, G.A., PIECHOTA, T.C., AND GUTIERREZ, F. 2008. The relationships between Pacific and Atlantic Ocean sea surface temperatures and Colombian streamflow variability. *Journal of Hydrology* 349(3-4), 268-276.

[18] WALLACE, J.M., GUTZLER, D.S., AND BRETHERTON, C.S. 1992. Singular value decomposition of wintertime sea surface temperature and 500-mb height anomalies. *Journal of Climate* 5, 561–576.

[19] WOODHOUSE, C.A. 2003. A 431-yr reconstruction of western Colorado snowpack from tree rings. *Journal of Climate* 16, 1551–1561.

# EDISKCO: Energy Efficient Distributed In-Sensor-Network K-center Clustering with Outliers

Marwan Hassani Emmanuel Müller Thomas Seidl
Data management and exploration group
RWTH Aachen University, Germany
{hassani, mueller, seidl}@cs.rwth-aachen.de

## ABSTRACT

Clustering is an established data mining technique for grouping objects based on similarity. For sensor networks one aims at grouping sensor measurements in groups of similar measurements. As sensor networks have limited resources in terms of available memory and energy, a major task sensor clustering is efficient computation on sensor nodes. As a dominating energy consuming task, communication has to be reduced for a better energy efficiency. Considering memory, one has to reduce the amount of stored information on each sensor node.

For in-network clustering, $k$-center based approaches provide $k$ representatives out of the collected sensor measurements. We propose EDISKCO, an outlier aware incremental method for efficient detection of $k$-center clusters. Our novel approach is energy aware and reduces amount of required transmissions while producing high quality clustering results. In thorough experiments on synthetic and real world data sets, we show that our approach outperforms a competing technique in both clustering quality and energy efficiency. Thus, we achieve overall significantly better life times of our sensor networks.

## 1. INTRODUCTION

Clustering is a data mining task for summarizing data such that similar objects are grouped together while dissimilar ones are separated. In the special case of sensor networks, clustering aims at detection of similar sensor measurements. By detection of $k$ representative measurements in $k$-center clustering one ensures good clustering quality if each representative is assigned to only very similar measurements. In addition to cluster quality, one aims at an efficient cluster computation. For sensor nodes, resources are limited. Especially, available energy and memory pose restrictions on the clustering algorithms.

The distinguishing features of clustering streaming sensors act as future paths to research in this area, some of these are: (a) Concerning efficiency issues, sensors are limited in terms of resources, minimizing the consumption of these (mainly energy) is a major requirement to achieve a high lifetime, (b) A compact representation of both the data and the generated model is needed to enable fast and efficient transmission and access from mobile and embedded devices, (c) The final goal is to infer a global clustering structure of all relevant sensors; hence, approximate algorithms should be considered to prevent global data transmission [8].

In this paper, we propose a novel energy efficient $k$-center clustering approach. We propose an incremental algorithm on each sensor node that computes the $k$ representatives. For compact representation of sensor measurements we address outlier aware clustering, excluding deviating objects from the clustering structure. Our processing enhances clustering quality, specially for sensor measurements where noisy data is gathered.

For incremental adaptation of the cluster representatives we use a reclustering phase. However, as each reclustering causes heavy communication costs, our approach aims at reducing such reclusterings, and thus enhances energy efficiency. In addition, we aim at ensuring high clustering quality even with less reclustering operations.

A motivating example for having an energy-aware $k$-center clustering with outliers: Let $m$ distributed sensor nodes over different parts in a certain area in the jungle, and the target is to measure the existence of certain $k$ kinds of wild animals in this area using a combination of sound and vibration measurements. Each sensor node collects readings in its sensing range and performs $k$ clustering over it reflecting its most $k$ existing kinds of animals in its range. A combination of the results will be done in the base station to make a global knowledge about the existence of these kinds in the whole area. The existence of *outlier* animals not expected to be in this area passing over is very normal, which highlights the need to consider outliers. In addition, in such scenarios sensor nodes will be unattached for a long time or even not at all which highlights the need of having energy-aware applications running on it to prolong its battery lifetime. A slightly different scenario to this one is used in [17].

### 1.1 Challenges and Contribution

Many requirements arise when designing an energy aware in-sensor-network $k$-center stream clustering algorithm that considers outliers; some of these requirements are inherited

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09,* June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00

from different areas, the main requirements can be summarized in the following five aspects:

- **Single Passing, Storage awareness:** Due to the limited processing and storage resources in the sensor node, the clustering algorithm must perform only a single pass over the incoming data stream and storage must be independent on $n$ the size of input stream.
- **Minimal Communication:** As the energy consumption of transceiving data between the nodes is usually too big comparing to the computation cost inside the node, the size of data being sent from the sensor nodes to the base station must be minimized.
- **Incremental Clustering:** The algorithm must incrementally cluster the stream data points to detect evolving clusters over the time.
- **High Clustering Quality:** The algorithm must show a good approximation to the optimal clustering by reducing the clustering radius as much as possible, which is the criteria of measuring the k-center clustering quality.
- **Outlier Awareness:** The algorithm should not be sensitive to outliers, nevertheless, it must be able to detect trends in the input stream.

Apparently, not only the two parts of the last aspect are contradicting each other. The second aspect is met when additional in-the-node computations are done to reduce the size of the data to be sent to other node or to the base station, which opposes the first aspect. On the other hand the fulfillment of the third and fourth aspects contradicts again achieving the first one, high-quality incremental clustering needs to store summaries of old points and needs additional processing.

Although many attempts in the literature tried to fulfill the last three aspects, only limited work [5, 6, 10] has considered the first and the third ones. And to the best of our knowledge, no work has yet been done to consider all the previous aspects together.

Our algorithm considers all above aspects. Motivated by the Global Parallel Guessing algorithm [6], the key focus of our algorithm is to achieve better clustering quality by using less energy and being resource aware in addition to the consideration of outliers in the input stream.

The remainder of this document is organized as follows. Section 2 reviews previous work related to our clustering and communicating problem. In Section 3 we formulate the related problems. Section 4 describes our proposed EDISKCO algorithm in detail. Section 5 presents the experimental results. And finally we concludes the paper in Section 6 and discuss direction for future work.

# 2. RELATED WORK

We will list some previous work done in two strongly related areas: $k$-center clustering approaches and energy aware routing approaches for sensor networks.

## 2.1 $K$-Center Clustering Algorithms

In the $k$-center clustering problem of a group of points $P$, we are asked to find $k$ points from $P$ and the smallest radius $R$ such that if disks with a radius of $R$ were placed on those centers then every point from $P$ is covered [10]. The quality of the $k$-center clustering algorithms is measured using the approximation to the optimal clustering. Many clustering solutions have been presented for the $k$-center problem. We will start with reviewing available *offline* approaches which consider that all input data are available in the memory when applying the algorithm, then we will highlight some *online* algorithms which were developed mainly to deal with streaming inputs, we review then solutions which basically targeted the *distribution* of streaming input sources and we end up with solutions which considered the existence of *outliers* in the input stream.

### 2.1.1 Offline Approaches

These algorithm suggest that all of the $n$ input points are stored in the memory. In an early result, Gonzalez [9] gave the "Furthest Point Algorithm" which gives a 2-approximation to the optimal clustering by making *O(kn)* distance computations. Another 2-approximation approach, called the "Parametric Pruning" was given by Hochbaum and Shmoys [12]. The 2-approximation is the best possible solution since it was proved that it is *NP*-hard to find a $(2-\epsilon)$-approximation to the optimal clustering of the $k$-center problem for any $\epsilon > 0$ [7].

### 2.1.2 Online Approaches

Many algorithms were developed to cope up with streaming input, Charikar et. al [3] introduced the "Doubling Algorithm", a single pass streaming algorithm which guarantees an 8-factor approximation to the optimal clustering and uses *O(k)* space. The main idea is to apply the incremental clustering principle by using a *center-greedy* algorithm which continuously merges two cluster centers, making it possible to maintain new points without increasing the cluster radius. Although they have introduced the principle of incremental clustering with this algorithm, it was not clear how to get the approximation result by incrementally applying this algorithm. Guha [10] presented a $2(1+\epsilon)$ approximation single pass streaming algorithm using only $O(\frac{k}{\epsilon}\log\frac{1}{\epsilon})$ space. The algorithm directly forgets the input data and maintains the stored *summarization* of the input data, this summarization is limited to the available memory space which yields some summarization error. The algorithm is suitable for dealing with stream data by using a memory space independent on $n$, but it is not suitable for processing-limited machines (like sensor nodes) since it does multiple passing over the stored summarization. Cormode et. al [6] has formulated the "Parallel Guessing Algorithm" resulting with a $(2+\epsilon)$-approximation to the optimal clustering. This algorithm uses the first points in the input stream to make $\Delta$ guesses of $R$ as $(1+\frac{\epsilon}{2}), (1+\frac{\epsilon}{2})^2, (1+\frac{\epsilon}{2})^3, \ldots$ and then scans in parallel this part of the input stream using the different guesses. For each guess, it stops when it returns $k$ centers of this input stream using its radius. This will end up by storing $O(\frac{k}{\epsilon}\log\Delta)$ points, where $\Delta$ is the maximum number of guesses of the clustering radius that can be done. The smallest guess is then first used for clustering input data. Whenever a new point that is not covered by the recent clustering arrives, the current guess is announced as invalid,

and another bigger guessing must be selected. The algorithm is very sensitive to the first $k$ centers selected from received points and some of them might even be outliers, which reduces the clustering quality by using a big guess. The storage is dependent on $\Delta$ which can be in reality a big value for the limited storage of sensor nodes. In addition, the parallel nature of the algorithm does suite the limited processing ability of sensor nodes even for small values of $\Delta$.

#### 2.1.3 Distributed Approaches

The idea of having distributed *sites* each is maintaining a $k$-center clustering algorithm on his local input stream was originally raised by Cormode et. al [6]. The idea is to have $m$ remote sites applying the parallel guessing algorithm or an *online* furthest point algorithm on its local data. The site send its $k$-centers to a central site called *coordinator* which in turn applies another $k$-center clustering algorithm on the $k \times m$ centers. They proved that if the $k$-center clustering algorithm on the site side gives an $\alpha$-approximation and the one on the coordinator site gives a $\beta$-approximation then the resulting k-center clustering algorithm offers a $(\alpha + \beta)$ approximation to the optimal clustering of the whole input data. The suggested distributed algorithm was not mainly targeting sensor networks in the means of energy consumption.

#### 2.1.4 K-Center Clustering with Outliers

The idea of $k$-center clustering with outliers was first presented by Charikar et. al [4]. They gave an offline algorithm with a 3-approximation which drops $z$ outliers. Our algorithm in contrast drops $z$ far and non-dense outliers online by achieving $(2 + \epsilon)$-approximation. McCutchen et. al [13] presented another algorithm which gives a $(4 + \epsilon)$-approximation using $O(\frac{kz}{\epsilon})$ memory space. The algorithm reads the input in batches of size $(kz)$, stores them, drops all *non-outliers* and then applies an offline $k$-center clustering algorithm with outliers on them. Our algorithm in contrast perform a single pass over the input points without storing them or performing an offline processing.

### 2.2 Energy-aware Routing Approaches in Wireless Sensor Networks

After collecting their measurements from the physical environment and processing them, sensor nodes have to send these data to one or more base stations. By having the base station(s) within the radio range of each sensor node, the naive single-hop communication between each node and the base station is possible but not energy-efficient and not reliable because of possible resulting interferences. Additionally, most of the applications do not allow this single-hop deployment, which raised the need for a multi-hop routing path that takes energy efficiency into consideration. This is a pure network communication routing problem which was revisited for energy efficiency concerns. Figure 1 illustrates one solution to this problem. Choosing the right underlying routing protocol is particularly important when applying an energy-efficient clustering algorithm. This will guarantee a maximum compatibility for the sake of energy saving. Low-Energy Adaptive Clustering (LEACH) protocol [11] dynamically groups sensor nodes in a small number of clusters. The randomly chosen representatives (cluster heads) locally fuse data from their neighboring and transmit it to the base station, which results of a factor of 8 improvement compared to direct transmissions. In the Hybrid Energy-Efficient Distributed HEED Clustering Approach [16] the cluster head selection is mainly based on residual energy and the neighbor proximity of each node. We efficiently apply our algorithm over a networking protocol that extends the lifetime of the wireless sensor network. The protocol efficiently groups local sensor nodes that locally send their data to a one of them called coordinator which in turn aggregates these data and sends it to the far base station. The coordinator is iteratively changed depending upon the residual energy which is accurately estimated by our algorithm.

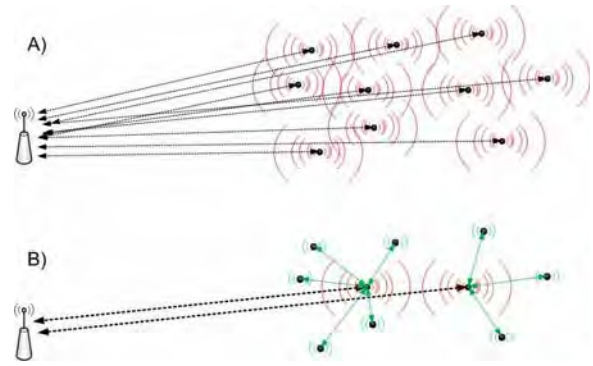



**Figure 1: An example of a routing protocol: A) All sensor nodes use a big sending power to send their data to the base station. B) Sensor nodes use lower sending power to send data locally to a local node which aggregates and sends them to the base station.**

## 3. PROBLEMS FORMULATION

We will try in this section to formally define the related problems to our algorithm.

### 3.1 The K-center Problem

Given a set $P$ of $n$ points $P = \{p_1, \ldots, p_n\}$, a distance function $d(p_a, p_b) \geq 0$ satisfying the triangle inequality and an integer $k < n$, the $k$-center set is $C \subseteq P$ such that $|C| = k$. The $k$-center problem is to find a $k$-center set $C$ that minimizes the maximum distance of each point $p \in P$ to its nearest center in $C$; i.e., the set which minimizes the quantity $max_{p \in P} min_{c \in C} d(p, c)$. The well-known $k$-median clustering problem is the the minsum variant of this problem where we seek to minimize $\sum_{p \in P} min_{c \in C} d(p, c)$ [10].

### 3.2 The Incremental Clustering Problem

Let $S = \{c_1, c_2, \ldots, c_k, R\}$ be a *current* solution of a $k$-center clustering algorithm **A** applied on $n$ input points that are arriving to the algorithm one by one in a sequence of updates. **A** is an incremental clustering algorithm if it can always maintain a valid solution over the flow of stream. In other words, whenever a new point arrives to the algorithm it should either be assigned to one of the clusters indicating the validity of current clustering, or it does not fit in any of the current clusters then the current $S$ must be changed into another solution $S'$ such that this new point is assigned to some cluster in the new solution $S'$. $S'$ can differ from $S$ by the centers, radius or by both of them.

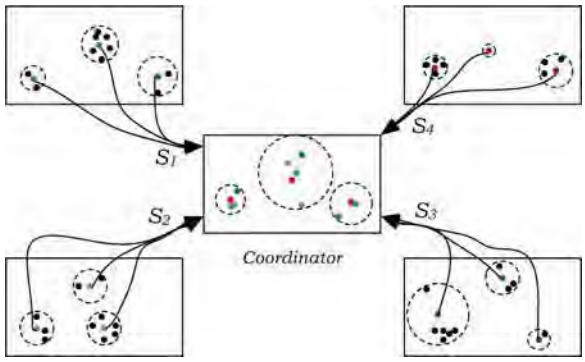



**Figure 2: An example of a distributed *k*-center clustering for $k$=3 clusters of data coming from $d$=4 sites, the coordinator applies another *k*-center clustering over the $k \times d$ centers sent from the sites.**

## 3.3 The Distributed Clustering

In distributed clustering we track each sensor node data locally, process it and then combine the results in a central node or a *coordinator*. The target is to minimize the communication and share the resources. We define the **distributed clustering problem.** Let $1, 2, \ldots, d$ be distributed sites, each site $i$ applies a clustering algorithm $A_i$ on his stream input of data $X_i$ and produces a solution $S_i$. It is required to perform a global clustering of all $X_i$; $i = 1 \ldots d$ input streams distributed over the sites. One efficient solution to do that is to have a central site which collects the union $\bigcup_{i=1}^{d} S_i$ and answers the querying or monitoring requests of the whole input streams. It is also possible that a further clustering algorithm $B$ at the coordinator to be applied on $\bigcup_{i=1}^{d} S_i$. In the distributed $k$-center clustering problem we will consider in this paper, the solution of $A_i$ at each site $i$ is $S_i = \{c_{i1}, c_{i2}, \ldots, c_{ik}, R_i\}$, and on the coordinator side, we perform another $k$-center clustering algorithm over the whole $k$ centers coming from the whole $d$ sites, i.e., $\bigcup_{i=1}^{d} \{c_{i1}, c_{i2}, \ldots, c_{ik}\}$. When applying incremental clustering algorithms, continuous updates with the new solutions must be sent from the sites to the coordinator. Figure 2 illustrates an example where the coordinator applies another $k$-center clustering over the $k \times d$ centers sent from the sites.

## 3.4 The Problem of $K$-center Clustering with Outliers

Taking the existence of some outliers in the input stream into account is a natural consideration. Since data is often noisy and this can dramatically affect the quality of the solution if not taken into account. This particularly applies to the $k$-center objective function which is extremely sensitive to the existence of points far from cluster centers [13]. This sensitivity has two effects: (a) On the clustering quality which appears in Figure 2 for example in site 3 where the cluster on the left has a bigger clustering radius (worse clustering quality) because of not considering one point too far from the center as an outlier. And (b) On the energy consumption in the distributed model since outliers causes current solutions to be invalid and need to updated with the coordinator. Formally in the ***k*-center clustering problem with outliers** we group $m$ out of the $n$ input points into the $k$ clusters by dropping $z = n - m$ points, the target is to minimize the clustering radius. The decision of labeling those $z$ points as *outliers* is done when they are farther than $R$ from the current $k$ centers and the number of neighboring *non-clustered* points is not "enough" to establish a new cluster.

## 3.5 The Energy Cost Model

Considering sensor nodes that contain sensing, processing and radio parts. The total energy consumption $E$ of a sensor node can be defined as:

$$E = E_{comp} + E_{transc} + E_{sensing} + E_{sleep} \quad (1)$$

where:

- $E_{comp} = E_{proc} + E_{acces}$: the energy consumption of the computations, divided into $E_{proc}$ the consumption of processing done by the microprocessor and $E_{acces}$ the energy cost of accessing an external memory to load or store the data.
- $E_{transc} = E_t + E_r$: the energy consumption of transmitting $E_t$ and receiving $E_r$ done by the radio.
- $E_{sensing}$: the energy consumption of the sensors.
- $E_{sleep}$: the energy consumption when the microprocessor is in the sleep mode.

The energy consumption of each part depends both on the time and current draw of this part when it is active. By considering both $E_{comp}$ and $E_{transc}$:

$$E_{comp} = E_{proc} + E_{acces} = V \cdot I_c \cdot t_c + V \cdot I_a \cdot t_a \quad (2)$$

$$E_{transc} = E_t + E_r = ss_t \cdot V \cdot I_t \cdot t_t + V \cdot I_r \cdot t_r \quad (3)$$

Where:
$V$: the working voltage of the sensor node
$I_c$: the current draw of the microprocessor in the active mode
$t_c$: the time in which the microprocessor is running
$I_a$: the current draw of accessing the external memory
$t_a$: the time taken to access the external memory
$ss_t$: the transmitting signal strength of the radio
$I_t$: the current draw of the radio when transmitting
$t_t$: the time when the radio is transmitting
$I_r$: the current draw of the radio when sending
$t_r$: the time when the radio is sending

Let $c_t$, $c_r$ be the size of the data to be transmitted or received, and $f_{transc}$ be the radio speed, then: $t_t = f_{transc} \cdot c_t$ and similarly: $t_r = f_{transc} \cdot c_r$, this makes Equation (3) looks like:

$$E_{transc} = E_t + E_r = f_{transc} \cdot V \left( ss_t \cdot I_t \cdot c_t + I_r \cdot c_r \right) \quad (4)$$

Usually in sensor nodes, the current draw of the microprocessor unit $I_c$ is too small compared to $I_a$, $I_t$ and $Ir$, in addition, the microprocessor speed is also too big compared to $f_{transc}$ or the external memory bus speed. All that together makes $E_{transc}$ and $E_{acces}$ too big compared to $E_{comp}$ in the sensor node. This highlights the importance of reducing

both $E_{transc}$ and $E_{acces}$ in an in-sensor-node energy-aware clustering algorithm. $E_{acces}$ is reduced by limiting the need to access the external memory as much as possible, a target which can be achieved by making as less as possible passes over the input data. And from the cost model in Equation (4), it can be seen that $E_{transc}$ can be reduced by reducing $c_t$ and $c_r$ the size of the data to be sent or received, while an energy aware routing protocol tries to reduce $ss_t$.

# 4. THE EDISKCO ALGORITHM

In the section we present the Energy-efficient Distributed In-Sensor $K$-center Clustering algorithm with Outliers EDISKCO. Each local sensor node receives its input stream through its sensors and produces a $k$-center clustering solution to it by considering the existence of outliers and sends this solution to the coordinator. The coordinator performs another clustering algorithm to the solutions coming from the sites. Therefore, the input streams are processed at each node locally, and a global clustering of all sensors data is performed globally in the coordinator . We will describe the algorithm on the node side, the coordinator side and on the server side. We introduce a special heap structure for storing a $k+z$ members, each member $c_j$; $j = 1 \ldots (k+z)$ in this heap represents a cluster, where: $c_j.center$ represents the center of this cluster and $c_j.count$ represents the number of members inside this cluster (density). The members in this heap are arranged in a descending oder according to $c_j.count$. The member on top of the heap where $(j = 1)$ represents the cluster with the highest density.

We define the following functions on $h$:

- ***maintain(h)***: applied after each change in $c_j.count$; $j = 1 \ldots k+z$ such that if $j > q$ then $count_q \geq count_j$ for all $q = 1 \ldots k+z$ and $j = 1 \ldots k+z$.
- ***size(h)***: returns the number of the members in $h$ which can be any value between 0 and $k+z$.
- ***get(h,j)***: returns the member $j$ from the heap.
- ***delete(h,j)***: deletes the member $j$ from the heap and directly maintains the heap.
- ***insert(h,p)***: inserts an input point $p$ from the stream in $h$, (see Algorithm 1). It scans the members of $h$ beginning by the high dense ones. When a cluster is found where $p$ is not further than $R$ from its center, this cluster's counter is incremented by one, and $p$ is forgotten. If $p$ was further than $R$ from all available cluster centers and there was less than $k+z$ members in $h$, then a new cluster is established with $p$ is its center. Otherwise an error is returned for not having a place to add $p$.

## 4.1 On The Node Local Side

Each node $i$ receives an input stream $X(i)$ and runs the NodeSideEDISKCO algorithm (see Algorithm 2) and sends updates to the coordinator with $k$ center outlier-aware clustering representation of $X(i)$ in addition to the corresponding radius $R_i$. Whenever an input point $p$ is received, the algorithm applies the $insert(p, h)$ function. If $p$ fits in one of the available clusters then its member counter is increased,

**Algorithm 1** ***insert(h,p)***

| | |
|---|---|
| **In** | *A heap h of current clusters and an input point p from the stream* |
| **Out** | 0:*if OK, j: if p is a new cluster in position j or err:if there is no place to add new cluster p* |

```
for j = 1 to size(h) do
  if d(p, c_j.center) ≤ R then
    c_j.count+ = 1
    maintain(h)
    return 0
  end if
end for
if size(h) < (k + z) then
  c_j.center = p
  c_j.count = 1
  return j
else
  return err
end if
```

if not then it is necessary to establish a new cluster containing $p$ as its center. We limit the number of clusters to be established to $(k+z)$: $k$ most dense clusters for representing the solution and $z$ lowest dense centers for representing the outliers. The main target of our algorithm is to reduce the communication with the coordinator by excluding outliers from final solution, so only if this new established cluster falls in the first $k$ most dense clusters then an update is sent to the coordinator containing *new_center* signal and the point $p$. If we need to establish a new cluster for $p$ and we already have $(k+z)$ clusters, then the current clustering is not longer valid and the solution needs to be maintained. The node $i$ does this by increasing the current radius. In order to select the best new radius size, the node $i$ sends a *radius_increase* request message to the coordinator. The coordinator in turn replies to $i$ with an acknowledgment message with the biggest clustering radius it has noticed from all nodes ($R_{global}$). Node $i$ buffers its input points from $X(i)$ till it receives the acknowledgment from the coordinator. When it receives it, the node $i$ selects $max\{R_i(1+\frac{\epsilon}{2}), R_{global}\}$ as the next radius, and starts scanning the available clusters in $h$ beginning with the highest density cluster by comparing the distance between their centers with the new $R_i$. If a center $j+1$ is less than the $R_i$ away from the center $j$ where $j = 1 \ldots k+z$, then the cluster member $j+1$ is deleted and the counter of the cluster $j$ is increased, and $j$ is compared with the new member in $j+1$ again. If the member in $j+1$ is farther than $R_i$ from $j$, then $j+1$ is kept and the same comparison happens between $j+1$ and $j+2$. Finally, we take the most dense $l < k$ resulting clusters available and delete the others. The last scanning procedure guarantees that we are keeping the effect of the most $l$ dense clusters which appeared in history solutions. On the other hand, taking only $l$ clusters will leave a space for the establishing new clusters if there was a new trend in the input stream. This is very essential for incremental clustering algorithm on the one side, on the other side this will reduce the number of *radius_increase* requests and consequently the energy consumption of the node.

In the initialization phase of the NodeSideEDISKCO algorithm (not shown in Algorithm 2 for readability), the node

increases the radius without contacting the coordinator until $\sum_{j=1}^{k} c_j.count \geq n$, where $n$ is the minimum number of points that we want to represent using the $k$ center. Only in this case the current solution and radius are valid and can be sent to the coordinator. The algorithm continues the comparison on the buffered input points from $X(i)$ and sends the resulting centers with the used radius to the coordinator. The decision of sending a *radius_increase* request to the coordinator is made also if the total number of points which considered to be in the outliers became more than $o$. Having more than $o$ outliers close to each other in $z$ or less clusters means that the decision of considering those points as outliers is not longer valid and needs to be changed by a new clustering. Because the coordinator is performing more communication and computation tasks, our algorithm is also aware of iteratively changing the coordinator in order not to have one node exhausted and died. The decision of choosing the node $i$ as the next coordinator is made by the server (as we will see) based on the residual energy of all nodes. In this case, the server sends a *next_coordinator* message to $i$. The node $i$ saves its local solution $C_{local}$ and switches to the coordinator algorithm and a new *phase* begins.

**Algorithm 2 NodeSideEDISKCO**$(X_i, k, z, o, l, \epsilon)$

**In** *An input stream $X_i$, num of clusters $k$, outlier clusters num $z$, maximum outliers num $o$, num of most dense clusters $l$ and step size $\epsilon$*

**Out** *update the coordinator with the new opened clusters centers and radii*

```
X_i(t) → p, c_1.center = p, c_1.count = 1, R = R_min
while there is input stream X_i(t) do
    X_i(t) → p
    fit = insert(p, h)
    if next_coordinator signal from server then
        Save current local centers in C_local
        Switch to CoordinatorSideEDISKCO
    end if
    if fit ≠ err and 0 < fit ≤ k then
        Send the new center get(h, fit) with a
        new_center signal to the coordinator.
    end if
    if fit == err or Σ_{j=k+1}^{size(h)} c_j.count > o then
        Send radius_increase request to coordinator
        while there is no reply from coordinator do
            Buffer incoming input points from X_i
        end while
        R ← max{R(1 + ε/2), R_global}
        while j < size(h) do
            if d(c_j.center, c_{j+1}.center) ≤ R then
                c_j.counter++, delete(h, j + 1)
            else
                j++
            end if
        end while
        Keep only the most l dense centers in h.
        Run this algorithm on buffered points.
        C_i = {c_1.center, c_2.center, ..., c_k̂.center}
        Send C_i, R_i as an update to the coordinator
    end if
end while
```

## 4.2 On The Coordinator Side

The coordinator receives the local solutions $C_i$, radii $R_i$ and *radius_increase* requests from the nodes and performs the CoordinatorSideEDISKCO algorithm (see Algorithm 3). When a new phase begins, the new coordinator receives the global solutions $C_{global}$ and the global radius $R_{global}$ from the previous coordinator. Then the new coordinator announces himself to the other nodes as the new coordinator. The coordinator continuously performs the Furthest Point algorithm [9] on the solutions $C_i$ arriving from the sites. It starts by applying it on its own local solution $C_{local}$, and then on $C_i$. The coordinator saves always the largest radius used by a node as the $R_{global}$ and whenever any node $i$ wants to increase its clustering radius, the coordinator replies an acknowledgment with $R_{global}$. The coordinator also adds the new centers $c_{ij}.centers$ arriving from the nodes to the current solution.

The coordinator keeps a special space for saving summary about the energy consumption of each node $i$. The total number of centers that were sent from a node $i$ and the total number of *radius_increase* requests sent from a node $i$ during this phase are saved under $numCenters_i$ and $numRequests_i$ respectively. These are important for the server to calculate the total energy consumption of each node including the coordinator during this phase. The server sends from time to time *consumption_update* requests to the coordinator which in turn replies with the them. According to the residual energy of each node and the coordinator, the server makes a decision of changing the current coordinator by sending *chg_coordinator* message to it, with the *id* of the next coordinator. In case of having a huge number of nodes, the server can group the nodes into different groups each covered by one coordinator, this is to avoid a possible collapse of a single coordinator due to the heavy messages traffic or the too big data to be saved in memory. On idea to do this is by considering the -already known by the server- spatial information of the node in making these groups and then considering the residual energy when deciding the next coordinator of each group. On the local side of each group, since each coordinator is transceiving data using minimal sending strength, then this will guarantee that a node will be covered by only one coordinator. It might happen that a node receives more than one *I_am_coordinator* signal, then it decides to be covered by the coordinator with the strongest signal, which is basically its group coordinator. Before the end of the next phase, the coordinator send $C_{global}$ and $R_{global}$ to the next coordinator, and switches to the node algorithm.

## 4.3 On The Server Side

The far server is responsible for selecting the coordinator for the next phase according to the residual energy in each node. All of the nodes start with the same level of energy. The server starts by randomly selecting one of them as the coordinator of this phase. And then from time to time, the server collects the $numCenters_i$ and $numRequests_i$ of each node from the coordinator. The server uses these information and the Equation (1) to estimate the energy consumption of each node, and also the coordinator. All nodes are sensing samples with the same frequency, so the server can estimate the total number of points taken in this phase $numPoints_i$.

The coordinator is collecting all $C_i$, requests from the node

**Algorithm 3** CoordinatorSideEDISKCO($C_i, R_i, c_{ij}.center$)

**In** *solutions $C_i$, $R_i$, new opened cluster centers cluster_increase requests from node i; $i = 1 \dots m$*

**Out** *send ack and $R_{global}$ to nodes, maintain $C_{global}$, $R_{global}$ and send them to next coordinator*

```
if this is not the first coordinator then
    Receive C_global, R_global from last coordinator
end if
Broadcast I_am_coordinator signal to all nodes
C_global ← FurthestPoint(C_global, C_local)
if radius_increase_i do
    Send to node i an ack with R_global
    Receive C_i, R_i from node i
    C_global ← FurthestPoint(C_global, C_i)
    R_global ← max{R_global, R_i}
end if
if new_center_i do
    C_global ← FurthestPoint(C_global, {c_ij.center})
end if
if consumption_update received from server then
    Send {numCenters_i, numRequests_i} for all
    i = 1 ... m nodes during this phase to server
end if
if chg_coordinator received from server then
    Send C_global, R_global to next coordinator
    Switch to NodeSideEDISKCO using R_global
end if
```

and replying with the acknowledgments and $R_{global}$ to each request. On the other hand it is answering the queries of the far server using its full sending strength and performing more complicated multipass $k$-center clustering algorithm over $C_i$'s. Thus it is consuming much more energy comparing to the normal nodes. If we keep using the same coordinator, it will die after a while, and we will lose the connection to the other *still alive* nodes.

The target is to change the coordinator to the node that contains the most residual energy, such that all of the nodes in the network will die as close as possible to each other. This will let us make maximum use of the whole network lifetime.

The server assumes the following worst cases: (a) All the $numCenters_i$ points needed $k$ comparisons until they were placed in a new cluster and then sent, (b) All the ($numPoints_i - numCenters_i$) points needed $(k+z)$ comparisons until they were saved and (c) Only $l$ centers were sent at once to the coordinator after sending *raduis_increase* to it, while the other ($numCenters_i - l \times numRequests_i$) were sent one by one to the coordinator during this phase. Although these conservative assumptions might be a little bit far from the real case, the server can continue using the last coordinator even if it was supposed to die as long as it is *really* still alive.

## 4.4 Lower Bounds of EDISKCO

EDISKCO algorithm was mainly motivated by the Global Parallel Guessing (Global-PG) algorithm which has an altogether $(4+\epsilon)$-approximation to the optimal clustering quality to the global optimal solution [6]. On the coordinator side, both PG and EDISKCO algorithms are using the Furthest Point algorithm which has a 2-approximation to the optimal clustering (see [6] for proof). After sending the first solution to the coordinator, our algorithm incrementally maintains a $k$-center solution for at least $n$ input points. The solution stays valid unless we have more than $z$ clusters of outliers or more than $o$ input points assigned as outliers. Following the same way used in proving the lower bound of the parallel guessing algorithm in [6] one can proof that the NodeSideEDISKCO finds a $(2+\epsilon)$-approximation to the optimal $k$-center clustering of $n$ input points. EDISKCO does not exclude the non-outlier points from the final solution, it simply delays the decision of increasing the radius until one is sure that those points are not really outliers. EDISKCO then either considers them in the final solution if they have formed a dense cluster, or reclusters taking all of them into consideration. By applying the Furthest point algorithm on the solutions on coordinator side which has a 2-approximation, the EDISKCO algorithm finds an altogether $(2+\epsilon)+2 = (4+\epsilon)$-approximation of the optimal global clustering solution, by saving only $(k+z)$ points and making a maximum of $(k+z)$ comparisons for each input point. Due to its insensitivity to the first input points and outliers in addition to its incremental nature compared to Local-PG, we will show in the next section that our algorithm has a better clustering quality and energy consumption than the Global-PG.

**Table 1: Total energy consumption of EDISKCO and Global-PG for each dataset in Joule**

| Dataset | Size | Nodes | EDISKCO | Global-PG |
|---|---|---|---|---|
| RW | 42000 | 19 | **573382.4** | 587015.5 |
| i9-Sensor | 40589 | 19 | **547270.2** | 558121.8 |
| Physio | 24000 | 12 | **205970.6** | 214380.8 |

# 5. EXPERIMENTS

In order to evaluate the performance of EDISKCO, we performed extensive experiments on both synthetic and real data. For comparison, we have chosen the Global-PG as a single-pass distributed $k$-center clustering algorithm on the node sides which applies also the furthest point algorithm on the coordinator side. In order to have fair results, we have implemented our suggested node-coordinator-server model on the Global-PG. We have implemented simulations of both algorithms in Java.

We have chosen one synthetic dataset and two real world datasets, we give here a small description of each.

**Synthetic data set: RandomWalk (RW)**[1] A synthetic data set based on the random walk model. The increments between two consecutive values are independent and identically distributed. Each increment: $t_{i+1}$ is produced by randomly adding or subtracting from $t_i$ a uniformly random value from the interval [1,10] . We generated 19 different data sets each for one node, each containing 42,000 measures. Subsequently, to produce a natural outliers effect, we replaced randomly selected vales (4.5% of the dataset size) with noise values, uniformly at random in the interval $[min, max]$ of the dataset.

**Real Dataset: I9 Sensor Dataset**[1] We have collected a

[1]dataset is available under http://dme.rwth-aachen.de/EDISKCO

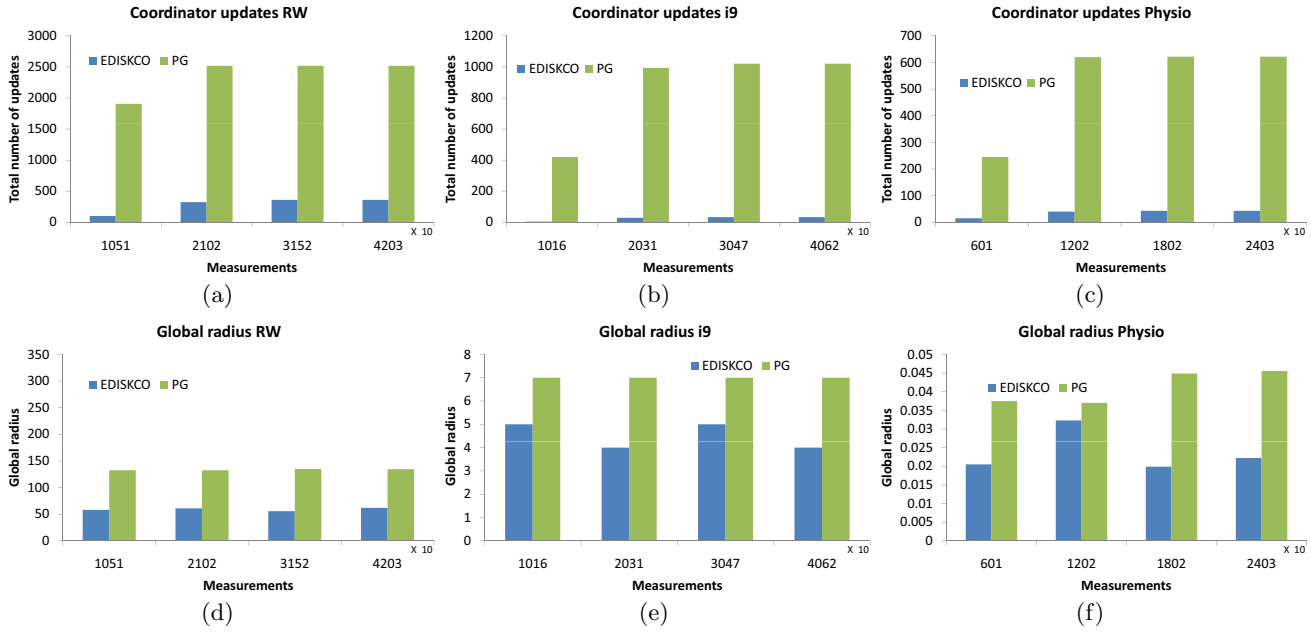



**Figure 3: Total number of requests (a)-(c), $R_{global}$ (d)-(f) of the: RW, i9-Sensor and Physio datasets**

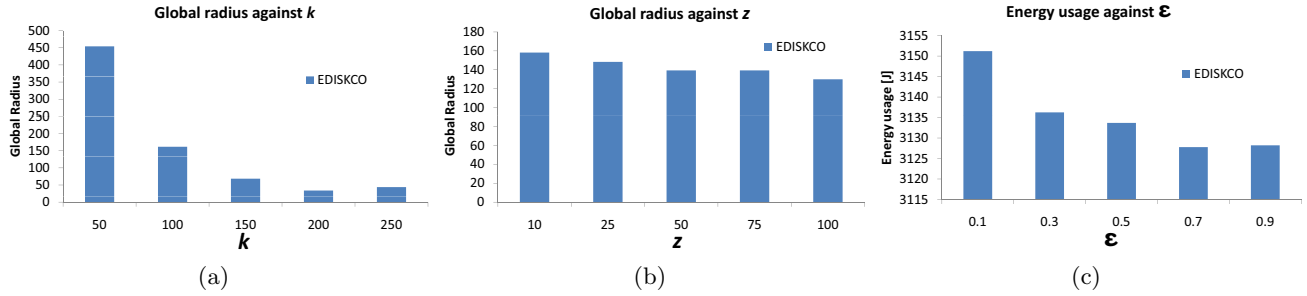



**Figure 4: Parameter sensitivity (a)-(b) in RW, (c) in i9-Sensor**

real data from a sensor network. We deployed 19 TelosB motes in our department area. All motes were programmed to collect temperature samples each 30 seconds and send them directly to a sink connected to a computer. The data was collected for more than 14 days between the 10th and the 23rd of April 2009 and forming 40,589 measures of each node. The minimum difference between raw measures is 1. Nodes were not always able to communicate perfectly with the sink due to collisions or loss of signal, this appeared in 2.9% of the total data. Instead of each measure that did not reach the sink, we introduced a noise data. In a different way of adding outliers to that of RW, a uniformly at random value from the interval $[0.1 \times (max - min), 0.25 \times (max - min)]$ was selected and then uniformly at a random either added to $max$ or deducted from $min$, the resulting value was inserted instead of the lost measure.

**Real Dataset: Physiological Sensor Dataset** This data was presented in ICML 2004 as a challenge for information extraction from streaming sensor data. The training data set consists of approximately 10,000 hours of this data containing: userID, gender, sessionID, sessionTime, annotation, characteristic 1, characteristic and sensor[1..9]. We have extracted the first 24,000 readings of sensor2 by userID. We have chosen the data of 12 different userIDs with the same gender, each representing a node. We did not add outliers to this dataset as they are naturally exist in such datasets.

## 5.1 Evaluation Criteria and Methodology

Three criteria were chosen to evaluate our algorithm, namely:
**Clustering Quality:** We have selected the global clustering radius as a measure to evaluate the quality of EDISKCO with Global-PG. In $k$-center clustering, better clustering uses smaller radius to cover all of the input points.
**Energy Consumption:** Two measures were selected to evaluate the energy efficiency of EDISKCO: (a) Total number of reclustering events, which tells about the total number of costly requests sent from all the nodes to the coordinator for increasing the current clustering radius. (b) Total energy consumption of the whole network in Joule based on the detailed cost model suggested in Equation (1) and the datasheets of TelosB mote, TI MSP430 microcontroller and the CC2420 radio chip in addition to the TinyOS 2.0.2 op-

erating system installed on the motes.
**Parameter Sensitivity:** We wanted to see the effect of varying $\epsilon$, $k$ and $z$ parameters on the previous measures of our algorithm.

## 5.2 Experimental Setup and Results

For performing the **clustering quality and the energy consumption evaluations**, we have selected for our algorithm for all datasets: $k = 200$, $z = \frac{k}{4}$, $\epsilon = 0.5$, the number of the most dense clusters to be sent to the coordinator after each cluster increase: $l = \frac{k}{4}$, the maximum allowed number of input points after which the node can send a solution to the coordinator: $n = 400$, the maximum allowed number of outliers in total: $o = 10\%n$. For the Global-PG we have selected the number of points collected at the beginning to perform the parallel guessing equals to our $n = 400$ and also $\epsilon = 0.5$. We performed the evaluations on a 3.00 Ghz core Duo, 4 GB RAM machine.
For both algorithms we need to set the initial radius $R_{min}(P) = min_{p,q \in P, p \neq q} d(p, q)$ for each dataset. This is usually possible since sensor nodes are dedicated for certain measurements and that we have experts in the field who can decide this value. Mostly this value is already determined by the precision of the sensors. For the i9 sensor dataset $R_{min}$ equals to 1, but for the physiological and RW datasets this was too small comparing to $R_{max}$. The direct selection of the $R_{min}$ in those two datasets was extremely affecting running time of the Global-PG algorithm due to the enormous number of guesses to be done in this case. Even the desktop machine we used for evaluation was hanging. This is one of the main drawbacks of Global-PG when applied on resource limited sensor nodes. In contrast, we have not observed that effect with our algorithm. For the experiments we have selected $R_{min}$ equals to 1 for the RW and equals to 0.0001 in the physiological dataset.
For performing the **parameter sensitivity** evaluations, we selected $k = 100$ other parameters are same as above. For both algorithm we applied our suggested routing protocol for changing the coordinator, the server was lazily querying the current coordinator for $numCenter_i, numRequests_i$. For the three datasets and the two algorithms, Table 1 represents the total energy consumption in Joules, while Figure 3 depicts the accumulated total number of reclustering events and the current global radius with respect to the flow of input stream.

Figure 3(a) shows that EDISKCO sends 85% less updates to the coordinator than Global-PG, and Figure 3(d) shows that EDISKCO achieves 55% better clustering quality of the RW data. Because all of the guesses are done on first received input stream, Global-PG needs to update the coordinator whenever it receives a new point which does not fit in the current clustering. This results in a heavy node-coordinator communication from one side, and a bigger clustering radius to include the new points on the other side. This effect appears even in the dataset which does not contain introduced outliers like Physio dataset (Figures 3(c) and 3(f)), where EDISKCO achieves 93% less number of updates and 14% to 58% better clustering quality. Another reason of this low updates number of EDISKCO is its scanning method when inserting a new point, which starts by scanning dens clusters that are more likely to contain additional points. In consequence, EDISKCO consumes less total energy than Global-PG.
Table 1 shows that EDISKCO saves 2.32% comparing to Global-PG when applied on RW dataset and 2%, 3.9% when applied on Sensor-i9, Physio datasets respectively. The saving is considerable for extending the lifetime of the whole network. In most of the updates, Local-PG sends only the one new center.
Our cost model suggests a direct relation between $E_{transc}$ and the size of data to be sent. This explains the differences between improvement percentages of Table 1 and Figure 3. In reality, heavy communication consumes more energy due the higher possibility of collisions and signal loss.
In Table 1, Global-PG consumes additional energy in the Physio dataset due to the huge number of guesses done at first when considering $R_{min} = 0.0001$. Figures 4(a) and 4(b) present the effect of changing $k$ parameter in the interval [0, 250] and the $z$ parameter in the interval [0, k] on $R_{global}$ in the RW dataset.
Figure 4(c) depicts the effect of changing the accuracy parameter $\epsilon$ on the single node usage in the i9-Sensor dataset. It shows that the less the clustering accuracy, the less the energy consumption, a natural relation which highlights the significance of our approach by achieving better clustering quality in combination with a longer lifetimes.

# 6. CONCLUSIONS AND FUTURE WORK

In this work we present our novel energy efficient k-center clustering solution. As a single-pass algorithm we develop an incremental processing which is aware of outliers in the data. We enhance the clustering quality by excluding these outlying objects from the clustering. Furthermore, we reduce the cost of intensive reclustering operations and achieve lower energy consumption. For the limited energy resources of sensor networks our energy efficient computation induces longer lifetimes of the network. In thorough experiments we present the high clustering accuracy and low energy consumption of our approach. Furthermore, our algorithm is also aware of limited memory resources in todays sensor nodes.

In future work, we aim in finding a way for predicting the best values of $k$ and $z$ for the next phase based on the data received in the last phase, such a work can be done on the server side using the prediction strength method developed by Tabshirani et al. [18].
We see further requirements for efficient clustering on sensor networks for high dimensional data. In recent applications, sensor nodes include many attributes as measuring and storing of data has become very cheap. Without thinking about the relevance of attributes during data collection, mining tasks have to cope with high dimensional data. In such data, distances grow more and more alike due to an effect termed the "curse of dimensionality" [2]. The full data space is thus sparse and all objects seem to be outliers as no global clusters can be observed. In future work we aim at tackling curse of dimensionality by clustering in subspace projections. We will extend our subspace clustering [1, 14] and outlier mining [15] approaches to sensor networks. Considering only relevant measurements in projection may lead to further reduction of transmission costs as irrelevant sensor measurement are not transmitted any more.

## Acknowledgments

This research was funded in part by the cluster of excellence on Ultra-high speed Mobile Information and Communication (UMIC) of the DFG (German Research Foundation grant EXC 89).

## 7. REFERENCES

[1] I. Assent, R. Krieger, E. Müller, and T. Seidl. INSCY: Indexing subspace clusters with in-process-removal of redundancy. In *Proc. IEEE ICDM*, pages 719–724, 2008.

[2] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbors meaningful. In *Proc. IDBT*, pages 217–235, 1999.

[3] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. In *Proc. ACM STOC:*, pages 626–635, 1997.

[4] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proc. SODA*, pages 642–651, 2001.

[5] M. Charikar, L. O'Callaghan, and R. Panigrahy. Better streaming algorithms for clustering problems. In *Proc. ACM STOC*, pages 30–39, 2003.

[6] G. Cormode, S. Muthukrishnan, and W. Zhuang. Conquering the divide: Continuous clustering of distributed data streams. In *Proc. IEEE ICDE*, pages 1036–1045, 2007.

[7] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proc. ACM STOC*, pages 434–444, 1988.

[8] A. R. Ganguly, J. Gama, O. A. Omitaomu, M. M. Gaber, and R. R. Vatsavai. *Knowledge Discovery from Sensor Data*. 2008.

[9] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, 1985.

[10] S. Guha. Tight results for clustering and summarizing data streams. In *Proc. ICDT*, pages 268–275, 2009.

[11] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. HICSS*, 2000.

[12] D. Hochbaum and D. Shmoys. A best possible approximation algorithm for the k-centre problem. *Math. of Operations Research*, 10:180–184, 1985.

[13] R. Matthew Mccutchen and S. Khuller. Streaming algorithms for k-center clustering with outliers and with anonymity. In *Proc. Workshop APPROX / RANDOM*, pages 165–178, 2008.

[14] E. Müller, I. Assent, R. Krieger, S. Günnemann, and T. Seidl. DensEst: Density estimation for data mining in high dimensional spaces. In *Proc. SIAM SDM*, pages 173–184, 2009.

[15] E. Müller, I. Assent, U. Steinhausen, and T. Seidl. OutRank: ranking outliers in high dimensional data. In *Proc. DBRank Workshop at IEEE ICDE*, pages 600–603, 2008.

[16] S. F. Ossama Younis. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 10 2004.

[17] A. S. Tanenbaum, C. Gamage, and B. Crispo. Taking sensor networks from the lab to the jungle. *Computer*, 39(8):98–100, 2006.

[18] R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational & Graphical Statistics*, 14(3):511–528, September 2005.

# Phenological Event Detection from Multitemporal Image Data

Ranga Raju Vatsavai
Computational Sciences and Engineering Division
Oak Ridge National Laboratory
MS-6017, Oak Ridge, TN 37831
vatsavairr@ornl.gov

## ABSTRACT

Monitoring biomass over large geographic regions for seasonal changes in vegetation and crop phenology is important for many applications. In this paper we a present a novel clustering based change detection method using MODIS NDVI time series data. We used well known EM technique to find GMM parameters and Bayesian Information Criteria (BIC) for determining the number of clusters. KL Divergence measure is then used to establish the cluster correspondence across two years (2001 and 2006) to determine changes between these two years. The changes identified were further analyzed for understanding phenological events. This preliminary study shows interesting relationships between key phenological events such as onset, length, end of growing seasons.

## Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; I.5 [**Pattern Recognition**]: Clustering

## Keywords

Clustering, EM, GMM, Remote Sensing, MODIS, NDVI

## 1. INTRODUCTION

The launch of NASA's Terra satellite in December of 1999, with the MODIS instrument aboard, introduced a new opportunity for terrestrial remote sensing. MODIS data sets represent a new and improved capability for terrestrial satellite remote sensing aimed at meeting the needs of global change research. With thirty-six spectral bands, seven designed for use in terrestrial application, MODIS provides daily coverage, of moderate spatial resolution, of most areas on the earth. Land cover products are available in 250m, 500m, or 1000m resolutions [4]. MODIS land products are generally available within weeks or even days of acquisition and distributed through the EROS Data Center (EDC) and are currently available free of charge.

(c) 2009 Association for Computing Machinery. ACM acknowledges that this contribution was authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only. SensorKDD-09, June 28, 2009, Paris, France. Copyright 2009 ACM 978-1-60558-668-7...$5.00

The availability of multi-temporal MODIS imagery has made it possible to study plant phenology, quantitatively describe NPP patterns in time and space, and monitor and map natural resources at regional and global scales. MODIS allows users to identify vegetation changes over time across a region and estimate quantitative biophysical parameters which can be incorporated into global climate models. Even though several cumulative vegetation indices can be found in the literature, in this study we used MODIS NDVI temporal profiles.

**NDVI Profiles**: A temporal profile is a graphical plot of sequential NDVI observations against time. These profiles quantify the remotely sensed vegetation's seasonality and dynamics. These profiles can be described with simple parameters, like the amplitude, mean, and standard deviation. We can understand the onset and peak of greenness and the length of growing season from analyzing these profiles.

MODIS data has been extensively used to study vegetation and crop phenological characteristics [5, 7], monitoring [11]. In this study, we tried to find the changes using automated unsupervised learning technique and relationships between key phenological events with respect to the changes. Rest of the paper is organized as following. In Section 2, we present the preprocessing techniques applied to automate the change detection and knowledge discovery processes. Clustering technique is presented in Section 3 followed by change detection process in Section 4. Summary of results were presented in Section 5.

## 2. DATA PRE-PROCESSING

Although remotely sensed imagery data are a common base for the development of GIS implementations, the use of MODIS data, and much of the data from NASA's Earth Observing System, has been limited in part by technical difficulties. A MODIS users survey at the Fall 2000 meeting of the American Geophysical Union revealed that the major limitations to access to this data included difficulty transferring and subsetting the large data files and not having software to handle the HDF-EOS file format in which MODIS files are delivered [3]. In addition, the data are delivered in an Integerized Sinusoidal Grid, which is a new projection, not well supported by most GIS software packages [10]. Though most of these problems have been solved, MODIS data requires preprocessing before data mining techniques can be applied. We have streamlined the data are preprocessed to reduce complexity and duplication of efforts. Pre-

processing tasks included 1) converting the data to a format recognizable by GIS software and reprojecting data into a UTM projection, 2) mosaicking scenes, 3) subsetting the data to the area of interest, 4) rescaling 16-bit data to 8-bit, 5) sampling mechanisms to extract random samples for clustering and classification, 6) atmospheric corrections, and 6) exporting the data to generic formats accessible by Matlab, R, and Weka systems. We now describe few common preprocessing techniques applied before performing clustering and change characterization.

**MODIS Land Products:** The MODIS land products are derived from the data available through the EROS Data Center (http://edcdaac.usgs.gov) and are delivered in HDF-EOS format projected onto an integerized sinusoidal grid. The first step in preprocessing these data for the MODIS download tool is to reproject them into a UTM zone 15 North projection using the MODIS reprojection tool [10], which outputs reprojected data in geoTiff files. First, image scenes covering the Iowa state are mosaicked into one scene. Next, the images are subset to the region of interest (state boundaries). Finally, the data are rescaled from 16-bit into 8-bit and output in generic binary format. The scaling procedure depends upon the valid data range for the product being processed and are given in Table1.

**Producing 250m NDVI:** Though 250m NDVI 16-day composite images are readily available (e.g., EROS Data Center), several applications, such as, biomass monitoring and change characterization requires analyzing images at a finer temporal resolutions. The 250m NDVI data are derived from 250m Level 1B radiances downloaded from the Goddard DAAC (http://ladsweb.nascom.nasa.gov/) or the MODIS Direct Broadcast at the University of Wisconsin-Madison Space Science and Engineering Center (http://eosweb.ssec.wisc.edu/cgi-bin/eosdb.cgi). The 250m Radiance and Geolocation product is delivered as unprojected 5-minute swath segments. The normalized difference vegetation index (NDVI) is calculated as

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{1}$$

where NIR are near infrared d and red band reflectance values [6]. Finally, any x number of days maximum value composite is produced by selecting the highest NDVI value for each pixel over the compositing period.

## 2.1 Atmospheric Correction

The continental United States biweekly MODIS dataset acquired from the USGS EROS Data Center (EDC) is, in a sense, atmospherically corrected by means of the Maximum Value Compositing (MVC) algorithm used in its production. This approach effectively eliminates most of the contamination. However, it also creates a bias toward higher NDVI values at both the beginning and end of the crop growing season. To solve this problem, we have adopted a simple but effective method developed by [12], which is based on a Fourier transformation [8].

Year-wise biweekly NDVI images were adjusted by using a weighted least-squares fitting of a Fourier series as shown in Figure 1. An example cloud-contaminated image is shown in Figure 2, with the corresponding Fourier-adjusted image shown in Figure 3.

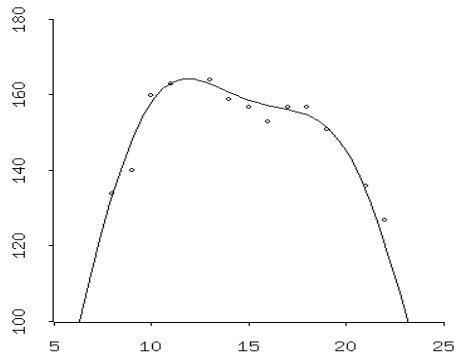



**Figure 1: Weighted least squares fit (smooth curves) of Fourier series**

# 3. GMM CLUSTERING

In this section we present a Gaussian Mixture Model (GMM) based clustering algorithm. First, we generated 1000 random point locations. From each of these locations, we extracted time series data from 2001 and 2006 MODIS NDVI images. We assume that these samples were drawn from a GMM. Now our objective is find the model parameters using the time series data. As we do not know the number of clusters, our second objective is to find the number of clusters as well. We now present solution to these two problems.

## 3.1 Estimating the GMM Parameters

First we solve the model parameter estimation problem by assuming that the training dataset $D_j$ is generated by a finite Gaussian mixture model consisting of $M$ components. If the labels for each of these components were known, then problem simply reduces to usual parameter estimation problem and we could have used MLE. We now describe a parameter estimation technique that is based on the well-known expectation maximization algorithm. Let us assume that each sample $x_j$ comes from a super-population $D$, which is a mixture of a finite number ($M$) of clusters, $D_1, \ldots, D_M$, in some proportions $\alpha_1, \ldots, \alpha_M$, respectively, where $\sum_{i=1}^{M} \alpha_i = 1$ and $\alpha_i \geq 0 (i = 1, \ldots, M)$. Now we can model the data $D = \{x_i\}_{i=1}^{n}$ as being generated independently from the following mixture density.

| Data Set | Scaled from | Scaled to |
|---|---|---|
| MODIS 500m, 8-day Surface Reflectance (MOD09A1) | 0.0 to 0.15 (bands 1,3,4)<br>0.0 to 0.5 (bands 2, 5, 6)<br>0.0 to 0.3 (band 7) | 1 to 255 (0 = fill value) |
| MODIS 500m, 16-day Surface Vegetation Index (MOD13A1) | -0.2 to 1.0 | 1 to 255 (0 = fill value) |
| MODIS 100m, 8-dat LAI/fPAR Product (MOD15A2) | 0 to 100 | 1 to 100 (255 = fill value) |
| MODIS 250m, 15-day NDVI (derived from MOD02QKM) | -0.2 to 1.0 | 1 to 255 (0 = fill value) |

**Table 1: Scaling factors for MODIS data sets**

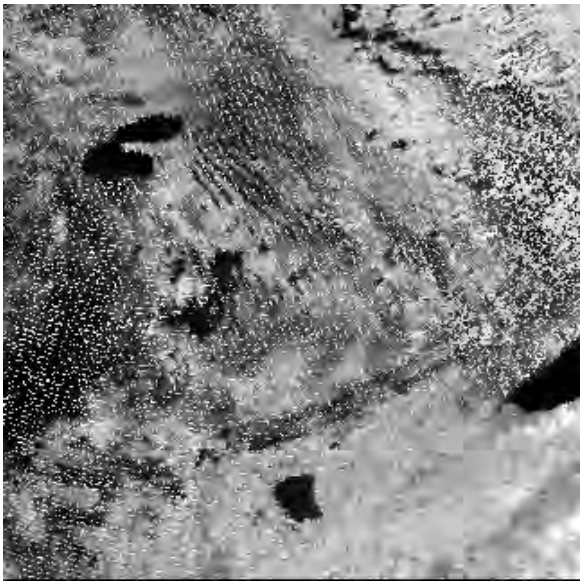


**Figure 2: Raw NDVI image contaminated by cloud cover**

**Figure 3: Fourier-adjusted image**

$$p(x_i|\Theta) = \sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \qquad (2)$$

$$L(\Theta) = \sum_{i=1}^{n} \ln \left[ \sum_{j=1}^{M} \alpha_j p_j(x_i|\theta_j) \right]. \qquad (3)$$

Here $p_j(x_i|\theta_j)$ is the pdf corresponding to the mixture $j$ and parameterized by $\theta_j$, and $\Theta = (\alpha_1, \ldots, \alpha_M, \theta_1, \ldots, \theta_M)$ denotes all unknown parameters associated with the $M$-component mixture density. The *log-likelihood* function for this mixture density is given in 3. In general, Equation 3 is difficult to optimize because it contains the ln of a sum term. However, this equation greatly simplifies in the presence of unobserved (or incomplete) samples. We now simply proceed to the expectation maximization algorithm, interested reader can find detailed derivation of parameters for GMM in [1]. The expectation maximization (EM) algorithm at the first step maximizes the expectation of the *log-likelihood* function, using the current estimate of the parameters and conditioned upon the observed samples. In the second step of the EM algorithm, called maximization, the new estimates of the parameters are computed. The EM algorithm iterates over these two steps until the convergence is reached. For multivariate normal distribution, the expectation $E[.]$, which is denoted by $p_{ij}$, is the probability that Gaussian mixture $j$ generated the data point i, and is given by:

$$p_{ij} = \frac{\left|\hat{\Sigma}_j\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_j)^t \hat{\Sigma}_j^{-1}(x_i-\hat{\mu}_j)\right\}}}{\sum_{l=1}^{M} \left|\hat{\Sigma}_l\right|^{-1/2} e^{\left\{-\frac{1}{2}(x_i-\hat{\mu}_l)^t \hat{\Sigma}_l^{-1}(x_i-\hat{\mu}_l)\right\}}} \qquad (4)$$

The new estimates (at the $k^{th}$ iteration) of parameters in terms of the old parameters at the M-step are given by the following equations:

$$\hat{\alpha}_j^k = \frac{1}{n} \sum_{i=1}^{n} p_{ij} \qquad (5) \qquad\qquad \hat{\mu}_j^k = \frac{\sum_{i=1}^{n} x_i p_{ij}}{\sum_{i=1}^{n} p_{ij}} \qquad (6)$$

$$\hat{\Sigma}_j^k = \frac{\sum_{i=1}^n p_{ij}(x_i - \hat{\mu}_j^k)(x_i - \hat{\mu}_j^k)^t}{\sum_{i=1}^n p_{ij}} \quad (7)$$

We can now put together these individual pieces into the following algorithm (Table 2) which computes the parameters for each component in the finite Gaussian mixture model that generated our training dataset $D$.

**Inputs:** $D_j$, training dataset; M, the number of clusters

**Initial Estimates:** Do clustering by K-Means, and estimate initial parameter using MLE, to find $\hat{\theta}$

**Loop:** While the complete data *log-likelihood* improves:

**E-step:** Use current classifier to estimate the cluster membership of each data sample, i.e., the probability that each Gaussian mixture component generated the given sample point, $p_{ij}$ (see Equation 4).

**M-step:** Re-estimate the parameter, $\hat{\theta}$, given the estimated Gaussian mixture component membership of each data sample (see Equations 5, 6, 7)

**Output:** Parameter vector $\Theta$.

**Table 2: Algorithm for Computing Parameter of Finite Gaussian Mixture Model Over Unlabeled Training Data**

## 3.2 Estimating the Number of Clusters

In the previous section, we described a general technique to estimate GMM parameters for any arbitrary number of $M$-components (clusters) as long as there are sufficient number of samples available for each component and the covariance matrix does not become singular. Then the question remains, which $M$-component model is better? This question is addressed in the area of model selection literature, where the objective is to chose a model that maximizes a cost function. There are several cost functions available in the literature, most commonly used measures are Akaike's information criterion (AIC), Bayesian information criteria (BIC), and minimum description length (MDL). A review of these methods can be found in [9]. The common criteria behind these models is to penalize the models with additional parameters, so BIC and AIC based model selection criteria follows the principal of parsimony. In this study we considered BIC as a model selection criteria, which is also takes the same form as MDL. We also chose BIC, as it is defined in terms of maximized log-likelihood which any way we are computing in our parameter estimation procedure defined in the previous section. BIC can be defined as

$$BIC = MDL = -2\log L(\Theta) + m\log(N) \quad (8)$$

where $N$ is the number of samples and $m$ is the number of parameters. For finding the number of clusters, we applied the GMM clustering algorithm (Table 2) for different values of M. The best model for which BIC is maximum gives us the number of clusters.

## 3.3 Clustering

. Once the GMM is fitted to the training data, we can use the model to predict labels for each cluster. The assignment of label is carried out using the maximum likelihood (ML) procedure. The discriminant function g(.) given by ML principle is as following:

$$g_i(x) = -\ln|\Sigma_i| - (x-\mu_i)^t|\Sigma_i|^{-1}(x-\mu_i) \quad (9)$$

For each pixel (feature vector), we assign a cluster label $i$, if $g_i(x)$ is maximum over all cluster labels.

# 4. CHANGE ANALYSIS

In this section we present simple techniques to analyze the changes between 2001 and 2006 MODIS NDVI images. First, we applied the GMM clustering algorithm presented in the previous section on 1000 samples extracted from each of these time-series images. Both 2001 and 2006 NDVI sample data have yielded a GMM with 10 components each. We then applied clustering technique (Section 3.3) to assign cluster labels each pixel in the Iowa MODIS time series image. Figure 4 shows the clustering results on 2001 image, (a) shows the raw image, (b) shows the BIC values, (c) shows the bivariate density plot of the GMM model for which BIC is maximum (10 components), and (d) shows the clustered image. Figure 5 shows the corresponding results for 2006 MODIS NDVI time series image data.

In order to identify changes, we need to establish the correspondence between clusters found in image 2001 to the clusters found in 2006. We now describe a novel technique to establish the correspondence between the Gaussian mixture models learned using the sample training data. We used KL divergence measure to compute the distance between component Gaussians in both models. Table 3 shows the pair-wise KL divergence across GMM components found in 2001 and 2006 models.

We have chosen the shortest distance as the cluster match. Table 4 shows the cluster matching based on shortest distance metric. Figure 6 shows the time series plots (cluster means) between two clusters matched.

| | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 1 | 3 | 288.33 |
| 2 | 2 | 5 | 87.32 |
| 3 | 3 | 3 | 734.09 |
| 4 | 4 | 4 | 340.51 |
| 5 | 5 | 7 | 943.44 |
| 6 | 6 | 7 | 1585.55 |
| 7 | 7 | 7 | 1054.03 |
| 8 | 8 | 8 | 1233.87 |
| 9 | 9 | 9 | 1163.46 |
| 10 | 10 | 10 | 52869.90 |

**Table 4: Matched Clusters Based on Shortest KLD**

We selected a threshold of 1000 on KL divergence and treated cluster pairs above this threshold as possible changes. The cluster pairs starting with number 6 in the Table 4 were

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 295.28 | 361.69 | 288.33 | 564.45 | 473.89 | 570.04 | 712.52 | 476.13 | 863.47 | 642.34 |
| 2 | 0.00 | 370.34 | 106.00 | 360.65 | 87.32 | 217.35 | 419.58 | 200.69 | 583.69 | 288.38 |
| 3 | 0.00 | 0.00 | 734.09 | 753.69 | 1101.01 | 950.31 | 867.11 | 883.32 | 982.57 | 1250.18 |
| 4 | 0.00 | 0.00 | 0.00 | 340.51 | 638.52 | 629.34 | 592.87 | 524.58 | 760.54 | 781.54 |
| 5 | 0.00 | 0.00 | 0.00 | 0.00 | 1507.34 | 1151.43 | 943.44 | 1051.31 | 993.30 | 1525.18 |
| 6 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1996.87 | 1585.55 | 1849.75 | 1643.70 | 2770.96 |
| 7 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1054.03 | 1151.13 | 1149.99 | 1680.90 |
| 8 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1233.87 | 1345.55 | 1811.13 |
| 9 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1163.46 | 1377.39 |
| 10 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 52869.90 |

**Table 3: KL divergence between 2001 and 2006 GMMs**

identified as possible changes. We used a simple technique to identify these cluster pairs in the images. These pairs were accounted for about 32% of the total pixels in the image. The time series pair at each pixel location is then further analyzed based on the following three phenological events: green-up onset, peak/maturity, and dormancy onset. We found key points in the time series based on the notion of major inclines concept used in time series data compression [2]. These points were then used to compare to compare the difference in two dates. Our analysis shows that at 35% of locations, the green-up points between 2006 happened earlier than the 2001. 50% of locations differed in the peak NDVI (maturity) of which more than 60% 2006 occurred before 2001. We also observed slight increase in growing season at 18% of locations. We believe these changes are important in understand changes in phenology. However, further research is required to find the robustness of these techniques. Especially there is a need to establish the onset periods with ground truth data which is sparse and aggregated at best. If there is strong correlation between ground observations and NDVI based phenological events, then it may be possible to scale the algorithms presented in this paper to large geographic regions and provide fine-grained and accurate information on crop growing patterns and changes across the globe.

## 5. CONCLUSIONS

In this study we presented an unsupervised change detection technique. We used well known EM technique to find GMM parameters and Bayesian Information Criteria (BIC) for determining the number of clusters. The correspondence between the GMM components across two different years is established through the KL Divergence measure. Shortest KL divergence is used to establish the cluster correspondence. A threshold is used to identify the cluster which are different between the two years. The changes identified were further analyzed for understanding phenological events. Our study shows interesting relationships between key phenological events such as onset, length, end of growing seasons. Further research is needed to extend the approach across large geographic regions and to correlate the changes identified with the ground truth.

## 6. ACKNOWLEDGMENTS

The comments and inputs of E. Bright and B. Bhaduri have greatly improved the technical accuracy and readability of this paper. Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

## 7. REFERENCES

[1] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report, University of Berkeley, ICSI-TR-97-021, 1997., 1997.

[2] E. Fink, K. B. Pratt, and H. S. Gandhi. Indexing of time series by major minima and maxima. In *Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 2332–2335, 2003.

[3] S. Graham and C. Parkinson. Minutes of the aqua science working group meeting. *The Earth Observer*, 13(2), March/April 2001.

[4] C. O. Justice, E. Vermote, J. R. Townshend, R. Defries, D. P. Roy, D. K. Hall, V. V. Salomonson, J. L. Privette, G. Riggs, A. Strahler, W. Lucht, R. B. Myneni, Y. Knyazikhin, S. W. Running, S. W. Steve W. Nemani, Z. Wan, A. R. Huete, W. van Leeuwen, R. E. Wolfe, L. Giglio, J.-P. Muller, P. Lewis, and M. J. Barnsley. The moderate resolution imagin spectrradiometer (modis): Land remote sensing for global chang research. *IEEE Transactions on Geosciences and Remote Sensing*, 36:1228–1249, 1998.

[5] S. R. Karlsen, A. Tolvanen, E. Kubin, J. Poikolainen, K. A. H£gda, B. Johansen, F. S. Danks, P. Aspholm, F. E. Wielgolaski, and O. Makarova. Modis-ndvi-based mapping of the length of the growing season in northern fennoscandia. *International Journal of Applied Earth Observation and Geoinformation*, 10(3):253 – 266, 2008.

[6] T. M. Lillesand and R. W. Kiefer. *Remote Sensing and Image Interpretation.* John Wiley and Sons, Inc, New York, 2000.

[7] T. Sakamoto, M. Yokozawa, H. Toritani, M. Shibayama, N. Ishitsuka, and H. Ohno. A crop phenology detection method using time-series modis data. *Remote Sensing of Environment*, 96(3-4):366 – 374, 2005.

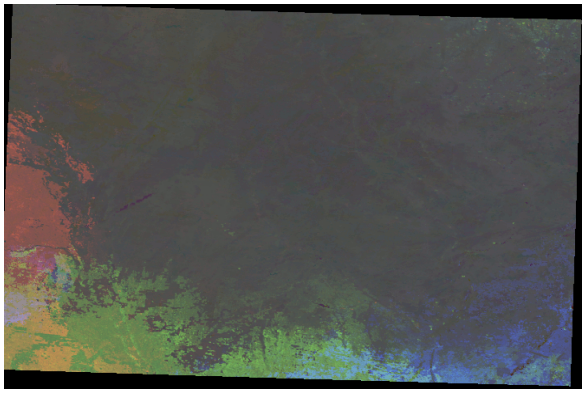
[8] P. Sellers, C. Tucker, G. Collatz, S. Los, C. Justice, D. Dazlich, and D. Randall. A global $1^o$ by $1^o$ ndvi data set for climate studies. part 2: The generation of global fields of terrestrial biophysical parameters from the ndvi. *International Journal of Remote Sensing*, 15:3519–3545, 1994.

[9] E. J. Ward. A review and comparison of four commonly used bayesian and maximum likelihood model selection tools. *Ecological Modelling*, 211:1–10, 2008.

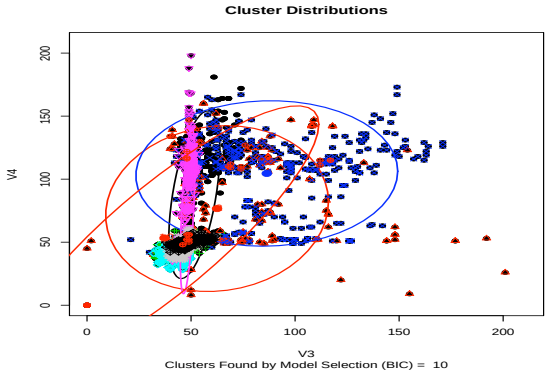
[10] J. M. Weiss, T. Logar, G. Stubbendieck, J. Sneller, R. Burrell, and G. Schmidt. *MODIS Reprojection Tool User's Manual, Release 2.3*. Department of Mathematics and Computer Science, South Dakota School of Mines and Technology, 2002.

[11] M. A. White and R. R. Nemani. Real-time monitoring and short-term forecasting of land surface phenology. *Remote Sensing of Environment*, 104(1):43 – 49, 2006.

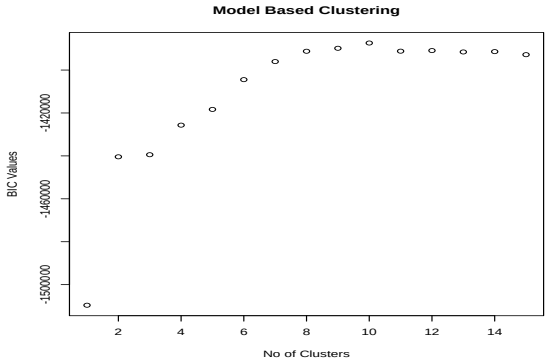
[12] B. Wilson and T. Burk. Fourier adjustment of ndvi time-series data. Department of Forest Resources, University of MN, TR-01/00.



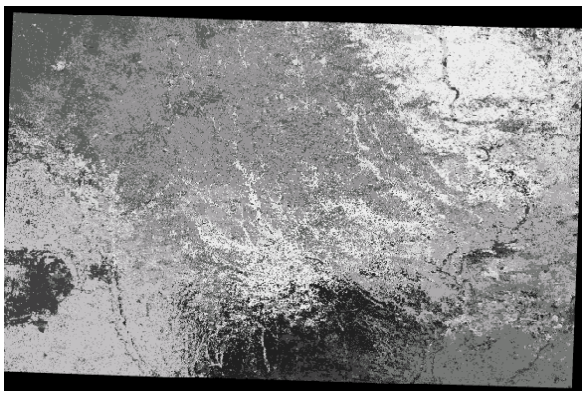
(a) MODIS NDVI 2001




(b) GMM; Bivariate Density Plots

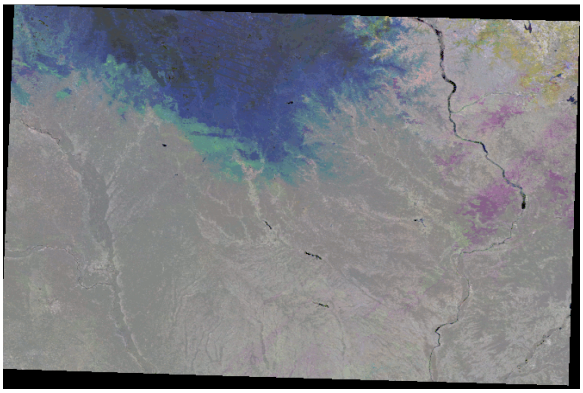



(c) GMM; BIC, Max at 10 components

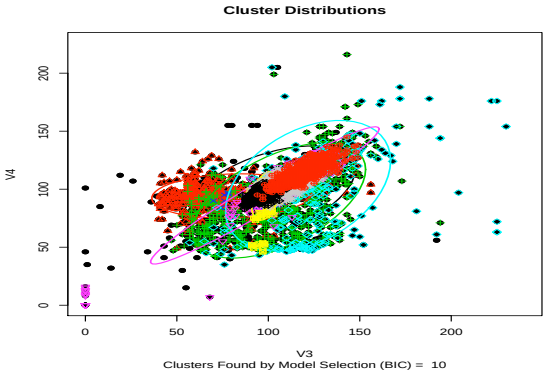


(d) GMM; Clustered Image

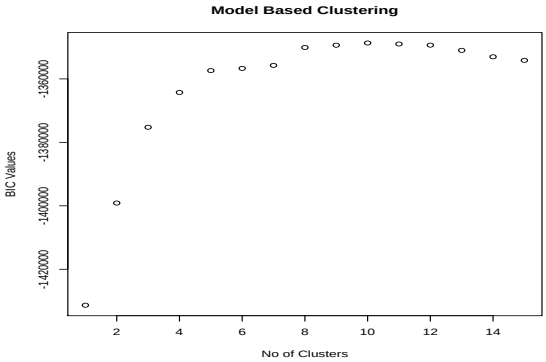
**Figure 4: GMM Clustering - MODIS 2001**



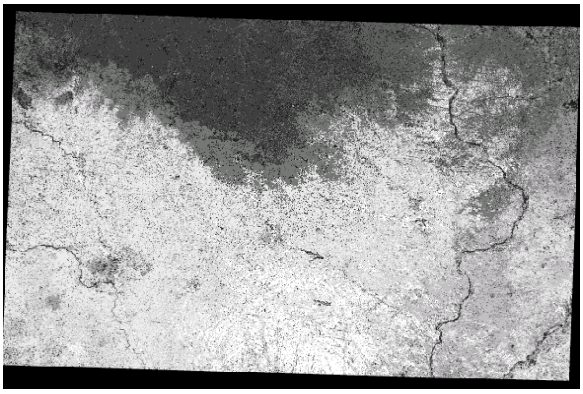
(a) MODIS NDVI 2006




(b) GMM; Bivariate Density Plots




(c) GMM; BIC, Max at 10 components



(d) GMM; Clustered Image

**Figure 5: GMM Clustering - MODIS 2006**

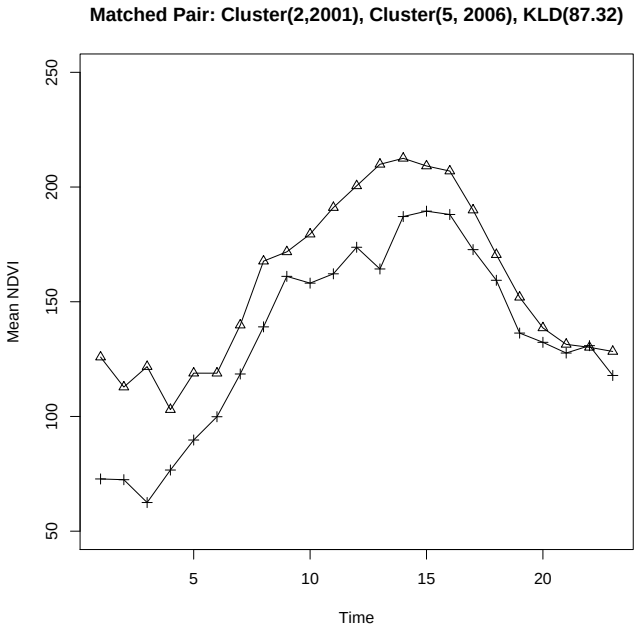



**Figure 6: Time series plot of two matched clusters**

# Mining in a Mobile Environment

Sean McRoskey
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, Indiana 46556
smcroske@nd.edu

James Notwell
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, Indiana 46556
jnotwell@nd.edu

Nitesh V. Chawla
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, Indiana 46556
nchawla@cse.nd.edu

Christian Poellabauer
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, Indiana 46556
cpoellab@cse.nd.edu

## ABSTRACT

Distributed PRocessing in Mobile Environments (DPRiME) is a framework for processing large data sets across an ad-hoc network. Developed to address the shortcomings of Google's MapReduce outside of a fully-connected network, DPRiME separates nodes on the network into a master and workers; the master distributes sections of the data to available one-hop workers to process in parallel. Upon returning results to its master, a worker is assigned an unfinished task. Five data mining classifiers were implemented to process the data: decision trees, k-means, k-nearest neighbor, Naïve Bayes, and artificial neural networks. Ensembles were used so the classification tasks could be performed in parallel. This framework is well-suited for many tasks because it handles communications, node movement, node failure, packet loss, data partitioning, and result collection automatically. Therefore, DPRiME allows users with little knowledge of networking or distributed systems to harness the processing power of an entire network of single- and multi-hop nodes.

## Keywords

Ad-hoc network, classifier, data mining, MapReduce, ensemble.

## 1. INTRODUCTION

Technology is exhibiting a trend toward wireless. For many, standards such as Bluetooth and Wi-Fi are replacing their wired counterparts, and powerful mobile devices are more popular than ever [1]. In other areas, there is a desperate need for real-time data analysis and extraction [2]. A government agency may need to quickly evaluate camera data from an airport to determine if it is indicative of terrorist activity. In another scenario, a financial institution may need to check a customer's credit card transaction for evidence of fraud. These data mining applications demand an immediate response. Processing large amounts of data on the mobile devices that are present in these areas, however, is taxing on their processors, memory, and batteries.

MapReduce is Google's solution to processing vast amounts of data on many commodity machines. It does so by dividing the machines into a master and many workers; the master splits the work into many small pieces and is responsible for coordinating the processing of each piece. MapReduce is an example of distributed processing; by employing the resources of the entire network, several less-powerful devices can outperform a single device with greater processing capability and resources. Unfortunately, MapReduce possesses several attributes that prevent it from being deployed in mobile environments. For this reason, we developed DPRiME, which utilizes this approach by partitioning the data into manageable pieces and distributing them among available wireless devices.

Much like Google's MapReduce framework, DPRiME represents an abstract framework and associated implementation for processing large data sets in a distributed environment. The nature of mobile environments precludes several important features of MapReduce, such as a fully-connected network and shared storage space, from being implemented on wireless devices. This served as the impetus for developing DPRiME as a framework for addressing the same problem with a mechanism suited for wireless ad-hoc networks. DPRiME retains the relevant features of MapReduce, with several necessary additions. Because DPRiME automates data partitioning, task management, and communications, it only requires the user to define the processing task, data, and processing parameters during run-time. Thus, no knowledge of the underlying workings of the system is required.

### 1.1 MapReduce

#### 1.1.1 Framework

Google developed MapReduce to simplify distributed computing by separating data processing functions from the underlying distribution and parallelization tools. With MapRe-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09* June 28, 2009, Paris, France
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

duce, users simply specify a Map and Reduce function, and the framework handles data partitioning, task assignment, fault tolerance, and other aspects of the distributed computing process. In this way, a user with no knowledge of distributed systems could still harness the capabilities of a large computing cluster [3].

The MapReduce process consists of two main parts: the Map task, which takes a set of input key/value pairs and produces an intermediate set of key/value pairs; and the Reduce task, which merges all of the values for each distinct intermediate key. An example of an input pair could be a word to count and a string in which to search for it; in this case, the intermediate pair would consist of this word and its count. The reduce task would then produce a count for all map tasks that searched for this word.

After the user specifies Map and Reduce functions, the framework designates one machine as the master and the rest as workers. MapReduce begins by partitioning the data into manageable pieces. The master, which coordinates the map and reduce tasks, assigns pieces of the data, as well as the Map task, to idle workers. The Map workers parse the data and pass them to the Map function, which produces intermediate key/value pairs that are periodically written to memory; pointers to the intermediate pairs are then passed back to the master. When the master receives a pointer, it relays it to a Reduce worker; the Reduce worker reads the remotely stored key/value pairs into memory, sorts it by intermediate key, and passes the intermediate values for corresponding keys to the reduce function, which produces an output for each intermediate key.

Several features within this process distinguish MapReduce as a robust and efficient model: fault tolerance, task granularity, and back-up. To address worker failure, the master periodically pings its workers. If it does not receive a response, it labels the worker as failed, cancels any task the worker is processing, and resets the worker to an idle state; the task the worker was processing is then reassigned. Task granularity encompasses the idea of splitting the data into many fine pieces. Doing so improves dynamic load balancing and also reduces the costs of worker failure. Back-up is implemented when MapReduce nears completion. It consists of the master assigning all currently running tasks to idle workers; a task is completed when either the primary or back-up worker finish. Back-up helps to mitigate the effects of stragglers, machines that take an especially long time to complete one of the final Map or Reduce tasks [3].

#### *1.1.2 Limitations*

MapReduce was designed and optimized to operate on large clusters of commodity machines that are fully connected over Ethernet. It reduces bandwidth usage by relaying data locations from Map workers to Reduce workers through the master; the Reduce workers then read the remotely stored data into memory. In doing so, MapReduce relies on shared memory and fully-connected machines. These features are not possible in mobile environments, in which data can only be shared by physically passing data over the network. In addition, every node in the network is not connected to every other node, requiring data to be transmitted to several nodes before reaching its intended destination. For these reasons, we found it unproductive to implement MapReduce for wireless environments.

## 2. DPRIME

To address the shortcomings of MapReduce in mobile environments, we found it necessary to develop a distributed framework that retained MapReduce‘s key features while incorporating communication and storage optimized for a mobile environment. DPRiME uses a similar approach to MapReduce—a master assigns portions of a data set to available workers, which process them using an assigned function. Several important features, however, distinguish DPRiME and make it ideal for deployment in mobile environments.

For this paper, DPRiME was used to distribute a classification task across a network, but it could easily be used to distribute many other types of jobs. It has the benefit of being able to process data stored on a single node or distributed on nodes throughout the network.

### 2.1 Master Functionality & Worker Discovery

A mobile device is designated as the master when the user specifies an input data set and processing parameters. Unlike the fully-connected computers that compose a cluster, the ad-hoc network used in a mobile environment prevents the master from knowing how many workers, if any, are available at a specific time. To initiate the process, the master broadcasts a PING message and listens for responses from workers. After an interval where the master waits for responses, it partitions the data according to the number of available workers and assigns a section to each for processing.

Much like in MapReduce, the master maintains data structures which track workers and assigned tasks. Each worker is tagged with a WorkerTag instance that contains information about the worker id, worker address, worker state, task assigned, and elapsed time from the workerÕs last communication . These allow the master to track workers and tasks throughout the DPRiME process, noting failed workers and unfinished tasks for fault tolerance and backup.

### 2.2 Worker Functionality

Worker nodes, if available, respond to a master PING with an AVAI response. The master then issues the worker a command packet (CMND) containing data and processing parameters. The master controls data processing by passing a task identification number, from a group of pre-programmed tasks, along with any required parameters for that task. In addition, the master can specify whether the worker should process locally stored data by passing the name of the file where the data would be stored.

### 2.3 Execution Overview

When the user specifies a data set, processing task, and processing parameters, DPRiME proceeds through the following sequence:

1. The master, begins by attempting to open the file containing the data and reading them into memory. If the data are smaller than a threshold value, the master processes them itself.

2. Otherwise, the master pings its one-hop neighbors, waits for responses, partitions the data according to the number of responses, and sends a piece of the data to each available one- hop neighbor.

3. If no neighbors respond to the ping, the master processes the data itself.

4. If there are no data to read into memory, the master sends the name of the file that would contain data to each one-hop neighbor.

5. If there are no data to read into memory and no one-hop neighbors, the task terminates.

When a worker receives a task from its master, it proceeds through the following sequence:

1. If any data were received, the worker begins by examining them. If the data are smaller than a threshold value, it processes them itself.

2. Otherwise, it distributes the task like its master. If there are no one-hop neighbors, the worker processes the data itself.

3. After completing the processing task, the worker processes local data in the same manner as its master.

4. Finally, it returns the results to its master.

Like MapReduce, DPRiME incorporates a back-up functionality; it maintains a list of tasks, through which it iterates. The master continually loops through the list and assigns any unfinished task to idle workers until every task is completed; a task is completed when the first worker replies with its result.

# 3. IMPLEMENTATION

## 3.1 DPRiME Protocol

Communication between devices was implemented in User Datagram Protocol (UDP), with several key acknowledgement and retransmission features built in for reliability. The combination of UDP's unreliable packet delivery and the mobility of workers in an ad-hoc environment necessitated a robust system for efficiently handling node failures and dropped packets.

Throughout the process, various scenarios were addressed which could cause long delays or, in some cases, failure of the process. For instance, a master's failure to notice an available worker at the beginning of the DPRiME process is costly, since data partitioning depends on the number of known idle neighbors. The initial PING is thus sent out several times to ensure the first master-worker communication is established. Once the process has begun, the master periodically broadcasts a ping to check for any new available workers.

Because the command packets (CMND) could be very large in some cases, they are sliced at roughly 8KB intervals and sent as a series to the worker. The worker records each received piece and sends a "command complete" (CMPL) packet to the master upon receiving the entire CMND. If a short timeout elapses before the complete command is received, the worker notifies the master with a receipt packet (RECV) enumerating the missing pieces; the master then resends these fragments.

Just as workers can come into the master's range in the middle of the process, they can just as easily leave. Worker mobility was thus addressed in several ways. When the

**Figure 1: Execution Overview.**

master sends a command, it defines a maximum silence period within which it requires a response from the worker. Workers begin working on a task and send "progress reports" (PROG) according to the maximum silence defined by the master. If a master does not hear from a worker within the maximum silence period, it issues a "hello" packet (HELO) and waits one more timeout period before marking the worker as missing.

Broken connections are expensive in time and processing power if a worker is unable to send the results of a tedious task back to the master. Thus, Dynamic Source Routing was implemented to address this challenge.

## 3.2 Dynamic Source Routing (DSR)

Especially in an ad-hoc environment, workers may frequently move away from their master while working on a task. It is infeasible, however, to simply reassign lost tasks. Besides costing precious time, there is no guarantee that the next worker will finish the same task within range. Therefore, DSR was implemented as a simple means for workers to find a route back to their master, to send both progress reports and results.

After sending a progress report or a results packet, workers expect an acknowledgement from the master (RECV). After three unacknowledged packets, the worker assumes it has lost connection to the master and begins building a new route using DSR. After a new route has been established, all communication from the worker to the master and vice versa includes a "DSR-data" (DSRD) header and is sent along this route. If the worker eventually determines it has lost the master again, it will simply restart the route-building process. In this way, the framework relies on multi-hop routing to ensure delivery of results and to avoid reassigning completed tasks.

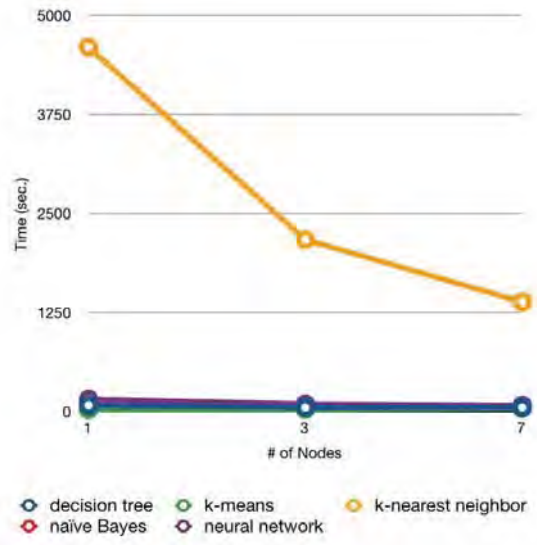
# 4. MINING

To simulate data mining in the field, we implemented five classifiers: Iterative Dichotomiser 3 (ID3) decision trees, k-means, k-nearest neighbor, Naïve Bayes, and artificial neural networks. On their own, these classifiers have been studied and are well-understood [4], [5]; implementing them in parallel, however, is the subject of ongoing research [6], [7], [8].

Initially, we considered developing a parallel implementation of each classifier. For decision trees, each branch would be parallelized as in [8]. Developing a parallelization for each classifier is non-trivial though, and would also limit DPRiME ' s capacity to accommodate new classifiers. For these reason, we chose to split the data, train a classifier on each split, and treat the classifiers as an ensemble.

Ensembles were selected for their ability to generate highly accurate classification results [9]. In our framework, after a classifier had been trained, it would classify each of the testing examples it had received; this took place on the worker nodes. Because only two-class data-sets were used (the framework is capable of processing data sets with any number of classes), the classification results were stored in a 2 x $n$ array, where each row represented a class and each column (0, ..., $n$) represented a testing example. This array was then transmitted to the worker's master, where the arrays representing the classifications from different workers were added together. If this master was a sub-master, it would transmit the sum array to its master, which would do the same. When the results reached the master where the task originated, it would compare the number of votes for each of the two classes and assign each test example the class with the majority vote; ties were decided by randomly assigning a class.

**Figure 2: Classifier completion times.**




# 5. RESULTS

For our experiment, DPRiME was implemented in C# using the .NET Compact Framework. It was deployed on Motorola MC35 Windows Mobile devices. Figure 2 and Figure 3 display the time taken to perform classification tasks for three different scenarios. Both figures display the same information, but Figure 3 does not include the k-nearest neighbor classifier results. Experiments were conducted using three different configurations: one master, one master with two workers, and one master with two workers and two workers for each of the master's workers.

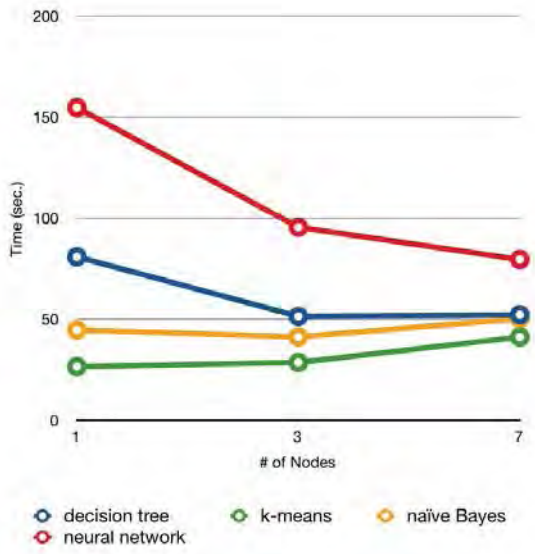
## 5.1 Discussion

Different parameters were used for the mining algorithms. The minimum leaf size for decision trees was two. K-means used five centroids for classification. K-nearest neighbor used three neighbors. Neural networks used 3 layers with 10 iterations and the learning rate was 0.1.

The same imbalanced data set was used for each classifier. It consisted of thirteen thousand examples with sixteen continuous features and two classes. One thousand testing examples were classified.

Not surprisingly, k-nearest neighbor exhibited the greatest improvement when distributed across this framework. This is because, unlike the other classifiers, no model was created for classification. Instead, classification required the entire data set split to be processed for each testing example making it the longest-running classifier so the communication overheads were only a small fraction of total time. K-means, which took the least amount of time, performed worse when distributed across our framework. This is because the additional communication costs outweighed the benefits of using multiple devices to process a task. We conclude that the

**Figure 3: Classifier completion times (without k-nearest neighbor).**




more computationally intensive a task, the more it benefits from using this framework.

## 5.2 Future Work

There is a great deal of work to be done in evaluating the effectiveness of this framework as well as determining parameters for the most effective deployment of DPRiME. Thresholds need to be established for the maximum file size that a worker will process and the size of splits made by a master. In addition, extensive testing will need to be performed to evaluate the time requirements versus accuracy of the different classifiers used in this framework. Finally, the performance gains made by using different combinations of single-hop and multi-hop neighbors need to be compared.

# 6. CONCLUSION

With the advent of ubiquitous wireless sensor networks and improvements in wireless technology, there will be an increasing demand for processing large amounts of data at their source. DPRiME simplifies this process by providing an abstraction that separates the more complicated distribution and communications tasks from the user-defined function that processes the data. By incorporating the important features of MapReduce into a robust framework capable of handling the shortcomings of MapReduce outside of a fully-connected network, we believe DPRiME will become a valuable tool for simplifying the data-intensive tasks in mobile environments.

# 7. ACKNOWLEDGMENTS

We would like to thank Notre Dame's Computer Science and Engineering department for the support, facilities, and equipment used in the development of DPRiME. This work was supported by National Science Foundation Grant CNS-0754933.

# 8. REFERENCES

[1] C. Pettey, "Gartner Says Worldwide Smartphone Sales Grew 29 Percent in First Quarter of 2008," Gartner, 6 June 2008, 30 July 2008 <http://www.gartner.com/it/page.jsp?id=688116>.

[2] B. Thuraisingham, L. Khan, C. Clifton, J. Maurer, and M. Ceruti, "Dependable Real-Time Data Mining," *Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing* (2005).

[3] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large CLusters," *Sixth Symposium on Operating System Design and Implementation* (2004), 15 July 2008 <http://labs.google.com/papers/mapreduce.html>.

[4] T. M. Mitchell, *Machine Learning*, Ed. E. M. Munson, Singapore: The McGraw-Hill Companies, Inc., 1997.

[5] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[6] K. W. Bowyer, L. O. Hall, T. Moore, and N. Chawla, "A Parallel Decision Tree Builder for Mining Very Large Visualization Data Sets," *IEEE International Conference on Systems, Man, and Cybernetics* 3 (2000): 1888-893.

[7] R. Agrawal and J. C. Shafer, "Parallel Mining of Association Rules." *Ieee Trans. On Knowledge And Data Engineering* 8 (1996): 962-69.

[8] G. J. Narlikar, "A Parallel, Multithreaded Decision Tree Builder," (1998).

[9] T. G. Dietterich, "Ensemble Methods in Machine Learning." *Lecture Notes in Computer Science* (2000).

[10] C. Chu, S. K. Kim, Y. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Oluktun, "Map-Reduce for Machine Learning on Multicore."

# SHORT RESEARCH PAPERS

# On the Identification of Intra-seasonal Changes in the Indian Summer Monsoon

Shivam Tripathi and Rao S. Govindaraju
School of Civil Engineering, Purdue University
West Lafayette, Indiana, USA
govind@purdue.edu

## ABSTRACT

Intra-seasonal changes in the Indian summer monsoon are generally characterized by its active and break (A&B) states. Existing methods for identifying the A&B states using rainfall data rely on subjective thresholds, ignore temporal dependence in the data, and disregard inherent uncertainty in their identification. This paper develops a method to identify intra-seasonal changes in the monsoon using a hidden Markov model (HMM) that allows objective classification of the monsoon states. The method facilitates probabilistic interpretation which is especially useful during the transition period between the two monsoon states. The developed method can also be used to - (i) identify monsoon states in real time, (ii) forecast rainfall values, and (iii) generate synthetic data. Comparisons of the results from the proposed model with those from existing methods suggest that the new method is a promising for detecting intra-seasonal changes in the Indian summer monsoon.

## Categories and Subject Descriptors

G.3 [**Probability and statistics**]: Markov processes, probabilistic algorithms; J.2 [**Physical sciences and engineering**]: Earth and atmospheric sciences

## General Terms

Algorithms

## Keywords

Indian summer monsoon, active and break states, hidden Markov models, variational Bayes

## 1. INTRODUCTION

Recent studies on the Indian summer monsoon have evinced that a substantial portion of spatio-temporal variability in rainfall arises from *monsoon intra-seasonal oscillations* (MISOs). MISOs are manifested in the form of prolonged periods of wet and dry states of monsoon, known as *active and break* (A&B) states. The timings and durations of the A&B states have a direct impact on the demand and supply of irrigation water, and hence their timely forecast can facilitate agricultural planning and water management in the monsoon region. Such forecasts also assist in reducing the number and magnitude of artificial floods that are created by sub-optimal operation of reservoirs.

The first step in building a forecast model for the A&B states is the identification of these states. Identification is also essential for understanding the mechanisms of genesis and propagation of MISOs. More than 50 years of research on the identification of A&B states has resulted in numerous definitions using different variables (rainfall, winds, pressure, outgoing long-wave radiations), involving different regions (all India, central India, core monsoon region), during different periods (July-August, June-September), and applying different thresholds. Over the last few years, a consensus is emerging on identifying the A&B states using rainfall data because of its direct bearing on the socio-economic fabric of the society. Many identification methods based on rainfall data have been recently proposed [7, 22, 20, 17, 13, 21]. However, there are a few common deficiencies in most of these methods:

(a) Choice of arbitrary thresholds and regions. This drawback can be attributed to the lack of a quantifiable objective function in them.

(b) Negligence of spatio-temporal dependence in the rainfall data.

(c) Avoidance of accounting inherent uncertainty in identifying the A&B states.

This study develops a method to identify the A&B states from the rainfall data using a hidden Markov model (HMM; Rabiner [19]) with an intent to overcome these limitations. The developed method performs real-time predictive analysis of geographically distributed rainfall data. The method is developed in a Bayesian framework that provides a probabilistic approach to address the problem, and offers a basis to develop methods for:

(a) identifying A&B states in real-time;

(b) generating synthetic rainfall data that can be used in designing water management policies; and

(c) forecasting A&B states.

To evaluate the performance of the developed model, it is compared with a recently proposed A&B state identification method by Rajeevan et al. [21] from National Climate

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09,* June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

Center (NCC) of the India meteorological department (the model will be referred to as the *NCC model* in this paper). The remainder of the paper is structured as follows - (i) the data used in the study are described first, (ii) the mathematical formulation of the HMM is then outlined, (iii) the steps involved in developing the model are described, (iv) the results obtained are discussed, and finally (iv) conclusions are presented.

## 2. DATA USED IN THE STUDY

Rajeevan et al. [20] developed a daily gridded rainfall data set for India with $1^\circ \times 1^\circ$ spatial resolution, and extends from 1951 to 2003 (later extended to 2004). The gridded data set was obtained by interpolating data from 1803 raingages spread over India that have at least 90% availability during the analysis period.

The data set prepared by Rajeevan et al. [20] is now considered to be a standard data set and has been widely used in monsoon related studies [8, 24, 11]. The India meteorological department is currently working on increasing the number of rain gage stations in the analysis, and also planning on providing this data set in real-time [21]. The latter development will allow the use of the model developed in this study for practical applications requiring on-line identification and forecasting of the A&B states.

## 3. HIDDEN MARKOV MODEL

The HMM is a statistical model in which observations from a system are assumed to be conditioned on the state of the system which is hidden and follows Markov property. The HMM was developed in late 1960s and early 1970s for speech recognition, since then it has been successfully used in many applications including hydrology [10, 26]. In the following sub-section, the mathematical formulation of the HMM and its adaption for identifying the monsoon states are briefly presented. Readers are referred to [19] and [4] for additional details on the HMM.

### 3.1 HMM for identifying monsoon states

Let the rainfall at time $t$ be denoted by $x_t$, $t = 1, \ldots, N$ $\{x_t \in \Re$ and $\boldsymbol{x} = [x_1, \ldots, x_N]^{\mathrm{T}}\}$. In a standard HMM, the rainfall $x_t$ is assumed to depend only on the state variable $\boldsymbol{z}_t$ $\{\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^{\mathrm{T}}\}$ that denotes state of the monsoon, is hidden (not observed) and follows the first order Markov property. The state variable $\boldsymbol{z}_t$ is a $K$-dimensional binary random variable in which only one element $z_{tk}$ is equal to unity and all other variables are equal to zero, i.e. $z_{tk} = \{0, 1\}$ and $\sum_k z_{tk} = 1$. If the number of states, $K$, are known *a priori*, the standard HMM can be parameterized using the following three distributions:

(a) The conditional distribution of the rainfall given the monsoon state, $p(x_t|\boldsymbol{z}_t)$, referred to as the *emission distribution*.

(b) The conditional distribution of the present monsoon state given the previous state i.e. $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$. Because $\boldsymbol{z}_t$ is a $K$ dimensional binary variable, the conditional distribution is given by a $K \times K$ transition matrix $\boldsymbol{A}$ whose element $A_{jk} = p(z_{tk} = 1|z_{t-1,j} = 1)$.

(c) The marginal distribution of the monsoon state at the first time step, $p(\boldsymbol{z}_1)$, given by a $K$ dimensional vector $\boldsymbol{\pi}$ whose element $\pi_k = p(z_{1k} = 1)$.

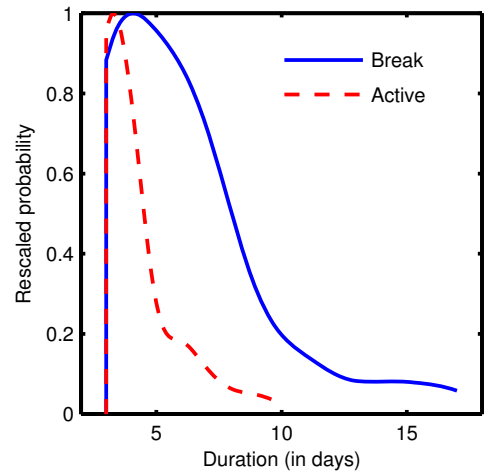



**Figure 1: Distribution of the duration of the active and the break states identified by the NCC model in the core monsoon region.**

For modeling daily rainfall data, the emission distribution is commonly chosen to be a Gamma distribution or a mixture of Exponential distributions [3, 9]. The parameters of the HMM which are essentially the parameters of the distributions $p(x_t|\boldsymbol{z}_t)$, $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$, and $p(\boldsymbol{z}_1)$ can be estimated by using the *maximum likelihood* (ML) method [4].

A major limitation of the standard HMM in modeling daily monsoon rainfall is that the *state duration* is a Geometric distribution given by

$$p_k(\Delta T) = (A_{kk})^{\Delta T}(1 - A_{kk}) \propto \exp(\Delta T \ln(A_{kk})) \qquad (1)$$

where $\Delta T$ is the number of time steps (or duration) for which the system remains in state $K$. This exponentially decaying distribution may not be a realistic model for the duration of the A&B states in the Indian summer monsoon rainfall. Figure 1 shows the distribution of durations of the A&B states as defined by the NCC model [21]. Clearly, these distributions are significantly different from the Geometric distribution. Hence, the standard HMM should be modified to provide more flexibility in modeling state-duration. This flexibility can be achieved by either using Hidden semi-Markov model (HSSM; [15]) or expanded state HMM (ESHMM; [23, 12]).

In this study, both HSSM and ESHMM were considered, and a preliminary investigation of the models on the monsoon data suggested that both yield qualitatively similar results. The ESHMM was selected for detailed study because of its simplicity [5]. In the ESHMM, a state is associated with sub-states each having the same emission probability. The sub-states can be arranged in different ways and are characterized by their topology. The *type A* topology, as defined by [23] and shown in Fig. 2, was used here. In this topology, the self-transition probabilities within a sub-state are constrained to be equal. The distribution of the state duration for an ESHMM with a *type A* topology is given by a convolution of the Geometric distribution also known as *negative Binomial distribution*, and it offers the desired flexibility in modeling state duration with minimal increase in number of parameters [5]. The only additional parameter,

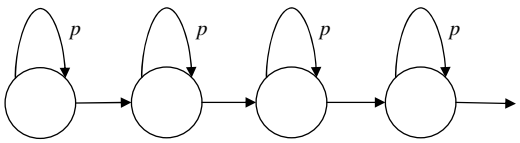



**Figure 2: Type A topology for an expanded state HMM. Circles denote sub-states within a monsoon state, and $p$ is the self-transition probability shared by all the sub-states.**

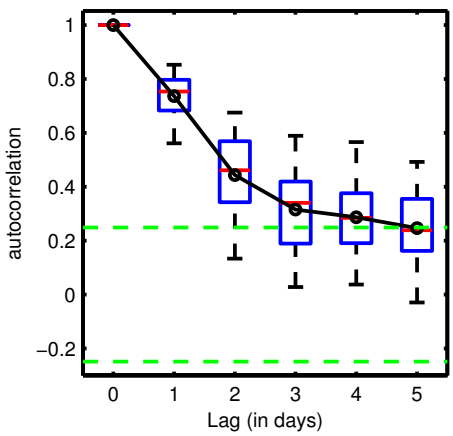



**Figure 3: Autocorrelation in the observed rainfall time series of the core monsoon region. The Circles denote the mean of the autocorrelation in the July and August rainfall during 1951 to 2004. The horizontal line in the middle of the box represents median, and the top and bottom of the box ranges from 25 to 75 percentile while the whiskers extend to 10 and 90 percentile. The dashed lines denote 95% confidence interval.**

compared to the standard HMM, is the number of sub-states ($S_K$).

Another limitation of the standard HMM is that it is poor in capturing long range dependence in the observed rainfall. Figure 3 shows the auto-correlation of the rainfall time-series over the core monsoon region used in the NCC model. Evidently, correlations up to four days lag are significant. These longer lag correlations cannot be captured by the standard HMM because they are mediated by the first-order Markov chain of hidden sates [4]. An alternative is a generalized HMM referred to as *autoregressive hidden Markov model* (ARHMM; [6]). This study employs an ARHMM with Gaussian emission density in which the rainfall at time $t$, $x_t$, depends not only on the monsoon state $\boldsymbol{z}_t$, but also on the the previous rainfall values $\breve{\boldsymbol{x}}_t = [x_{t-1}, \ldots, x_{t-L}]^{\mathrm{T}}$ where $L$ is the number of lags. The emission distribution is given by

$$p(x_t|\boldsymbol{z}_t, \breve{\boldsymbol{x}}_t) = \prod_{k=1}^{K} \mathcal{N}(x_t|\boldsymbol{w}_k^{\mathrm{T}} \breve{\boldsymbol{x}}_t, \beta_k^{-1})^{z_{tk}} \tag{2}$$

where $\boldsymbol{w}_k = [w_1, \ldots, w_L]^{\mathrm{T}}$ is the weight vector and $\beta_k$ is the precision term, both associated to the monsoon state $k$.

The ML method commonly used to estimate parameters of a HMM is not particularly suitable in the present context because the modeling effort aspires to - (i) account for the inherent uncertainty in identifying the A&B states, (ii) identify the A&B states in an online mode which will require updating the model parameters in real time, and (iii) incorporate prior knowledge of MISOs into the HMM model. These requirements can be met by estimating parameters in the Bayesian framework, which is briefly described in the following subsection.

## 3.2 Bayesian parameter estimation for autoregressive HMM

The joint distribution of the observed rainfall and the hidden monsoon states according to an ARHMM model is

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{Z}|\Theta) =& p(\boldsymbol{z}_1|\boldsymbol{\pi}) \prod_{t=2}^{N} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}, \boldsymbol{A}) \\ & \prod_{t=1}^{N} \prod_{k=1}^{K} p(x_t|z_{tk}, \breve{\boldsymbol{x}}_t, \boldsymbol{w}_k, \beta_k) \end{aligned} \tag{3}$$

where $\Theta = \{\boldsymbol{A}, \boldsymbol{\pi}, \boldsymbol{w}_k, \beta_k\}$ is the set of model parameters. The likelihood function can be estimated by summing over the hidden states as

$$p(\boldsymbol{x}|\Theta) = \sum_{\boldsymbol{Z}} p(\boldsymbol{x}, \boldsymbol{Z}|\Theta) \tag{4}$$

In Bayesian inference, the hidden states and the model parameters are estimated by first assigning a prior distribution to the model parameters, $p(\Theta)$, and then calculating the posterior distribution using Bayes rule as

$$p(\boldsymbol{Z}, \Theta|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{Z}|\Theta)p(\Theta)}{\int p(\boldsymbol{x}|\Theta)p(\Theta)d\Theta}. \tag{5}$$

For ARHMM, the posterior distributions do not have a closed form because the integration in the denominator of Eq. 5 is not tractable, and therefore the posterior distribution is approximated using variational methods [4]. The method aims to approximate $p(\boldsymbol{Z}, \Theta|\boldsymbol{x})$ by $Q(\boldsymbol{Z}, \Theta|\boldsymbol{x})$ such that $Q(\boldsymbol{Z}, \Theta|\boldsymbol{x})$ is similar to $p(\boldsymbol{Z}, \Theta|\boldsymbol{x})$ and has a closed form. The variational method used in this study has its origin in *mean field theory* [18] where it was developed to approximate lowest-energy state in quantum mechanical calculations. Similar methods were used by MacKay [16] and Shihao Ji et al. [25] for estimating parameters of the standard HMM; this paper extends the methodology to ARHMM.

The method assumes that the approximating distribution, $Q(\boldsymbol{Z}, \Theta|\boldsymbol{x})$, factorizes as

$$Q(\boldsymbol{Z}, \Theta) = q(\boldsymbol{Z})q(\boldsymbol{A})q(\boldsymbol{\pi}) \prod_{k=1}^{K} q(\boldsymbol{w}_k)q(\beta_k). \tag{6}$$

In Eq. 6, to make the notation uncluttered, explicit dependence of approximating distribution on the data has been dropped i.e. $Q(\boldsymbol{Z}, \Theta|\boldsymbol{x}) \equiv Q(\boldsymbol{Z}, \Theta)$. To estimate $Q(\boldsymbol{Z}, \Theta)$, the Kullback-Leibler (KL) divergence [14] between $Q(\boldsymbol{Z}, \Theta)$ and $p(\boldsymbol{Z}, \Theta|\boldsymbol{x})$ (Eq. 7), which is a measure of distance between the two distributions, is minimized.

$$\mathrm{KL}(Q\|p) = -\int \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z}, \Theta) \ln \left[ \frac{p(\boldsymbol{Z}, \Theta|\boldsymbol{x})}{Q(\boldsymbol{Z}, \Theta)} \right] d\Theta. \tag{7}$$

Since KL-divergence is a convex function for each factor (taken independently) in the right hand side of Eq. 6, so an iterative algorithm is guaranteed to converge to the global

minimum. The optimal value of the approximating distributions is represented by

$$Q^\star(\boldsymbol{Z},\Theta) = q^\star(\boldsymbol{Z})q^\star(\boldsymbol{A})q^\star(\boldsymbol{\pi})\prod_{k=1}^{K} q^\star(\boldsymbol{w}_k)q^\star(\beta_k). \quad (8)$$

An additional benefit of variational methods is that they can be used for model selection, i.e. selecting the best model from a set of candidate models. For an ARHMM, it will correspond to selecting the number of monsoon states ($K$), sub-states ($S_K$), and lags ($L$). The model evidence can be decomposed as:

$$\ln p(\boldsymbol{x}) = \mathcal{L}(Q) + \mathrm{KL}(Q\|p) \quad (9)$$

where $\mathcal{L}(Q) = \int \sum_{\boldsymbol{Z}} Q(\boldsymbol{Z},\Theta)\ln[p(\boldsymbol{x},\boldsymbol{Z},\Theta)/Q(\boldsymbol{Z},\Theta)]d\Theta$ is known as *lower bound* or *negative free energy* and can be easily estimated for a given $Q(\boldsymbol{Z},\Theta)$. The best candidate model is the one that maximizes the lower bound [4]. A secondary advantage of calculating $\mathcal{L}(Q)$ is realized in monitoring the convergence of the iterative estimation procedure and checking the software implementation by noting that each iteration should always increase the value of $\mathcal{L}(Q)$.

## 4. MODEL DEVELOPMENT

Following are the steps involved in training an ARHMM:

(1) *Initialization step* - Initialize the ARHMM model by providing a guess value for the monsoon states $q(\boldsymbol{Z})$. Various methods - random initialization, initialization based on the definitions of existing methods, and initialization from the output of a *Gamma mixture model* fitted to data (treating data as i.i.d) - were tested. All the methods, except random initialization, yielded qualitatively similar results. The random initialization often resulted in slow convergence. Initialize iteration counter '*iter*' to zero.

(2) *Prior distribution* Use the following conjugate priors for the model parameters

$$p(\Theta) = p(\boldsymbol{A})p(\boldsymbol{\pi})\prod_{k=1}^{K} p(\boldsymbol{w}_k|\boldsymbol{\alpha}_k)p(\boldsymbol{\alpha}_k)p(\beta_k) \quad (10)$$

where $\boldsymbol{\alpha}_k = [\alpha_{k1},\ldots,\alpha_{kL}]^{\mathrm{T}}$ is a hyper-parameter and

$$\begin{aligned} p(\boldsymbol{A}) &= \prod_{k=1}^{K}\mathrm{Dir}(A_{k1},\ldots,A_{kK}|u^A_{k1},\ldots,u^A_{kK}),\\ p(\boldsymbol{\pi}) &= \mathrm{Dir}(\pi_1,\ldots,\pi_K|u^\pi_1,\ldots,u^\pi_K),\\ p(\boldsymbol{w}_k|\boldsymbol{\alpha}_k) &= \prod_{l=1}^{L}\mathcal{N}(w_{kl}|\mu^w_l,\alpha^{-1}_{kl}),\\ p(\boldsymbol{\alpha}_k) &= \prod_{l=1}^{L}\mathrm{Gamma}(\alpha_{kl}|a_0,b_0),\\ p(\beta_k) &= \mathrm{Gamma}(\beta_k|c_0,d_0). \end{aligned}$$

The functional form of the Dirichlet distribution is given in the Appendix. In this study, to make the prior distributions *noninformative* (broad) the hyperparameters were selected as:

$$\left.\begin{aligned} u^A_{kl} = u^\pi_k = a_0 = b_0 = c_0 = d_0 = 0.01\\ \mu^w_l = 0 \end{aligned}\right\} \quad \forall k,l$$

and define $\boldsymbol{\mu}_0 = [\mu^w_1,\ldots,\mu^w_L]^{\mathrm{T}}$.

(3) *Maximization step* - Increment '*iter*' to '*iter*+1' and estimate the posterior distribution of the model parameters $Q^\star(\Theta)$ by using variational Bayes method described in section 3.2. The expression for the posterior distribution obtained by minimizing KL divergence (Eq. 7) is given in the Appendix.

(4) *Expectation step* - Estimate the posterior distribution of the monsoon states $q^\star(Z)$ by using the standard *forward-backward* algorithm, also known as *Baum-Welch* algorithm [2].

(5) *Convergence check* - Estimate the lower bound, $\mathcal{L}(Q)$ (Eq. 9), using the posterior distribution of the model parameters and the monsoon states estimated in the foregoing steps. If '*iter*' is less than maximum number of iterations '$iter_{\max}$' (50 in this study) and the increase in $\mathcal{L}(Q)$ from the previous iteration is greater than a specified threshold ('*tol*' $= 10^{-3}$), continue with step (3)

(6) *Model selection* - Repeat all the previous steps for different number of monsoon states ($K = 2$ to 8), sub-states ($S_K = 1$ to 7), and lags ($L = 1$ to 7). The best model is the one for which the value of $\mathcal{L}(Q)$ at convergence is maximum. For most of the cases studied, $\mathcal{L}(Q)$ can distinctively select the optimal model. However, in case of ambiguity, preference was given to the more parsimonious model.

The steps outlined are for off-line training of an ARHMM using rainfall series of a single monsoon season. For training the ARHMM on rainfall data of multiple monsoon seasons, the variational Bayes algorithm requires a straightforward modification. The posterior distribution $q^\star(\boldsymbol{Z})$, at convergence, provides a probabilistic classification of the monsoon states in a rainfall time series.

The ARHMM training algorithm presented here can be used for real-time identification of the monsoon states, for which the model parameters should be updated on-line by treating their posterior distribution at the current time step as the prior distribution for the next time step. In addition, the on-line algorithm can provide predictive distribution for the next time step, $x_{N+1}$, based on the observed data, $\boldsymbol{x} = [x_1,\ldots,N]^{\mathrm{T}}$, as

$$\begin{aligned} p(x_{N+1}|\boldsymbol{x}) = &\sum_{\boldsymbol{z}_{N+1}} p(x_{N+1}|\boldsymbol{z}_{N+1})\\ &\sum_{\boldsymbol{z}_N} p(\boldsymbol{z}_{N+1}|\boldsymbol{z}_N)q^\star(\boldsymbol{z}_N). \end{aligned} \quad (11)$$

Longer lead prediction can also be made by using predictive distributions at the previous time steps. However, with increase in lead time, the uncertainty in the prediction will accumulate.

## 5. RESULTS AND DISCUSSION

The results of ARHMM based A&B state identification me-thod are first compared with the NCC model [21]. To identify the A&B states, the NCC model uses July and August rainfall over a region in central India, referred to as the *core monsoon region* (CMR; Fig. 4). For comparison, an ARHMM was trained in offline mode on the spatially averaged rainfall in the CMR for July and August months. The training period was from 1951 to 2004.

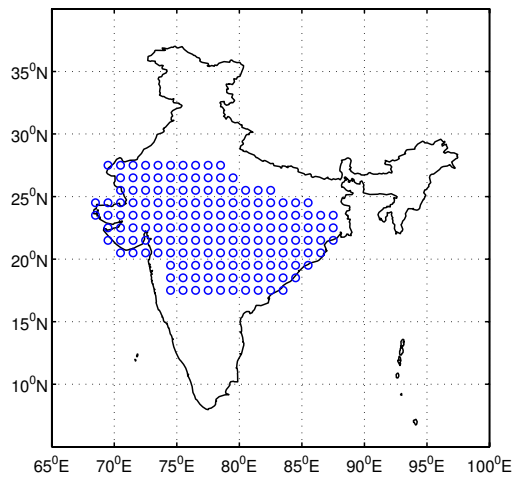



**Figure 4: The core monsoon region used in the NCC model to identify A&B states.**

Developing ARHMM requires selection of - (i) the number of monsoon states ($K$), (ii) the number of lags ($L$), and (iii) the number of sub-states ($S_K$). Herein, these parameters were selected using *variational lower bound*, i.e. the candidate model that maximized the lower bound was selected as the best model. The search for the best model was performed in the three dimensional space of $K$, $L$, and $S_K$. The results are given in Fig. 5. To facilitate interpretation, the lower bound is shown separately for each variable by setting the other two variables to their optimal values. The optimal values selected were $K = 4$, $L = 3$, and $S_K = 1$.

To check the derivation of variational ARHMM equations and its software implementation, the variational lower bound was monitored during the training phase. Figure 6 shows the evolution of the lower bound against the iteration steps. As desired, the lower bound increases with every iteration and asymptotically reaches a constant value.

Figure 7 presents the monsoon states identified by the AR-HMM for three years - an extreme dry year (2002), an extreme wet year (1994), and a normal year (1981). The A&B states identified by the NCC model are also shown. The ARHMM classifies a monsoon day into four states labeled - break, weak, normal, and active states, with certain probability, whereas the NCC model discretely classifies a day as an active, break or normal day.

For 2002, both the methods classify most of the days in July as break state. The NCC model classifies the period between $18^{\text{th}}$ and $20^{\text{th}}$ July as normal period, whereas the ARHMM classifies it as a weak period. Thus the higher resolution of the ARHMM (four states compared to three states in the NCC model) provides finer details that are lost in the NCC model. According to the ARHMM there is a high probability that the monsoon was active during $24^{\text{th}}$ and $25^{\text{th}}$ August. In contrast, the NCC model classifies the monsoon state during those days to be normal because, according to the model, the rainfall should exceed a specified threshold for *at least three days* for that period to be classified as an active period. This example highlights that specifying arbitrary thresholds may lead to misleading conclusions about the characteristics (duration and frequency)

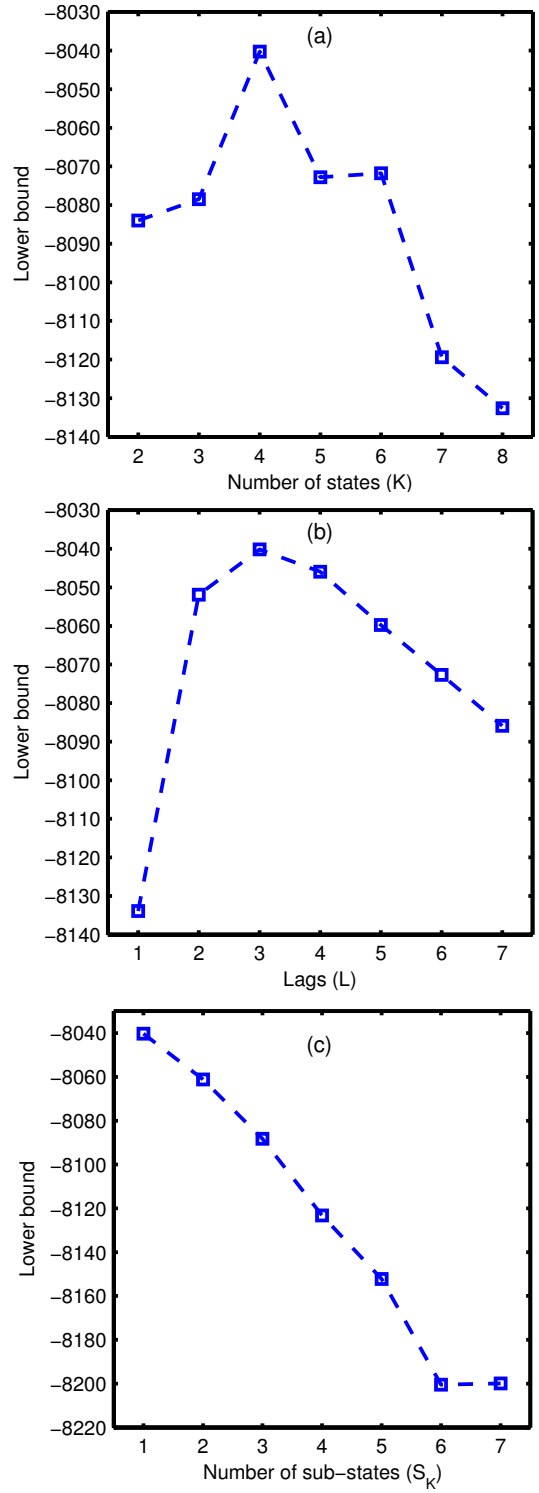



**Figure 5: Selection of (a) the number of monsoon states ($K$), (b) the number of lags ($L$), and (c) the number of sub-states ($S_K$) using variational lower bound. The parameters selected were $K = 4$, $L = 3$, and $S_K = 1$.**

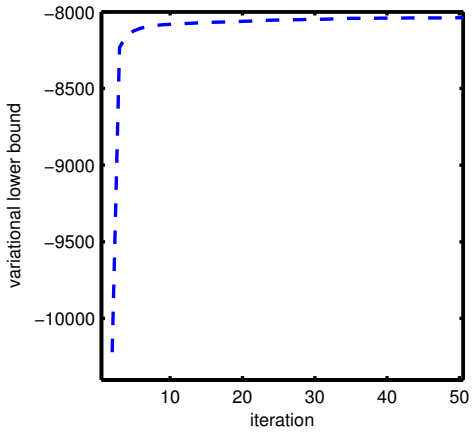



**Figure 6: Variational lower bond vs iteration number plotted to monitor the convergence of ARHMM.**

**Table 1: Transition probability of the monsoon states in the core monsoon region estimated by the offline ARHMM.**

| State | Break | Weak | Normal | Active |
|---|---|---|---|---|
| Break | 0.78 | 0.22 | 0.00 | 0.00 |
| Weak | 0.04 | 0.75 | 0.21 | 0.00 |
| Normal | 0.01 | 0.12 | 0.64 | 0.23 |
| Active | 0.00 | 0.00 | 0.72 | 0.28 |

of the A&B states. Similar situations exist in the other two years shown in Fig. 7(b and c). For instance, in 1981 the magnitude of rainfall during 9$^{th}$ and 10$^{th}$ August is considerably greater than rainfall between 7$^{th}$ and 10$^{th}$ July. The ARHMM (correctly) gives more probability for the former event to be in the active state than the latter. On contrary, the NCC model classifies 7$^{th}$ and 10$^{th}$ July as an active period but 9$^{th}$ and 10$^{th}$ August as a normal period because it was only a two day event.

Table 1 presents the transition matrix, $A$, of the monsoon states in the CMR. Among all the states, the active state has the smallest probability of self-transition (0.28) indicating that this state has relatively shorter duration. The self-transition probability of the break states is largest (0.78) suggesting that the these states tend to have a longer lifespan, which is consistent with the results reported by Rajeevan et al. [21]. It was observed that unlike active states, the break states often lasted for a week or longer. Further, Table 1 indicates that the weak state has a higher probability to go into a normal state than into a break state, while the normal state is more likely to move into an active spell than into a weak spell. The information in the transition matrix could be exploited for forecasting rainfall which will be discussed shortly.

Figure 8 shows the online identification of the monsoon states by an ARHMM for 2002. The prior distributions of the model parameters for the online method were set equal to the posterior distributions of the model parameters of an ARHMM trained in an offline mode on rainfall data from 1951 to 2001. For comparison, the NCC model results are reproduced in the top panel of the figure. The break spell identified by the online ARHMM overlaps with the break

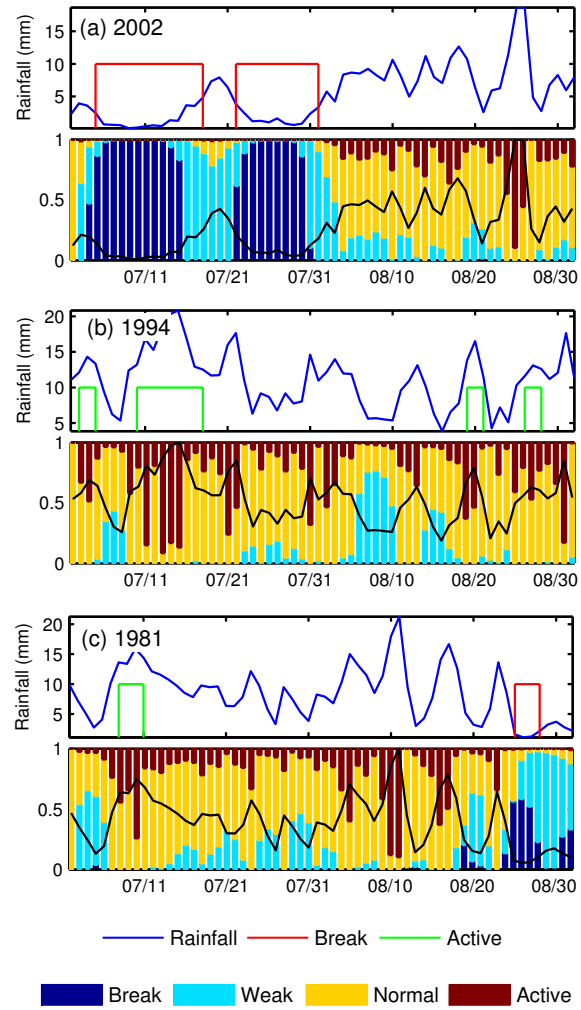



**Figure 7: Monsoon states identified by the NCC model (top panels) and an offline ARHMM (bottom panels). The legends used in the figure are shown in the top-most panel. The years 2002, 1994, and 1981 represents an extremely dry, an extremely wet, and a normal year.**

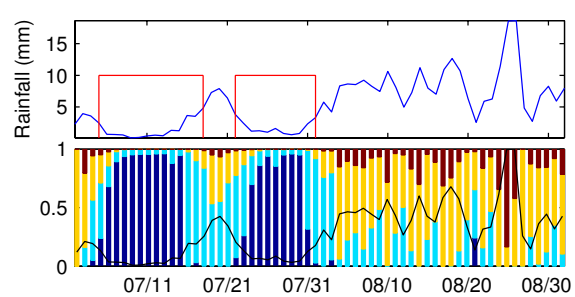



**Figure 8: Monsoon states identified by the NCC model (top panel) and an online ARHMM (bottom panel) for 2002. The legends used in the figure are same as those used in Fig. 7.**

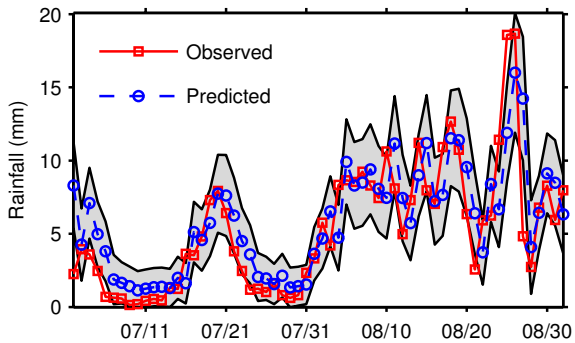



**Figure 9: One day ahead forecast of the rainfall in the core monsoon region for 2002 using an online ARHMM. The correlation between the observed and the forecasted values is 0.81.**

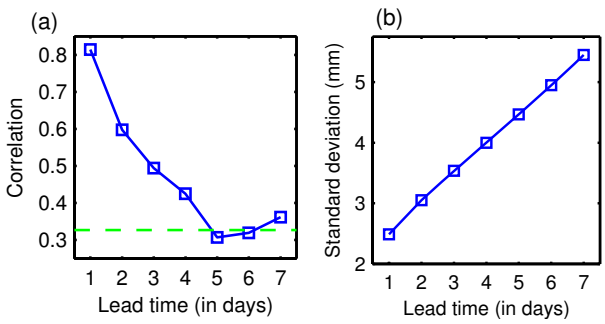



**Figure 10: (a) Correlation between the observed and the forecasted rainfall in the core monsoon region for 2002 at different lead times using ARHMM, and (b) mean of the standard deviation of the forecasted values.**

spell identified by the NCC model. Comparison of the online and the offline ARHMM (Fig. 7) indicates that the probabilities associated to a state are slightly tempered in the online model. For example, the probability of break spell during $23^{\text{rd}}$ to $26^{\text{th}}$ July by the offline model is close to unity, but the probability by the online model is around 0.95.

The online ARHMM can be used to forecast rainfall. Figure 9 presents one day ahead forecast for 2002 monsoon rainfall in the CMR. The shaded region in the figure represents one standard deviation in the forecasted values. The correlation coefficient between the observed and the forecasted values is 0.81. With increase in the forecast lead time, the correlation drops almost exponentially, while the standard deviation of the forecasted value rises linearly (Fig. 10). The correlation remains significant at 1% level up to four days lead time. It should be noted that the forecast is solely based on the temporal structure in the historical record of the rainfall data learned by an ARHMM. The forecast is likely to improve if potential predictors like sea surface temperature are included in the model [11, 27].

The ARHMM is a generative model, i.e. it can simulate observed rainfall time series. To test this feature of the ARHMM, 500 simulations were obtained from an offline ARHMM model trained on rainfall time series of the CMR for period 1951 to 2004. Figure 11(a) shows the mean of the

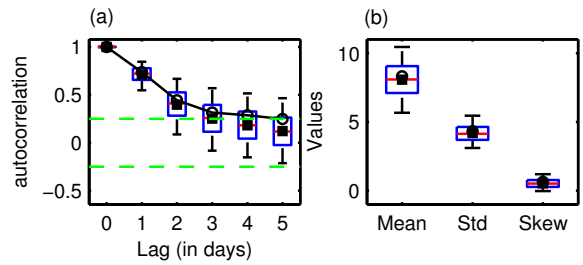



**Figure 11: Statistics of the observed and the simulated rainfall in the core monsoon region during 1951 to 2004. The horizontal line in the middle of the box represents median of the simulated value. The circle denotes the mean value of the statistics in the observed rainfall and the darkened square represents the mean value of the same in the simulated rainfall. The top and bottom of the box ranges from 25 to 75 percentile while the whiskers extend to 10 and 90 percentile. The dashed-lines in the first plot represent 95% confidence interval, and the terms 'std' and 'skew' in the second plot refer to standard deviation and skewness, respectively.**

autocorrelations in the observed data and in the simulated data. The autocorrelation in the simulated data closely resembles those in the observed data up to three days lag. For longer lags, the simulated rainfall has comparatively lower values of autocorrelation than in the observed data. However, the simulated time series could replicate the first three moments in the observed data Fig. 11(b).

## 6. CONCLUDING REMARKS

A hidden Markov model (HMM) for the identification of the A&B states in the Indian summer monsoon was proposed. To achieve the required flexibility in modeling monsoon rainfall, an auto-regressive HMM (ARHMM) with expanded states was used. The parameters of the model were estimated using variational Bayes method. The proposed model does not require specification of thresholds that makes existing models for identifying the A&B states subjective. Further, the proposed method exploits temporal dependence in the rainfall data which is vastly ignored in existing methods. Comparison of the proposed model with the NCC model [21] shows that both the models give similar results when the A&B states are conspicuous, but in case of ambiguity the proposed model has advantages due to its probabilistic setting. In addition, the proposed model can be used for - (i) identifying monsoon states in real time, (ii) forecasting rainfall, and (ii) generating synthetic rainfall data.

The model developed in the study provides a promising framework to develop models for forecasting monsoon intra-seasonal oscillations. However, several avenues should be explored to further refine this attempt. The proposed approach can be readily extended to incorporate predictors like sea surface temperature and geopotential height to improve forecast of the monsoon intra-seasonal oscillations at longer lead times. To bring out the advantages of the proposed method, its predictive and data generating abilities should be compared with conventional time-series techniques. Extended research work in this direction is underway.

## 7. REFERENCES


[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions*. Dover, 1965.

[2] L. E. Baum. An inequality and associated maximization technique in statistical estimation of probalistic functions of Markov processes. *Inequalities*, 3:1–8, 1972.

[3] E. Bellone, J. P. Hughes, and P. Guttorp. A hidden Markov model for downscaling synoptic atmospheric patterns to precipitation amounts. *Climate Research*, 15(1):1–12, 2000.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, USA, first edition, Aug 2006.

[5] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic Acids*. Cambridge University Press, Cambridge, U.K., 1998.

[6] Y. Ephraim, D. Malah, and B. H. Juang. On the application of hidden Markov models for enhancing noisy speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 533–536, 1988.

[7] S. Gadgil and P. V. Joseph. On breaks of the Indian monsoon. *Proceedings of the Indian Academy of Sciences, Earth and Planetary Sciences*, 112(4):529–558, 2003.

[8] B. N. Goswami, V. Venugopal, D. Sengupta, M. S. Madhusoodanan, and P. K. Xavier. Increasing trend of extreme rain events over India in a warming environment. *Science*, 314(5804):1442–1445, 2006.

[9] A. M. Greene, A. W. Robertson, and S. Kirshner. Analysis of Indian monsoon daily rainfall on subseasonal to multidecadal time-scales using a hidden Markov model. *Quarterly Journal of the Royal Meteorological Society*, 134(633):875–887, 2008.

[10] B. B. Jackson. Markov mixture models for drought lengths. *Water Resources Research*, 11(1):64–74, 1975.

[11] J. Jian, P. J. Webster, and C. D. Hoyos. Large-scale controls on Ganges and Brahmaputra river discharge on intraseasonal and seasonal time-scales. *Quarterly Journal of the Royal Meteorological Society*, 135(639):353–370, 2009.

[12] M. T. Johnson. Capacity and complexity of HMM duration modeling techniques. *IEEE Signal Processing Letters*, 12(5):407–410, 2005.

[13] V. Krishnamurthy and J. Shukla. Seasonal persistence and propagation of intraseasonal patterns over the Indian monsoon region. *Climate Dynamics*, 30(4):353–369, 2008.

[14] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[15] S. E. Levinson. Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):29–45, 1986.

[16] D. J. C. MacKay. Ensemble learning for hidden Markov models. *Unpublished paper*, `http://www.inference.phy.cam.ac.uk/mackay/abstracts/ensemblePaper.html`, 1997.

[17] S. K. Mandke, A. K. Sahai, M. A. Shinde, S. Joseph, and R. Chattopadhyay. Simulated changes in active\break spells during the Indian summer monsoon due to enhanced $CO_2$ concentrations: assessment from selected coupled atmosphere-ocean global climate models. *International Journal of Climatology*, 27(7):837–859, 2007.

[18] G. Parisi. *Statistical Field Theory*. Addison-Wesley, New York, 1988.

[19] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[20] M. Rajeevan, J. Bhate, J. D. Kale, and B. Lal. High resolution daily gridded rainfall data for the Indian region: analysis of break and active monsoon spells. *Current Science*, 91(3):296–306, 2006.

[21] M. Rajeevan, S. Gadgil, and J. Bhate. *Active and break spells of the Indian summer monsoon*. National Climate Center research report, India Meteorological Department, Poona, India, 2008.

[22] M. R. Ramesh Kumar and U. R. Prabhu Dessai. A new criterion for identifying breaks in monsoon conditions over the Indian subcontinent. *Geophysical Research Letters*, 31(18):L18201, 2004.

[23] M. Russell and A. Cook. Experimental evaluation of duration modelling techniques for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing,ICASSP '87*, volume 12, pages 2376–2379, 1987.

[24] P. Satyanarayana and V. V. Srinivas. Regional frequency analysis of precipitation using large-scale atmospheric variables. *Journal of Geophysical Research (Atmospheres)*, 113, 2008.

[25] Shihao Ji, B. Krishnapuram, and L. Carin. Variational Bayes for continuous hidden Markov models and its application to active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, 28(4):522–532, 2006.

[26] M. Thyer and G. Kuczera. A hidden Markov model for modelling long-term persistence in multi-site rainfall time series 1. model calibration using a Bayesian approach. *Journal of Hydrology*, 275(1–2):12–26, 2003.

[27] P. J. Webster and C. Hoyos. Prediction of monsoon rainfall and river discharge on 15-30-day time scales. *Bulletin of the American Meteorological Society*, 85(11):1745–1765, 2004.


## APPENDIX

## A. DIRICHLET DISTRIBUTION

For a $K$-dimensional random variable $\boldsymbol{x} = [x_1, \ldots, x_K]^{\mathrm{T}}$, such that $0 \leq x_k \leq 1$, and $\sum_{k=1}^{K} x_k = 1$, the Dirichlet distribution with parameters $\boldsymbol{u} = [u_1, \ldots, u_k]^{\mathrm{T}}$ was defined as

$$\mathrm{Dir}(\boldsymbol{x}|\boldsymbol{u}) = \frac{\Gamma(u_1 + \ldots + u_K)}{\Gamma(u_1) \cdots \Gamma(u_K)} \prod_{k=1}^{K} x_k^{u_k - 1} \tag{12}$$

where $\Gamma(\cdot)$ is the *gamma function*.

## B. POSTERIOR DISTRIBUTIONS

The posterior distribution of the ARHMM parameters ($\Theta$) and the monsoon states ($\boldsymbol{Z}$) was estimated as

$$Q^\star(\boldsymbol{Z},\Theta) = q^\star(\boldsymbol{Z})q^\star(\boldsymbol{A})q^\star(\boldsymbol{\pi})\prod_{k=1}^{K} q^\star(\boldsymbol{w}_k)q^\star(\boldsymbol{\alpha}_k)q^\star(\beta_k). \tag{13}$$

The component distributions were derived to be:

$$\begin{aligned}
q^\star(\boldsymbol{A}) &= \prod_{k=1}^{K} \mathrm{Dir}(A_{k1},\ldots,A_{kK}|U^A_{k1},\ldots,U^A_{kK}),\\
q^\star(\boldsymbol{\pi}) &= \mathrm{Dir}(\pi_1,\ldots,\pi_K|U^\pi_1,\ldots,U^\pi_K),\\
q^\star(\boldsymbol{w}_k) &= \mathcal{N}(\boldsymbol{w}_k|\boldsymbol{m}_k,\boldsymbol{P}_k^{-1}),\\
q^\star(\boldsymbol{\alpha}_k) &= \prod_{l=1}^{L} \mathrm{Gamma}(\alpha_{kl}|a_{kl},b_{kl}),\\
q^\star(\beta_k) &= \mathrm{Gamma}(\beta_k|c_k,d_k),\\
q^\star(\boldsymbol{Z}) &= \frac{1}{\Upsilon}\pi^\star_{s_1}\prod_{t=1}^{N-1} A^\star_{s_t,s_{t+1}}\prod_{t=1}^{N} f^\star(x_t|s_t)
\end{aligned}$$

where $s_t = k$ if $z_{tk} = 1$, $\Upsilon$ is a normalizing constant, and after defining

$$\begin{aligned}
\gamma_{tk} &= q^\star(\boldsymbol{Z})\delta(z_{tk}=1), \quad \text{and}\\
\xi_{kj} &= \sum_{t=1}^{N-1} q^\star(\boldsymbol{Z})\delta(z_{tk}=1, z_{t+1,j}=1),
\end{aligned}$$

the other terms were estimated as

$$\begin{aligned}
U^A_{kj} &= u^A_{kj} + \xi_{kj},\\
U^\pi_j &= u^\pi_j + \gamma_{1j},\\
\boldsymbol{P}_k &= \langle\beta_k\rangle\sum_{t=1}^{N}\gamma_{tk}\breve{\boldsymbol{x}}_t\breve{\boldsymbol{x}}_t^{\mathrm{T}} + \mathrm{diag}(\langle\boldsymbol{\alpha}_k\rangle),\\
\boldsymbol{m}_k &= \boldsymbol{P}_k^{-1}\left[\langle\beta_k\rangle\sum_{t=1}^{N}\gamma_{tk}\breve{\boldsymbol{x}}_t x_t + \mathrm{diag}(\langle\boldsymbol{\alpha}_k\rangle)\boldsymbol{\mu}_0\right],\\
a_{kl} &= a_0 + \frac{1}{2},\\
b_{kl} &= b_0 + \frac{1}{2}\left[\langle w_{kl}\rangle^2 + (\mu^w_l)^2 - 2\langle w_{kl}\rangle\mu^w_l\right],\\
c_k &= c_0 + \sum_{t=1}^{N}\gamma_{tk},\\
d_k &= d_0 + \sum_{t=1}^{N}[\gamma_{tk}(x_t^2 + \mathrm{trace}\left(\left\langle\boldsymbol{w}_k\boldsymbol{w}_k^{\mathrm{T}}\right\rangle\breve{\boldsymbol{x}}_t\breve{\boldsymbol{x}}_t^{\mathrm{T}}\right)\\
&\quad - 2x_t\langle\boldsymbol{w}_k\rangle^{\mathrm{T}}\breve{\boldsymbol{x}}_t)],\\
\pi^\star_{s_1} &= \exp\left[\psi(U^\pi_{s_1}) - \psi\left(\sum_{j=1}^{K}U^\pi_j\right)\right],\\
A^\star_{s_t,s_{t+1}} &= \exp\left[\psi(U^A_{s_t,s_{t+1}}) - \psi\left(\sum_{j=1}^{K}U^A_{s_t,j}\right)\right], \quad \text{and}\\
f^\star(x_t|s_t) &= \exp\{\frac{1}{2}[\langle\ln(\beta_{st})\rangle - \ln(2\pi) - \langle\beta_{s_t}\rangle\\
&\quad \left(x_t^2 + \mathrm{trace}\left(\left\langle\boldsymbol{w}_{s_t}\boldsymbol{w}_{s_t}^{\mathrm{T}}\right\rangle\breve{\boldsymbol{x}}_t\breve{\boldsymbol{x}}_t^{\mathrm{T}}\right) - 2x_t\langle\boldsymbol{w}_{s_t}\rangle^{\mathrm{T}}\breve{\boldsymbol{x}}_t\right)]\}.
\end{aligned}$$

The function $\psi(\cdot)$ is the *digamma function* [1], and the symbol $\langle\cdot\rangle$ denotes *expected value* which was estimated using the following properties. For a Gaussian distribution, $p(\boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{P}^{-1})$,

$$\begin{aligned}
\langle\boldsymbol{x}\rangle &= \boldsymbol{\mu},\\
\left\langle\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right\rangle &= \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}} + \boldsymbol{P}^{-1},\\
\left\langle\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}\right\rangle &= \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\mu} + \mathrm{trace}(\boldsymbol{P}^{-1});
\end{aligned}$$

and for a Gamma distribution, $p(x) \sim \mathrm{Gamma}(x|a,b)$,

$$\begin{aligned}
\langle x\rangle &= \frac{a}{b},\\
\left\langle x^2\right\rangle - \langle x\rangle^2 &= \frac{a}{b^2},\\
\langle\ln(x)\rangle &= \psi(a) - \ln(b).
\end{aligned}$$

# Reduction of Ground-Based Sensor Sites for Spatio-Temporal Analysis of Aerosols

Vladan Radosavljevic
Center for Information Science and Technology, Temple University
Philadelphia, PA 19122, USA
vladan@ist.temple.edu

Slobodan Vucetic
Center for Information Science and Technology, Temple University
Philadelphia, PA 19122, USA
vucetic@ist.temple.edu

Zoran Obradovic
Center for Information Science and Technology, Temple University
Philadelphia, PA 19122, USA
zoran@ist.temple.edu

## ABSTRACT
In many remote sensing applications it is important to use multiple sensors to be able to understand the major spatio-temporal distribution patterns of an observed phenomenon. A particular remote sensing application addressed in this study is estimation of an important property of atmosphere, called Aerosol Optical Depth (AOD). Remote sensing data for AOD estimation are collected from ground and satellite-based sensors. Satellite-based measurements can be used as attributes for estimation of AOD and in this way could lead to better understanding of spatio-temporal aerosol patterns on a global scale. Ground-based AOD estimation is more accurate and is traditionally used as ground-truth information in validation of satellite-based AOD estimations. In contrast to this traditional role of ground-based sensors, a data mining approach allows more active use of ground-based measurements as labels in supervised learning of a regression model for AOD estimation from satellite measurements. Considering the high operational costs of ground-based sensors, we are studying a budget-cut scenario that requires a reduction in a number of ground-based sensors. To minimize loss of information, the objective is to retain sensors that are the most useful as a source of labeled data. The proposed goodness criterion for the selection is how close the accuracy of a regression model built on data from a reduced sensor set is to the accuracy of a model built of the entire set of sensors. We developed an iterative method that removes sensors one by one from locations where AOD can be predicted most accurately using training data from the remaining sites. Extensive experiments on two years of globally distributed AERONET ground-based sensor data provide strong evidence that sensors selected using the proposed algorithm are more informative than the competing approaches that select sensors at random or that select sensors based on spatial diversity.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7...$5.00.

## Categories and Subject Descriptors
J.3 [**Computer Applications**]: Physical Sciences and Engineering – *Earth and atmospheric sciences*.

## General Terms
Algorithms, Measurement, Experimentation.

## Keywords
Remote Sensing, Data mining, Regression, Active Learning

## 1. INTRODUCTION
Aerosols, minute particles suspended in the atmosphere originating from natural and man-made sources, have become one of the main topics in climate research studies [8]. They have significant effect on health [14], vegetation, precipitation [17] and global climate [4]. Aerosols were identified as a central component missing from general circulation models (GCMs) that simulate climate changes [10]. After accounting for the aerosol effects, model-simulated climate changes have become more realistic [11] and an agreement between GCMs and real observations has been significantly improved.

The main optical property of aerosols is Aerosol Optical Depth (AOD) [13]. AOD is a measure of the visual or optical thickness of an aerosol layer. The process of predicting AOD using ground [5] or satellite [9] based observations is known as *AOD retrieval*. Ground based observations are mostly obtained by AErosol RObotic NETwork (AERONET) [5] which is a global remote

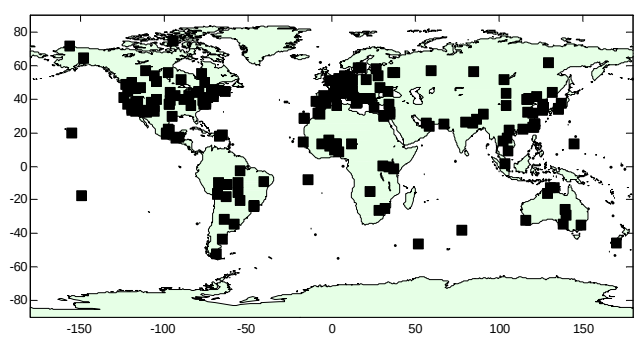



**Figure 1. Global distribution of AERONET sites.**

sensing network of radiometers that measure AOD several times per hour from specific geographic locations. AERONET instruments provide accurate estimation of local aerosol abundance, but they have low spatial coverage which limits their applicability in understanding global aerosol properties. On the other hand, satellite observations provide global coverage on a daily basis, but are less accurate because the signal that a satellite instrument receives is a mixture of reflected radiation by both the Earth's surface and the aerosol layer [7]. Accuracy of satellite-based AOD retrieval is one of the major limiting factors influencing simulation-based climate change studies [13].

The operational AOD retrieval algorithms are typically manually tuned by domain scientists [16]. While this guarantees that the retrievals are based on sound physical principles, it also creates problems when there is an opportunity to use ground truth data to improve the algorithm. In contrast to domain-driven methods for AOD retrieval that use a network of sensors installed on ground for validation purposes only, a data-driven approach is using them directly to train an algorithm for AOD retrieval from satellite observations. This approach is possible when a data set is available that consists of satellite observations and collocated ground-truth measurements from AERONET radiometers. Given such data, a regression model can be constructed that predicts the ground-truth labels from the satellite observations. In our previous studies to retrieve AOD from satellite observations, a predictor was trained on satellite observations spatially and temporally collocated with AERONET retrievals [18]. It has been shown that such a statistical approach could improve the accuracy of retrievals significantly as compared to the operational domain-based methods. Clearly, this improvement comes from the utilization of highly accurate ground-based measurements directly in the prediction model.

However, ground based stations are often located without a rigorous statistical design. Decisions are typically based on practical circumstances (e.g. overrepresentation in urban regions and industrialized nations) and according to domain experts' assumptions about the importance of specific sites. Furthermore, the total number of sensor sites depends on financial constraints. Costs related to equipment, location, and the availability of trained staff often dictate the number of sites and their global distribution. As shown in Figure 1, AERONET sites are not uniformly distributed over the globe. The highest density is within the U.S. and Europe. On the other hand, continental Asia, Africa, and Australia are poorly covered. Given these circumstances, the aims of our study are to evaluate performance of the current design of AERONET sensor network and to apply data mining techniques to assist in future modifications of the sensor network.

In a tribute to the ongoing economic crisis, a specific scenario considered in this paper assumes that there is a pending budget cut for maintenance of the existing AERONET sites. The objective is to shut down a fraction of the AERONET sites while making sure that the utility of the remaining sites is as high as possible. In this paper, we make a simplifying assumption that operational costs for each AERONET site around the globe are equal.

Common to most selection techniques originating from the spatial statistics is a tendency to overlook the time dimension of data collected by the sensor network. In this paper, for the problem of selecting a subset of data collection sites, we consider series of observations and propose to optimize AERONET sensor selection based on the concept of retrieval accuracy. The intuition behind our proposal is straightforward. Each AERONET site provides a time series which can be used in training a regression model to retrieve future AOD. Sites that can be removed are those whose observations are best predicted by the model built on data from the remaining sites. The performance of the proposed approach is compared with the random site selection and with the classical selection principle of selecting spatially dispersed sites.

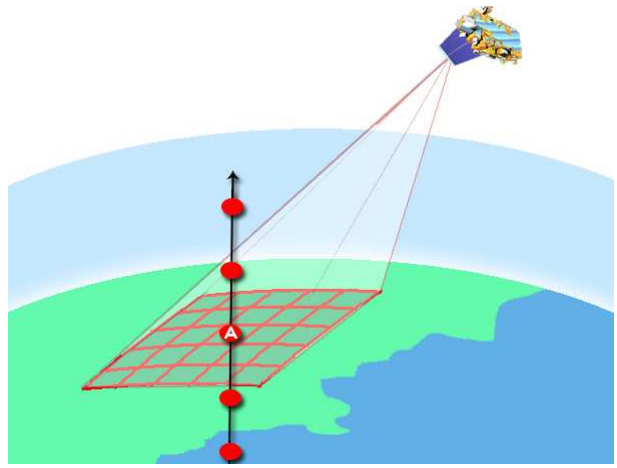



**Figure 2. Spatio-temporal collocation of MODIS and AERONET data. A is an AERONET site with AOD retrieved within a short time before and after the satellite overpass (circle dots). The square regions are MODIS observations in a proximity of site A at the satellite overpass time.**

# 2. METHODOLOGY

## 2.1 Data Fusion

Given a data set that consists of satellite observations and AERONET AOD measurements, a regression model can be trained to use satellite observations as attributes and predict the labels which are AERONET AODs. For that reason, satellite observations need to be collocated and merged with AERONET measurements.

In this study we consider data from MODerate resolution Imaging Spectrometer (MODIS), an instrument aboard NASA's Terra and Aqua satellites. Instruments mounted on Terra observe the Earth during morning whereas those mounted on Aqua observe the Earth during afternoon. In this study, we use data only from Terra satellite.

MODIS has high spatial resolution (pixel is as small as $250x250m^2$) and achieves global coverage daily. On the other hand, AERONET sites, situated at fixed geographical locations, acquire data at intervals of 15 min on average. This gives rise to the need for both spatial and temporal data fusion (Figure 2). The fusion method involves aggregating MODIS pixels into blocks of size $50x50\ km^2$ and spatially collocating them with an AERONET site. The MODIS observations are said to be temporally collocated with the corresponding AERONET AOD retrievals if there is a valid AERONET AOD retrieval within 30 minutes of the satellite overpass. The data collocated in this way can be obtained from the official MODIS website of NASA [6].

## 2.2 Regression Model

Let us assume we have access to data from a set of $N$ AERONET sites $S = \{S_i, i = 1, 2, \ldots N\}$. At site $S_i$ there is a sequence $\{(X_{it}, y_{it})\}$ of multivariate radiance observations $X_{it}$ collected from a satellite instrument spatio-temporally collocated with the corresponding ground-based AERONET AOD values $y_{it}$. To be able to accurately retrieve AOD from MODIS measurements we are using all labeled data from sensor set $S$ to train a regression model $f$ aimed to estimate target AOD values. Typically, the following data-generating model is assumed

$$Y = f(X) + \varepsilon, \varepsilon \sim \mathrm{N}(0, \sigma^2), \quad (1)$$

where $\varepsilon$ is Gaussian additive noise with constant variance $\sigma^2$.

Neural networks are often a regression model of choice in data-driven retrieval of atmospheric properties [1, 12]. In our previous work, neural networks have been trained to predict AERONET AOD over continental US [3] and whole globe [15] using attributes derived from satellite data. Comparing to the domain-based AOD retrievals, neural network AOD predictions were significantly more accurate.

The assumption of constant variance is a basic requirement in constructing a model. In many cases there is no reason to suspect that the error variance is not constant. However, our inspection of residual plot $f(X) - Y$ as a function of $f(X)$ *provides evidence that this assumption is violated at a certain AERONET sites*. At Figure 3 we notice that in our application variance $\sigma^2$ is not constant, but is proportional to $f(X)$.

Variance stabilizing transformations of target variable are often useful in these cases [2]. The strength of transformation depends on its curvature. Square root and logarithmic transformations are popular in practice. In square root transformation, a regression model that predicts $Z = \sqrt{Y}$ is trained and the prediction is provided as $\hat{Y} = \hat{Z}^2$, while in logarithmic transformation Z = log(Y) and $\hat{Y} = \exp(\hat{Z})$. Square transformation is considered as a relatively mild [2] comparing to the logarithmic and is often applied when variance of residuals increases linearly with predicted variable. In the experimental section we compared both of them with the standard approach that does not transform the target variable.

## 2.3 Selection of Informative AERONET Sites

Let us assume that a mission objective is to close down a fraction (33% or 66% in our experiments) of AERONET sites as to reduce ground-based data collection costs. Given such a budget cut situation, question of our interest is how to select $M$ ($<N$) of the currently available $N$ AERONET sites such that this subset captures as much information as possible compared to the entire set $S$. The goodness criterion for a selection is accuracy of a regression model built on labeled data from the retained sites.

Intuitively, it appears that the selection of sites that are spatially dispersed would be a better choice than a random elimination. Such a spatial selection might be aided by domain experts — they would prefer to keep representative sites around the globe that cover a variety of meteorological and environmental conditions. However, regardless of the experts' effort, spatial representatives selected this way may not be optimal with respect to the quality of the resulting regression model $f$.

**Table 1. List of attributes collected from the collocated satellite observations**

| Attribute index | Description |
|---|---|
| 1-4 | Mean radiation in 50 x 50 km$^2$ blocks at seven wavelengths |
| 5-9 | Std. deviation of radiation in 50 x 50 km$^2$ blocks at seven wavelengths |
| 10-13 | Ancillary attributes (view angles, elevation) |

The sites selected by a domain expert are likely to be spatially diverse. To approximate the decision-making process of domain experts, for benchmarking purposes we use the spatial selection strategy based on spatial distance among sites. In the first step two sites that are closest to each other are determined. One of them whose removal better preserves global coverage is excluded from the set $S$. To decide which one is going to be removed, we are consulting the nearest neighbors of those two sites. The site which has the closer second nearest neighbor is removed. This procedure is iteratively repeated until the desired number of $M$ sites is reached.

Our proposed strategy for selection of $M$ sites out of $N$ is accuracy-based. At the first step, the regression model $f$ is trained on the data from the entire set of AERONET sites. At successive steps, every location is taken out and a model is built on data from the remaining sites. By $\hat{Y}$ we denote AOD retrieval obtained by a model trained on whole dataset and by $\hat{Y}^{(i)}$ AOD retrieval obtained by a model trained on $S \backslash S_i$ sites that exclude examples from site $S_i$. The intuition is that if AODs from site $S_i$ can be estimated with a model which has not seen that site, then site $S_i$ can be considered as redundant and therefore can be removed. To quantitatively define redundancy, we measure the difference in AOD retrieval accuracy between the model trained on the whole dataset and model trained on a dataset without examples from site $S_i$. The difference in retrieval accuracy is measured at data from site $S_i$ as a sum of squared differences in retrieved AODs computed over all points from site $S_i$

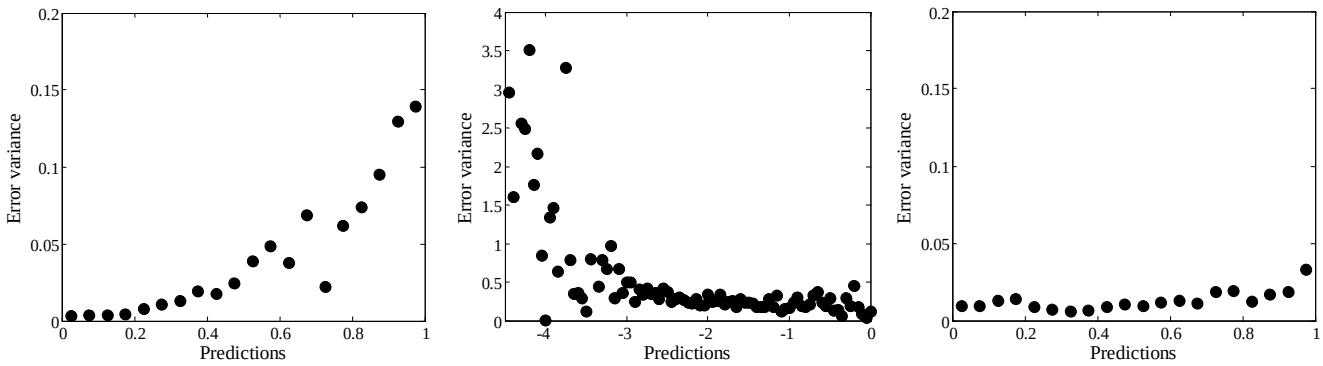
$$SSE_i = \sum_t (\hat{y}_i - \hat{y}_t^{(i)})^2 \quad (2)$$

A site that is removed is the one with the smallest *SSE* as its AOD is the easiest to estimate given data from the remaining sites. Once a site is removed the proposed procedure is repeated. It continues by comparing the reduced models to the model built on the entire data, where data from the most recently excluded site are used for calculating *SSE* based loss.

# 3. EXPERIMENTAL RESULTS

## 3.1 Dataset

There are several levels of AERONET AOD measurements [5]. To avoid potential problems with outliers in ground truth data, AERONET Level 2.0 observations were considered since they were cloud screened and manually verified.




**a)** **b)** **c)**

**Figure 3. Variance stabilizing effect of square root transformation. Error variance as a function of a) predictions without transformation b) predictions with log transformation c) predictions with square root transformation**

**Table 2. $R^2$ statistics on 2006 data for neural network models without (*NN*) and with log (*NNLG*) or square root (*NNSQ*) transformed output each built on ten different sets of 30 randomly selected sites using 2005 data.**

| **Model** | $R^2$ | | | | |
|---|---|---|---|---|---|
| | Mean | Std | Median | Min | Max |
| *NN* | 0.659 | 0.086 | 0.671 | 0.459 | 0.742 |
| *NNLG* | 0.664 | 0.091 | 0.703 | 0.444 | 0.721 |
| *NNSQ* | 0.746 | 0.042 | 0.754 | 0.644 | 0.789 |

For our study we collected MODIS Terra observations collocated with AERONET Level 2.0 points. We extracted satellite-based attributes that are used as inputs to knowledge based retrieval algorithms. The radiances at four wavelengths were taken from the MODIS range 440nm – 2100nm, as these are sufficient to describe aerosol properties [16]. An average and standard deviation of radiances of pixels in 50x50 $km^2$ blocks were then estimated. Attributes are listed in Table 1. Along with radiances we also extracted ancillary attributes. Information about geometry is characterized by solar and sensor angles. As surface elevation affects estimated AOD, it was also included in the set of attributes and has been extracted from AERONET data.

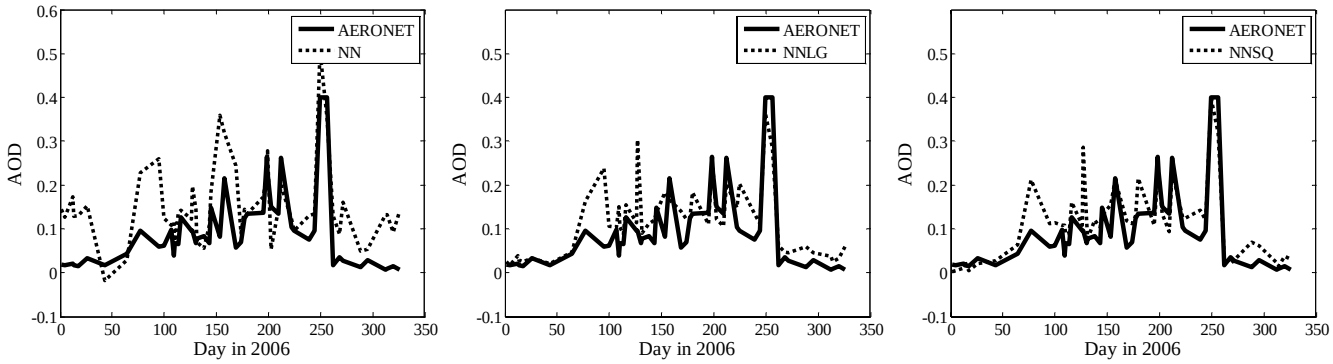
By convention, AOD is reported at the 550nm wavelength. Since AERONET sites do not provide AOD value at that particular wavelength, we performed a standard linear interpolation in the log scale of AERONET AOD measurements at 440nm and 670nm to estimate AOD at 550nm [16].

Data we collected are distributed over entire globe at 217 AERONET sites (Figure 1) during years 2005 and 2006. To assess efficiency of the proposed methods, we performed training on 2005 data and used 2006 data for testing. However, during that time period measurements from AERONET sites were not uniformly distributed, neither temporally or spatially. There were many more points from June to August than from January to May. Also, at some cloudy locations it was not possible to measure AOD and those locations contained a small number of data points. To maintain uniformity of the training dataset, in each training session we randomly selected 30 sites in year 2005 as the initial set *S*. Only 70 randomly chosen observations from each of those AERONET site were retained and remaining ones were removed. Finally, the training set consisted of 2,100 data points distributed over 30 AERONET sites each containing 70 collocated observations. As the test set, we randomly sampled 50 points from each site in 2006. Sites with less than 50 valid observations were excluded. The constructed test set contained 3,500 data points distributed over 70 AERONET sites each having 50 collocated observations. It is worth mentioning that among 70 test sites, 30 were the same as in the training set, while 40 sites were not seen during training. To evaluate the proposed approach, we report $R^2$ accuracy on the test set.

## 3.2 Determining an Appropriate AOD Transformation

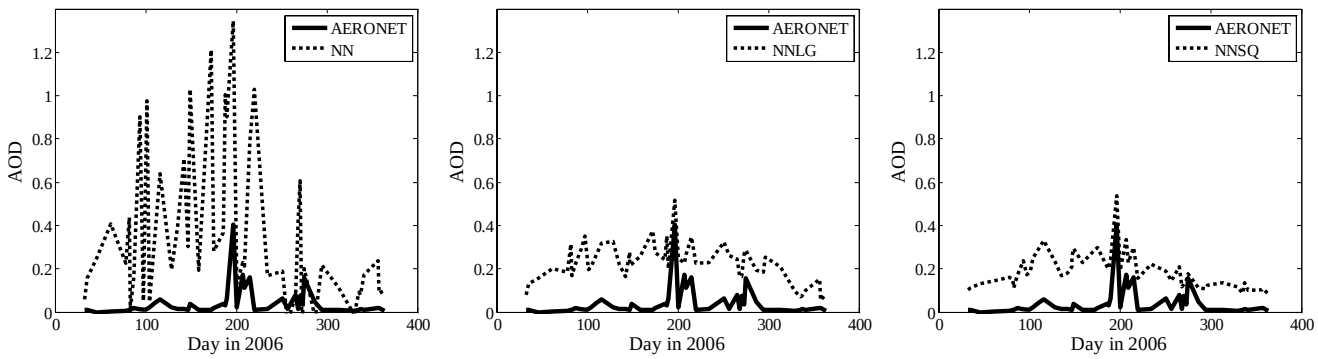
To validate the assumption that error variance is not constant and that empirically selected square root transformation is the most appropriate one, we performed the following experiment. Thirty sites in 2005 were chosen randomly. Three regression models, one with data preprocessed by the square root transformation (*NNSQ*), one with data preprocessed by the log transformation (*NNLG*) and the other without the transformation (*NN*), were trained on the selected dataset and compared on the test set. As a regression model we used a neural network with ten hidden neurons trained to optimize standard Mean Square Error (*MSE*) function.

This procedure was repeated ten times for different sets of 30 randomly selected sites. We report $R^2$ accuracy achieved on the fixed test set covering the 70 sites during 2006. To estimate sensitivity of constructed models to distribution of the initial 30 sites, we report mean, standard deviation, median and minimum and maximum of $R^2$ in those ten iterations. The results are presented in Table 2. These results provide strong evidence that the neural networks trained to predict AOD squared root (*NNSQ*) are more accurate than those trained to predict raw AOD (*NN*) or log-AOD (*NNLG*). Additionally, the presented results reveal that retrieval accuracy is sensitive to the choice of the initial set of *S* sites. Although each time the selected 30 sites were globally distributed covering all parts of the world, in some cases accuracy dropped significantly. A possible explanation could be that some of those sites have noisy data that negatively influence model performance.




**a)** **b)** **c)**

**Figure 4. AERONET AOD at site 'BSRN_BAO_Boulder', along with AOD retrievals by a) *NN*, b) *NNLG* and c) *NNSQ* models built on data from 30 sites**




**a)** **b)** **c)**

**Figure 5. AERONET AOD at site 'Izana', along with AOD retrievals by a) *NN*, b) *NNLG* and c) *NNSQ* models built on data from 30 sites**

To better illustrate the effect of the square root transformation, at Figure 3 we show variance of prediction errors as a function of predictions. As can be seen, if the transformation is not used, the error variance is large when large AODs are predicted. On the other hand, when the strong log-transformation is used, the error variance is large when small AODs are predicted. Finally, when square root transformation is used, error variance is practically constant and does not depend on the value of predicted AOD. Thus, minimizing MSE assuming constant variance (as in (1)) is justified for the square-root transformed AOD.

To get better insight how transformations influence prediction accuracy, we analyzed a series of AOD retrievals at AERONET site 'BSRN_BAO_Boulder' (40ºN, -105ºW). Sensor platform is on the rooftop of the building which is located on the high plains about 15 miles east of Boulder, CO, USA. Surrounding farmers' fields make satellite AOD retrieval easier in some yearly seasons. Satellite retrievals are more accurate over green regions which are often considered as dark [16] and therefore do not have an influence on observed radiation. AOD data from this AERONET site along with AOD retrievals of *NN*, *NNLG* and *NNSQ* are presented in Figure 4a, 4b and 4c respectively. By visual inspection of those plots we can see that if AERONET instrument from this site measures small AOD then model *NN* retrieves large AOD whereas both *NNLG* and *NNSQ* manage to retrieve small AOD. Although it looks that *NNLG* and *NNSQ* achieve similar accuracy comparing to AERONET AOD, by inspecting $R^2$ which is 0.72 for *NNSQ*, 0.54 for *NNLG* and -0.1 for *NN* we conclude that *NNSQ* is more accurate than *NNLG* and *NN*.

To explore how the proposed square root transformation influences prediction accuracy at some extreme situations we analyzed AOD predictions of the least accurate *NN*, *NNLG* and *NNSQ* neural networks. We observed that the largest retrieval errors were made on the site 'Izana' (28.3ºN, -16.5ºW) which is located on the island of Tenerife, Spain, at elevation of 2360m above sea level. The sensor platform is placed on the top of a mountain plateau. The sky is usually free of clouds and as a result is extremely clean and suitable for radiation measurements and calibrations. AOD data from this AERONET site along with AOD retrievals of *NN*, *NNLG* and *NNSQ* are presented in Figure 5a, 5b and 5c respectively. We can see that AERONET instrument from this site most of the time measures small AOD while all three models *NN*, *NNLG* and *NNSQ* predict large AOD. However, models *NNLG* and *NNSQ* trained on transformed data manage to make smaller predictions than *NN*.

Based on these results, we used *NNSQ* predictor in the following experiments.

## 3.3 Selection of Informative Sites

We are considering a scenario when current operational AERONET sites have to be reduced by 33% or 66%. In all experiments, we started from a set of 30 AERONET sites and applied the proposed method and the two alternatives (spatial and random selection) to identify a subset of 20 or 10 AERONET sites to be retained. The *NNSQ* models were trained on labeled data from 2005. To test the goodness of the identified subset we tested the *NNSQ* models on 70 sites from 2006 (as described in Section 3.1).

The $R^2$ results averaged over 10 repetitions are presented in Figure 6. We noticed that in some cases $R^2$ drops significantly when spatial and random selection strategies are used. Therefore, we also report median values of $R^2$ after 10 repetitions (Figure 7). In our experiments, the proposed accuracy-based selection achieved consistently better results than the alternatives. Also, accuracy of the proposed site reduction method did not change much even after removing 20 of the 30 AERONET sites. Interestingly, on average, the spatial selection strategy performed slightly worse than the random selection strategy.

Let us now consider the effect of the proposed sites reduction method on predictions at the site 'BSRN_BAO_Boulder' analyzed previously (Figure 4). Time series of AOD retrievals at this site for a single placement of 30 training sites are presented in Figure 8. *NNSQ* model trained on a reduced dataset was able to retrieve ground-truth AOD slightly less accurately than the model trained on data from all 30 sites. In terms of $R^2$ accuracy, *NNSQ* trained on a reduced dataset achieved $R^2$ = 0.64 while *NNSQ* trained on non-reduced dataset achieved $R^2$ = 0.72 at the site 'BSRN_BAO_Boulder'. The conclusion is that accuracy-based reduction retains most of the accuracy of the model built on non-reduced dataset.

In Figure 9 we illustrate site reduction for one initial placement of 30 AERONET sites. Spatial-based selection of AERONET sites nicely covers whole globe but it is not necessarily optimal for data-driven AOD retrieval problem as we already noticed (Figure 6). On the other hand, some regions of the world were underrepresented when an accuracy-based principle was applied (Figure 9e). The accuracy was retained to a certain extent although no site from East US or from middle Asia was selected.

## 4. DISCUSSION

In this work we presented a method for the reduction of a number of AERONET sites such that the remaining sites are as informative as possible. The goodness criterion for a site selection is the accuracy of a regression model built on the labeled data from the selected sites. We analyzed three different approaches for site selection. A common-sense approach used as a benchmark was a random selection of sites. An approach based on spatial distance among the sites was also considered. Sites were selected such that their spatial coverage was maximized. As an alternative that takes into account the actual measurements from ground-

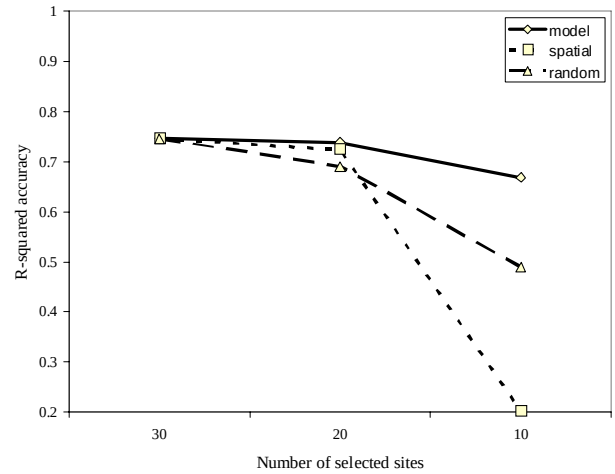



**Figure 6. Mean $R^2$ values in ten iterations for different initial sets of 30 AERONET sites.**

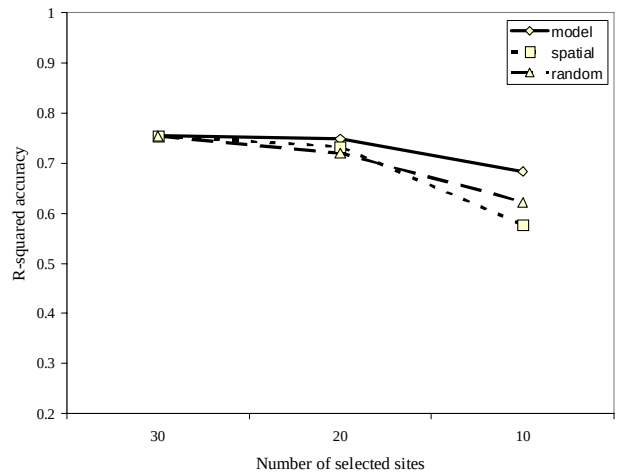



**Figure 7. Median $R^2$ values in ten iterations for different initial sets of 30 AERONET sites.**

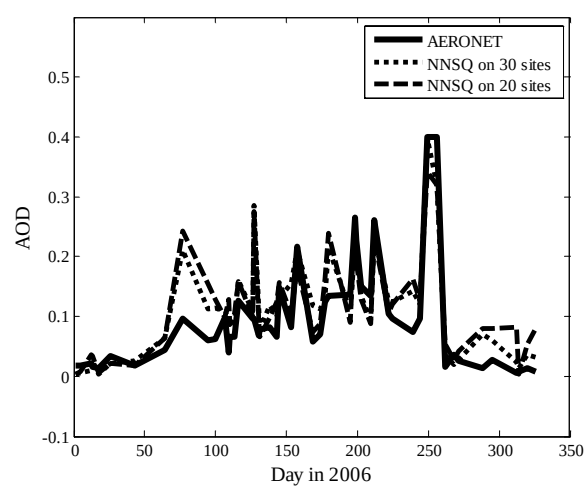



**Figure 8. AOD retrievals at site 'BSRN_BAO_Boulder' by *NNSQ* models trained on entire and reduced dataset.**

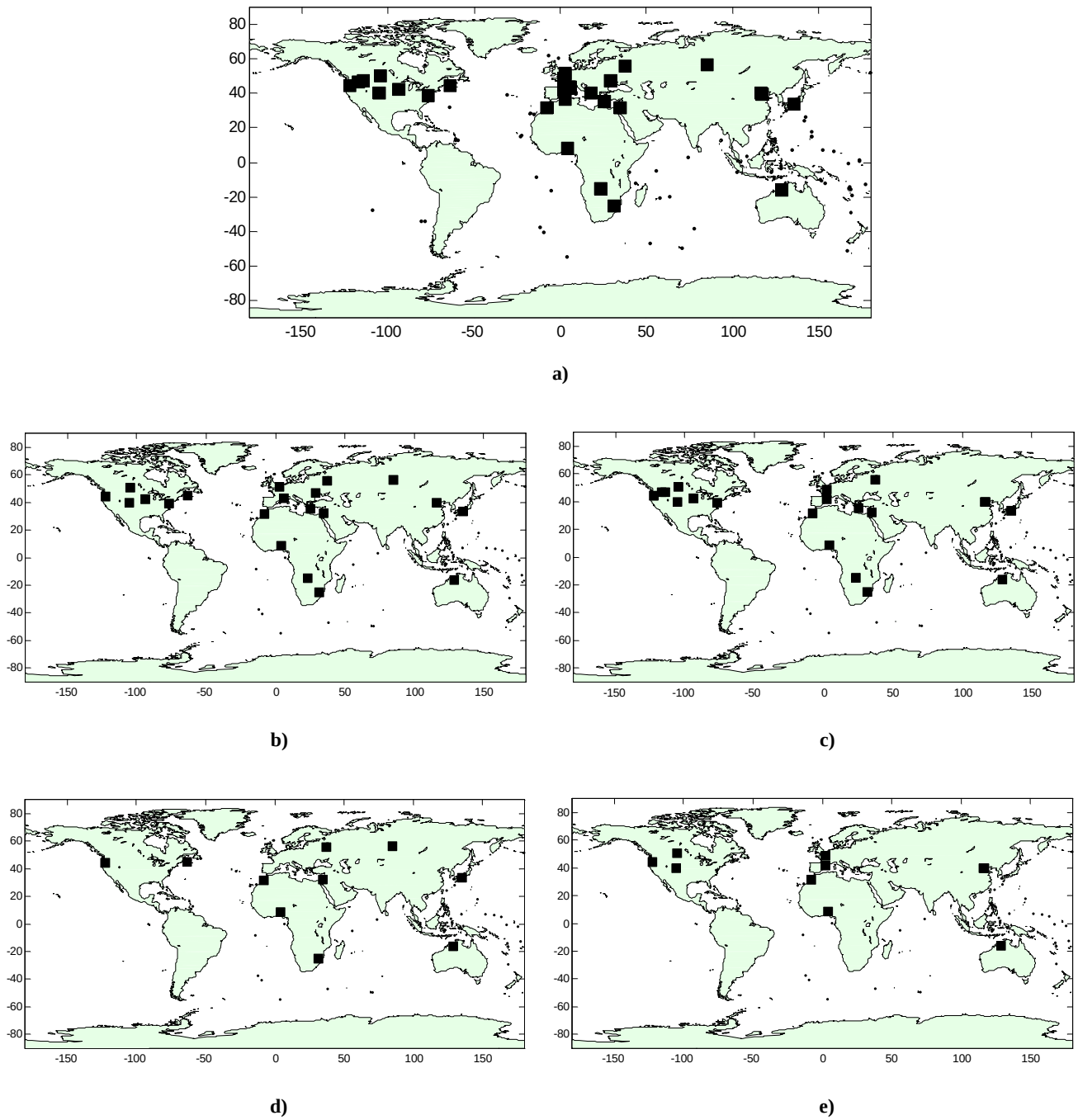



**Figure 9. a) Initial set of 30 AERONET sites b) spatial-based reduction to 20 sites c) accuracy-based reduction to 20 sites d) spatial-based reduction to 10 sites e) accuracy-based reduction to 10 sites.**

based AERONET sensors and satellite-based MODIS instruments, we proposed an accuracy-based selection approach. For this, a regression model was first trained on the labeled data from an entire set of sensors. After that, at successive steps, every location is excluded to check if AOD from that location can be predicted accurately by the model trained on labeled data from the remaining sites. The intuition was that if AODs from that site can be retrieved fairly well with a model which has not seen that site, then the ground-based measurements at that site can be considered as redundant.

According to presented results we conclude that the proposed accuracy-based sites reduction method is superior to spatially-based and random selection alternatives. In this report we did not address the question of determining the optimal number of sites to reduce the entire set in order to maintain a desired accuracy. This problem is addressed in our work in progress.

## ACKNOWLEDGEMENT

This work is funded in part by NSF grant IIS-0612149.

## 5. REFERENCES

[1] Aires, F., Prigent, C., Rossow, W. B., and Rothstein, M., 2001. A New Neural Network Approach Including First Guess for Retrieval of Atmospheric Watervapor, Cloud Liquid Water Path, Surface Temperature, and Emissivities Overland from Satellite Microwave Observations, J. Geophys. Res., 106, 14,887–14,908

[2] Box, George E. P.; Cox, D. R., 1964. An Analysis of Transformations, Journal of the Royal Statistical Society, Series B 26: 211–246.

[3] Han, B., Vucetic, S., Braverman, A., Obradovic, Z., 2006. A Statistical Complement to Deterministic Algorithms for the Retrieval of Aerosol Optical Thickness from Radiance Data, Engineering Applications of Artificial Intelligence, Vol. 19, No. 7, pp. 787-795.

[4] Hansen, J., Sato, M., and Ruedy, R., 1997. Radiative Forcing and Climate Response, J. Geophys. Res. 102, 6831-6864.

[5] Holben, B. N., Eck, T. F., Slutsker, I., Tanre, T., Buis, J. P., Setzer, A., Vermote, E., Reagan, J. A., Kaufman, Y. J., Nakajima, T., Lavenu, F., Jankowiak, I., and Smirnov, A., 1998. AERONET: A Federated Instrument Network and Data Archive for Aerosol Characterization, Remote Sens. Environ., 37, 2403 – 2412.

[6] http://modis.gsfc.nasa.gov/, Official MODIS website

[7] Kaufman, Y. J., Tanré, D., et al., 1997. Operational Remote Sensing of Tropospheric Aerosol Over Land From EOS Moderate Resolution Imaging Spectroradiometer", Journal of Geophysical Research - Atmospheres 102(D14): 17051-17067.

[8] Kaufman, Y. J., Tanre, D., and Boucher, O., 2002. A Satellite View of Aerosols in the Climate System, Nature, 419, (September 2002), 215-223.

[9] King, M. D., Kaufman, Y. J., Tanré, D., and Nakajima, T., 1999. Remote Sensing of Tropospheric Aerosols from Space: Past, Present, and Future, Bull. Amer. Meteor. Soc., 80, 2229-2259.

[10] Li, Z., 2004. Aerosol and Climate: A Perspective from East Asia, Observation, Theory and Modeling of Atmospheric Variability, (ed. Zhu), World Scientific Pub. Co., 501-525.

[11] Mitchell, J. F. B., Johns, T.C., Gregory, J.M., and Tett, S. F. B., 1995. Climate Response to Increasing Levels of Greenhouse Gases and Sulphate Aerosols, Nature, 376, 501-504.

[12] Muller, M. D., Kaifel, A. K., Weber, M., Tellmann, S., Burrows, J. P., and Loyola, D., 2003. Ozone profileretrieval from Global Ozone Monitoring Experiment (GOME) Data Using a Neural Network Approach (Neural Network Ozone Retrieval System (NNORSY)), J. Geophys. Res., 108(D16), 4497.

[13] Penner, J.E., et al., 2001. Aerosols, their Direct and Indirect Effects, Chapter 5 in Climate Change 2001, The Scientific Basis, Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change (IPCC), Cambridge University Press, 289-348.

[14] Pope, A. C., et al., 2002. Lung Cancer, Cardiopulmonary Mortality, and Long-term Exposure to Fine Particulate Air Pollution", J. Amer. Med. Assoc. 287: 1132-1141

[15] Radosavljevic, V., Vucetic, S., Obradovic, Z., 2008. Spatio-Temporal Partitioning for Improving Aerosol Prediction Accuracy, In Proceedings of the SIAM International Conference on Data Mining, SDM2008, pp. 609-620.

[16] Remer, L. A., Tanré, D., and Kaufman, Y. J., 2006. Algorithm for Remote Sensing of Tropospheric Aerosol from MODIS for Collection 005", http://modis-atmos.gsfc.nasa.gov/atbd02.pdf

[17] Rosenfeld, D., and Woodley, W., 2001. Pollution and Clouds, Physics World, 33-37.

[18] Vucetic, S., Han, B., Mi, W., Li, Z., Obradovic, Z., 2008. A Data Mining Approach for the Validation of Aerosol Retrievals, IEEE Geoscience and Remote Sensing Letters, Vol. 5 (1), 113-117.

# OcVFDT: One-class Very Fast Decision Tree for One-class Classification of Data Streams *

Chen Li
College of Information Engineering
Northwest A&F University, China
Yangling, Shaanxi Province, P.R. China, 712100
lichen_0810@nwsuaf.edu.cn

Yang Zhang†
College of Information Engineering
Northwest A&F University, China
Yangling, Shaanxi Province, P.R. China, 712100
zhangyang@nwsuaf.edu.cn

Xue Li
School of Information Technology and Electrical Engineering
The University of Queensland, Australia
Brisbane, Queensland 4072 Australia
xueli@itee.uq.edu.au

## ABSTRACT

Current research on data stream classification mainly focuses on supervised learning, in which a fully labeled data stream is needed for training. However, fully labeled data streams are expensive to obtain, which make the supervised learning approach difficult to be applied to real-life applications. In this paper, we model applications, such as credit fraud detection and intrusion detection, as a one-class data stream classification problem. The cost of fully labeling the data stream is reduced as users only need to provide some positive samples together with the unlabeled samples to the learner. Based on VFDT and POSC4.5, we propose our OcVFDT (One-class Very Fast Decision Tree) algorithm. Experimental study on both synthetic and real-life datasets shows that the OcVFDT has excellent classification performance. Even 80% of the samples in data stream are unlabeled, the classification performance of OcVFDT is still very close to that of VFDT, which is trained on fully labeled stream.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*data mining*; I.2.6 [**Artificial Intelligence**]: Learning—*concept learning*; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

†Corresponding author: Yang Zhang.
*This work is supported by Young Cadreman Supporting Program of Northwest A&F University (01140301) and Australian ARC Discovery Project DP0558879.

Permission to make digital or hard copies of all or partof this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

## General Terms

Decision trees, Hoeffding bound, one-class data streams, incremental learning

## 1. INTRODUCTION

In real world applications, such as credit fraud detection, network intrusion detection, and so on, huge volume of data arrives continuously with high speed. These applications could be modeled as a data stream classification problem. Currently, the research community of data stream classification analysis mainly focus on supervised learning [4][8][11][18][19], which requires fully labeled data streams for training. However, under the data stream scenarios, it is too expensive to label the full stream manually, and therefore the supervised data stream learning algorithms are not applicable to real-life applications.

In the case of credit fraud detection, the user behaviors that cause bad economic effect could be looked as positive samples. For those behaviors which have not caused any bad effect yet, only after thorough investigation could we decide whether they are fraud or not. As it is expensive, and sometimes impossible to investigate their true class labels, it is better to use them as unlabeled samples. The same scenario could be observed in network intrusion detection. In this paper, we model these applications as a problem of one-class classification of data streams.

In one class classification problems, classifiers are trained to distinguish a class of objects (called the target class) from all other objects [23]. In this paper, we identify the following characters of one-class data stream classification:

1. **No negative training samples**: only positive samples and unlabeled samples could be observed from the training data stream.
2. **High speed input data**: the classifier should have the ability to process huge volumes of data which arrive at high speed.
3. **Limited memory space**: only limited memory space is available to the classifier, which means that the classifier has to scan the input samples for only once.

In this paper, based on VFDT [4] and POSC4.5 [2], we present our OcVFDT (One-class Very Fast Decision Tree)

algorithm for one-class classification of data streams. The experimental results on both synthetic and real-life dataset show that even with a high percentage of samples in the stream kept unlabeled, the classification performance of OcVFDT is very close to VFDT, which is trained on fully labeled data stream. It still has excellent classification performance even when 80% of the samples in the stream is unlabeled.

This paper is organized as follows. Section 2 reviews the related work. The proposed OcVFDT algorithm is presented in section 3. The detailed experiment setup and results are shown in section 4. Section 5 concludes this paper and gives our future work.

## 2. RELATED WORK

**One-class classification**. Current research on one-class classification can be divided into two categories: (1) Methods that focus on the construction of basic classifiers: Schölkopf *et al.* proposed to apply SVM to one-class classification problem [17]. Denis *et al.* proposed one-class decision tree in [2]. Calvo *et al.* gave two methods to enhance the positive naïve bayes (PNB) classifier [1]. Elkan *et al.* assumed that the labeled examples are selected randomly from the positive examples, and gave two approaches for one-class learning [5]. (2) Methods that convert the one-class classification problem into a traditional supervised learning problem or semi-supervised learning problem: In [7][15][21], for text classification, negative text documents are extracted from the unlabeled documents, and supervised leaner is trained with help of these negative documents. In [13], Lee *et al.* transformed all the unlabeled samples into negative samples, and used a linear function to learn from the noisy examples. In [20], Yu proposed MC (Mapping Convergence) method to incrementally label negative data from unlabeled data using the *margin maximization* property of SVM.

**Data Stream Classification**. There are two main approaches for classifying data streams: single classifier based approach and ensemble based approach. For single classifier based approach, the most well-known classifiers are VFDT [4][10] and CVFDT [11]. The CVFDT improves the VFDT so that it can deal with concept drift. After that, many decision tree based algorithms were proposed for data stream classification [9][12]. For ensemble based approaches, the initial papers use static majority voting [18][19], while the current trend is to use dynamic classifier ensembles [22][24]. Besides, Gama *et al.* presented UFFT algorithm which generated a forest of trees from data streams [8].

**One-class classification of Data Streams**. In [23], Zhang *et al.* proposed the problem of one-class classification of data streams. To the best of our knowledge, this is the only work devoted to one-class classification of data streams. However, the algorithm proposed in [23] could only tackle text streams, while the algorithm proposed in this paper could cope with general data streams, including text streams. The algorithm in [23] follows the ensemble based approach to cope with concept drift in the stream, while the algorithm in this paper follows the single classifier approach. However, in this paper, concept drift is not considered, and is left for our future research.

## 3. BUILDING O*C*VFDT

In this paper, we only consider data streams with discrete attribute values. For data streams with continuous attribute values, the continuous attributes could be discretized to discrete attributes [6], so as to transform the original data stream into a data stream with discrete attributes only.

Given a stream data $S = (s_1, s_2, \ldots, s_i, \ldots)$, where $s_i$ is a sample in $S$, $s_i =< X_i, y_i, l_i >$. Here, $X_i$ is an attribute vector with $n$ discrete attributes; $y_i \in \{-1, +1\}$, represents the class label of $X_i$; $l_i \in \{0, 1\}$, represents whether $y_i$ is available or not to the learner. Under the one-class classification scenario, only positive samples are labeled. So if $l_i = 1$, we are sure that $y_i = +1$; and if $l_i = 0$, the true class label of $X_i$ is unknown.

For simplicity, in this paper, concept drift is not considered. And we assume that the positive training samples and unlabeled samples are distributed uniformly in the data stream. If the positive training samples and/or unlabeled samples distribute unevenly in the stream, then concept drifting is presented in this stream. We will study this kind of stream in our future work.

### 3.1 One-class Information Gain

The information gain algorithm is widely used in decision tree algorithms to decide which attribute being selected as the next splitting attribute [4][16]. Denis *et al.* proposed POSC4.5 algorithm and demonstrated how to measure one-class information gain for static datasets [2]. Here, based on POSC4.5, for the input sample $s_i$ in data stream $S$, and attribute set $A = \{A_1, A_2, \ldots, A_j, \ldots, A_{|A|}\}$, formula (1) is used to measure one-class information gain for the current tree node, denoted by *node*:

$$OcIG(A_j) = Entropy(s_i) - \sum_{a \in A_j} \frac{|s_{i,a}|}{i} Entropy(s_{i,a}) \quad (1)$$

Here, $Entropy(s_i)$ and $Entropy(s_{i,a})$ represent the entropy of the set $\{s_1, s_2, \ldots, s_i\}$, and set $s_{i,a} = \{s | s \in \{s_1, s_2, \ldots, s_i\}, s_i(A_j) = a\}$, respectively, which is computed by the following formulas:

$$p_{1,set} = \min\{\frac{|POS_{node,set}|}{POS_{i,set}} \times PosLevel \times \frac{|UNL_{i,set}|}{|UNL_{node,set}|}, 1\}$$

$$p_{0,set} = 1 - p_{1,set} \quad (2)$$

$$Entropy(set) = -p_{1,set} \log_2 p_{1,set} - p_{0,set} \log_2 p_{0,set} \quad (3)$$

Here, we write $|POS_{node,set}|$ and $|POS_{i,set}|$ for counting the positive samples in *set* observed at the *node*, and the first $i$ samples in $S$ respectively; $|UNL_{node,set}|$ and $|UNL_{i,set}|$ for counting the unlabeled samples in *set* observed at the *node*, and the first $i$ samples in $S$ respectively. Please note that in order to collect these statistical data, we only need to update some counters whenever a new sample arrives. For the sake of efficiency and limited memory available, we do not save all the samples observed from the stream for multiple scans.

The *PosLevel* in formula (2) is to estimate the probability of the observed positive samples from the data stream $S$. For a given classification task, *PosLevel* is always unknown to the learner. The next subsection will give more discussions on this parameter.

### 3.2 Building OcVFDT

In [4], Domingos *et al.* proposed to choose the best split attribute when constructing decision tree for data streams based on Hoeffding bound. Here, based on Hoeffding bound,

we use the one-class information gain as the split evaluation function to choose the best splitting attribute. As the true value of $PosLevel$ in formula (2) is unknown, we enumerate nine possible values of $PosLevel$, from 0.1 to 0.9. Then, we get a forest, $T$, with nine different OcVFDTs.

The best tree in $T$ is then chosen by estimating the classification performance of the trees with a chunk of validating samples. The validating chunk is filled with validating samples which are selected randomly with probability $p_{validate}$ from $S$. Note that due to the limited memory available, we cannot save all the validating samples that have been selected. Once the validating chunk is full, the samples inside it will be used to evaluate the trees in $T$, and then the validating chunk is cleared. Finally, these evaluation data will be used to choose the best tree as the final output classifier. Our algorithm for building one-class very fast decision tree is illustrated in Algorithm 1.

**Algorithm 1** Building one-class very fast decision tree

**Input:**
a stream of samples, $S$;
the size of validating chunk, $n_{validate}$;
the probability for a sample to be selected as validating sample, $p_{validate}$;
the feature space of stream $S$, $A$;
one minus the desired probability of choosing the correct attribute at any given node, $\delta$;
user-supplied tie threshold, $\tau$;
the number of samples between checks for growth, $n_{min}$.

**Output:**
a one-class very fast decision tree.

1: $ValidateChunk = \phi, T = \phi$;
2: **for** each $i \in [1, 9]$ **do**
3: initialize a tree $T_i$ with only a leaf (the root);
4: $T = T \bigcup T_i$;
5: **end for**
6: **for** each sample $s \in S$ **do**
7: **for** each $i \in [1, 9]$ **do**
8: $PosLevel = \frac{i}{10}$;
9: $Grow(T_i, PosLevel, s, A, \delta, \tau, n_{min})$;
10: **end for**
11: **if** $Random() \leq p_{validate}$ **then**
12: $ValidateChunk = ValidateChunk \bigcup \{s\}$;
13: **if** $|ValidateChunk| == n_{validate}$ **then**
14: $Estimate(T, ValidateChunk)$;
15: $ValidateChunk = \phi$;
16: **end if**
17: **end if**
18: **end for**
19: **if** $ValidateChunk! = \phi$ **then**
20: $Estimate(T, ValidateChunk)$;
21: **end if**
22: **return** $GetBestTree(T)$;

In this algorithm, the system is initialized in steps 1 to step 5; a forest of 9 trees is trained on $S$ in steps 7 to step 10; the validating chunk is maintained and the trees are evaluated on it in step 11 to steps 21. The function $Grow(T_i, PosLevel, s, A, \delta, \tau, n_{min})$ is used to grow a single one-class decision tree, please refer to section 3.3 for details. The function $Random()$ returns a random value in $[0, 1]$. The function $Estimate(T, ValidateChunk)$ is used to evaluate the classification performance of trees in $T$ on the validating chunk. And the function $GetBestTree(T)$ returns the best tree selected from forest $T$. These two functions are demonstrated in section 3.4.

## 3.3 Growth of Single OcVFDT

Based on the growth algorithm of VFDT [4], the process of growing single OcVFDT is listed in Algorithm 2.

**Algorithm 2** $Grow(T_i, PosLevel, s, A, \delta, \tau, n_{min})$

// Refer to Algorithm 1 for the details of input parameters;

1: $node = T_i.sort(s)$;
2: $n_{node} = getNumOfSamplesAtNode(node)$;
3: Set the class label of $node$ to the majority class at $node$ according to $p_1$ and $p_0$ following formula (2);
4: **if** $numOfClassesAtNode(node) > 1$ and $n_{node} \% n_{min} == 0$ **then**
5: **for** each $A_i \in node.A$ **do**
6: Compute $G_{node}(A_i) = OcIG(A_i)$ following formula (1);
7: **end for**
8: Choose attribute $A_a$ and $A_b$ with the highest and second-highest $G_{node}$;
9: $\Delta G_{node} = G_{node}(A_a) - G_{node}(A_b)$;
10: Compute $\epsilon$ following the *Hoeffding bound* [4];
11: **if** $\Delta G_{node} > \epsilon$ or $\Delta G_{node} \leq \epsilon < \tau$ **then**
12: **for** each $a_i \in A_a$ **do**
13: $node_i = addChildLeaf()$;
14: Set the class label of $node_i$ to the majority class at $node_i$ according to $p_1$ and $p_0$ following formula (2);
15: **end for**
16: **end if**
17: **end if**

Algorithm 2 could be summarized into two main processes. Firstly, from steps 1 to 3, the sample $s$ is traversed along the tree $T_i$ to reach leaf $node$, and the class label of $node$ is set to the majority class at $node$. Secondly, from steps 4 to 17, new leaves are grown at $node$. For each available attributes at $node$, the one-class information gain is derived using formula (1). If the condition at step 11 is satisfied, then $node$ should be split by attribute $A_a$, and new leaves should be generated. Here, note that when computing the majority class at new leaves, the count of positive and unlabeled samples comes from the parent node.

Let's write $v$ for the maximum number of values per attribute, $c$ for the number of classes, $l$ for the maximum depth of the tree $l < |A|$, and $|S|$ for the number of samples in data stream $S$, the time complexity of Algorithm 1 is $O(|S||A|vcl) + O(|S||A|vclp_{validate}) = O(|S|)$.

## 3.4 Tree Selection

A forest $T$ with nine trees is trained in Algorithm 1. Based on the POSC4.5 [2] algorithm, we design our algorithm for choosing the best tree as the final output classifier.

We collect the following statistical data with the help of the validating chunk of each tree $T_k \in T$.

1. $|POS|$: the count of positive samples in the validating chunk;

2. $|UNL|$: the count of unlabeled samples in the validating chunk;

3. $|PToU_k|$: the count of positive samples that are classified as unlabeled samples by $T_k$;

4. $|UToP_k|$: the count of unlabeled samples that are classified as positive samples by $T_k$.

Once the validating chunk is full, the function $Estimate(T, ValidateChunk)$ will be invoked to collect the statistical data. And the counting data among different runs should be accumulated together.

The following formula is used to evaluate the performance of each $T_k \in T$ :

$$e(T_k) = \frac{|PToU_k|}{|POS|} + \frac{|UToP_k|}{|UNL|} \tag{4}$$

And the best tree is chosen by:

$$T_j = \underset{k}{\operatorname{argmin}}(e(T_k)) \qquad 1 \leq k \leq 9 \tag{5}$$

# 4. EMPIRICAL STUDY

In order to measure the classification performance of our OcVFDT algorithm, we perform experiments on both synthetic dataset and real-life dataset. We compare the classification of OcVFDT with VFDT, as both OcVFDT and VFDT are the single-classifier based approaches. Furthermore, as VFDT is a supervised learner, this comparison can also help reveal the strong ability of OcVFDT to learn from unlabeled samples.

Our algorithms are implemented in Java language based on the WEKA[1] software packages, and the experiments are conducted on a PC with Core 2 CPU, 1G memory and Windows XP OS.

We measure the classification performance of the proposed classifier by accuracy and F1, which are commonly used in the research community for measuring the performance of one-class classifiers [2][5].

## 4.1 Synthetic Data

For each of the experiments settings here, twenty trails of experiments on synthetic data are conducted, and the averaged classification performance is reported.

In the rest of this paper, we write $PosLevel$ for the percentage of positive samples in the stream.

Similar to the experiment setting in [4], we set $\tau = 0.05$, and $\delta = 0.0000001$ for OcVFDT in all experiments.

### 4.1.1 Generating Synthetic Data Stream

We modify the software in VFML[2] package, which is used to generate synthetic data stream for VFDT [4], by adding a new parameter, $UnLevel$. $UnLevel$ represents the percentage of unlabeled samples in the stream. For each sample in the stream $S$, there are 100 discriminative attributes with binary attribute value, and 1 category attribute. The testing data stream is set to have the same data distribution and the same size as the training data stream.

[1]http://www.cs.waikato.ac.nz/ml/weka/
[2]http://www.cs.washington.edu/dm/vfml/

### 4.1.2 Analysis of Parameters in OcVFDT

**Parameter $n_{validate}$ and $p_{validate}$.** Two groups of experiments are conducted here, with $n_{validate}$ ranging from 200 to 500 in one group, and $p_{validate}$ ranging from 0.3 to 0.6 in the other. We set $|S| = 10^5$ and $UnLevel = 80\%$. The experiment result shows that, for both F1 and accuracy, the difference of classification performance among different experiment settings is non-significant on t-Test [3]. The detailed experiment result is omitted here for lacking of space. We set $n_{validate} = 200$, and $p_{validate}$=0.3 in all experiments for the sake of limited memory space.

**Parameter $n_{min}$.** We experimented with $|S| = 10^4$, $|S| = 10^5$ and $|S| = 10^6$, respectively, with $UnLevel = 80\%$, and $PosLevel = 50\%$. Please refer to Fig.1 for the experiment results. In Fig.1, the horizontal axis represents $n_{min}$, the vertical axis represents classification performance. Fig.1(A), Fig.1(B), and Fig.1(C) gives the experiment result for $|S| = 10^4$, $|S| = 10^5$, and $|S| = 10^6$, respectively.

It can be observed from Fig.1 that with increasing of $n_{min}$, the classification performance is decreasing. Hence, for the rest of this paper, we set $n_{min} = 200$.

### 4.1.3 OcVFDT vs. VFDT

**Comparison of Classification Performance.** In this group of experiments, we compare the classification performance of VFDT and OcVFDT. For VFDT, we set $\tau = 0.05$, $\delta = 0.0000001$, and $n_{min} = 200$.

We experimented with $|S| = 10^4$, $10^5$, $10^6$ and $10^7$; $PosLevel = 40\%$, 50%, and 60%; $UnLevel = 55\%$, 60%, 65%, 70%, 75%, and 80%. The experiment results are shown in Fig.2 and Fig.3. In Fig.2 and Fig.3, the horizontal axis represents $|S|$, the vertical axis represents accuracy, and F1, respectively. The different lines in these figures represent the classification result of VFDT, and OcVFDT with various $UnLevel$ parameter.

It could be observed from Fig.2 and Fig.3 that for a certain $|S|$ value, the classification performance of OcVFDT with different $UnLevel$ is very close to each other. It could also be observed that OcVFDT performs a little worse than VFDT, and for some $|S|$ value, it even performs better than VFDT. When $|S| = 10^7, UnLevel = 80\%$, the averaged accuracy of VFDT is 0.961, while the averaged accuracy of OcVFDT is 0.953. Compared with VFDT, the decrease in accuracy of OcVFDT by 0.008 is compensated by freeing the human power to label $8 \times 10^6$ samples. This is a great achievement of saving human power, and makes OcVFDT more applicable to real-life applications.

**Experiment on Unlabeled Samples.** In this group of experiments, we test the ability of OcVFDT to cope with unlabeled data. We set $PosLevel = 50\%$ for generating data stream $S$. When experimenting with VFDT, we sample $|POS|$ positive samples and $|POS|$ negative samples from $S$ to form the training data stream. When experimenting with OcVFDT, we use $|POS|$ positive samples from $S$ as labeled sample, and $|POS|+|S|(i-1)/10$ samples from $S$ as unlabeled samples to form the training data stream. Here, when we are generating training data streams $S$ for both VFDT and OcVFDT, we make sure that $PosLevel = 50\%$ in the generated stream, and keep the order among samples in $S$ to the generated data stream. By using this method, the assumption of even distribution of original data is broken which leads the performance of OcVFDT not so well.

Tab.1 shows the experiment results. In Tab.1, column

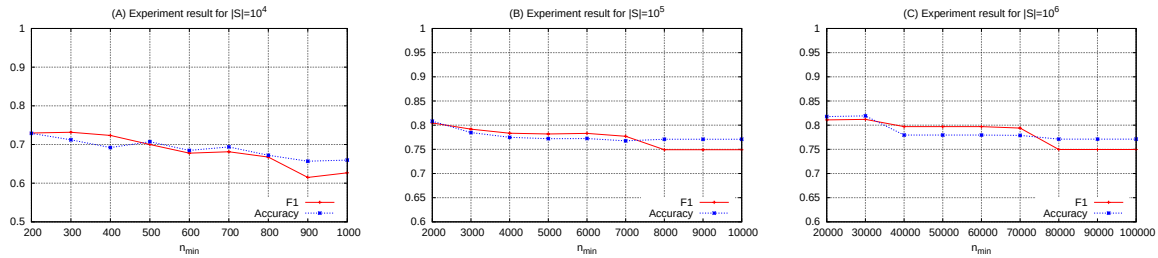



**Figure 1: Experiment with $n_{min}$.**

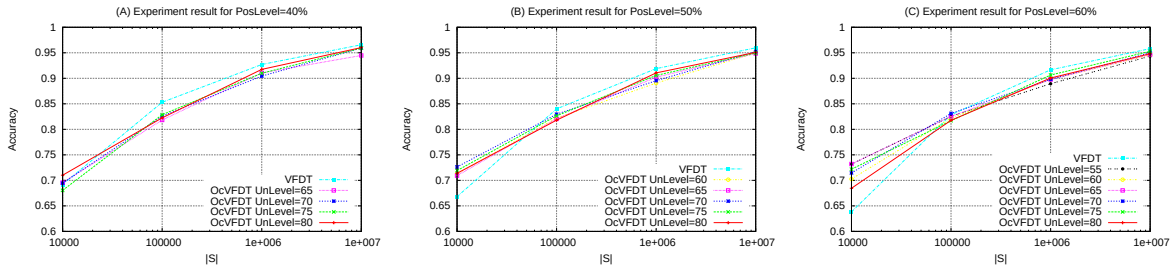



**Figure 2: Comparison of VFDT and OcVFDT Measured by Accuracy.**

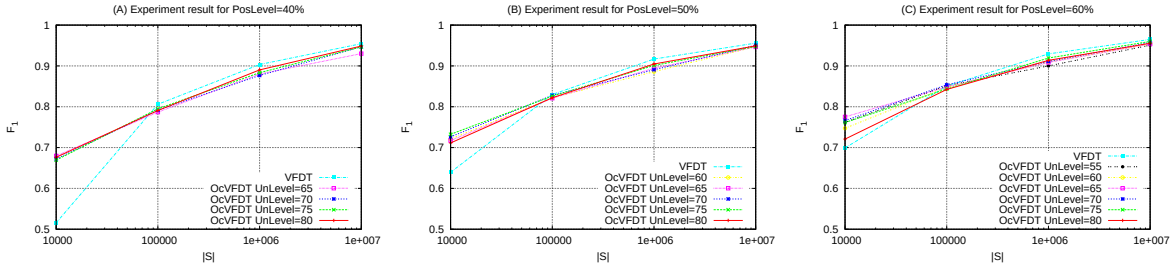



**Figure 3: Comparison of VFDT and OcVFDT Measured by F1.**

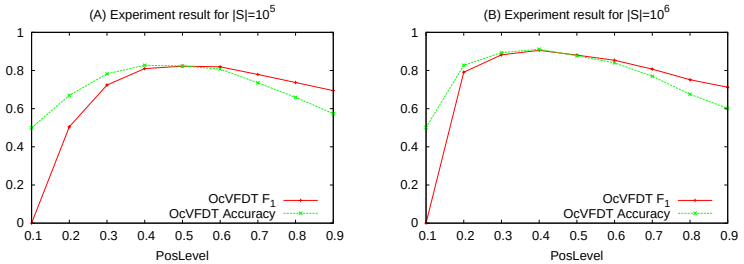



**Figure 4: Experiment with Tree Selections.**

**Table 1: Results of Different Sizes of Unlabeled Samples.**

| $\|S\|$ | $\|POS\|$ | VFDT | | OcVFDT | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $i=1$ | | $i=2$ | | $i=3$ | | $i=4$ | | $i=5$ | | $i=6$ | | $i=7$ | |
| | | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc | F1 | Acc |
| $10^4$ | 2000 | 0.606 | 0.614 | 0.668 | 0.501 | 0.668 | 0.501 | 0.668 | 0.501 | 0.667 | 0.607 | 0.689 | 0.625 | 0.721 | 0.670 | 0.729 | 0.715 |
| | 2500 | 0.603 | 0.618 | 0.667 | 0.501 | 0.667 | 0.501 | 0.665 | 0.557 | 0.667 | 0.586 | 0.708 | 0.664 | 0.709 | 0.706 | - | - |
| | 3000 | 0.572 | 0.621 | 0.666 | 0.500 | 0.660 | 0.552 | 0.667 | 0.579 | 0.713 | 0.651 | 0.733 | 0.710 | - | - | - | - |
| | 3500 | 0.522 | 0.631 | 0.660 | 0.512 | 0.673 | 0.573 | 0.690 | 0.630 | 0.711 | 0.696 | - | - | - | - | - | - |
| | 4000 | 0.507 | 0.636 | 0.694 | 0.605 | 0.719 | 0.669 | 0.742 | 0.734 | - | - | - | - | - | - | - | - |
| $10^5$ | 20000 | 0.786 | 0.788 | 0.706 | 0.597 | 0.733 | 0.660 | 0.749 | 0.683 | 0.779 | 0.746 | 0.789 | 0.759 | 0.801 | 0.779 | 0.824 | 0.818 |
| | 25000 | 0.807 | 0.798 | 0.732 | 0.661 | 0.741 | 0.665 | 0.773 | 0.730 | 0.791 | 0.765 | 0.804 | 0.791 | 0.822 | 0.825 | - | - |
| | 30000 | 0.811 | 0.812 | 0.697 | 0.572 | 0.787 | 0.757 | 0.799 | 0.780 | 0.801 | 0.781 | 0.825 | 0.824 | - | - | - | - |
| | 35000 | 0.817 | 0.824 | 0.774 | 0.731 | 0.792 | 0.769 | 0.793 | 0.765 | 0.824 | 0.822 | - | - | - | - | - | - |
| | 40000 | 0.824 | 0.832 | 0.785 | 0.758 | 0.788 | 0.766 | 0.824 | 0.824 | - | - | - | - | - | - | - | - |
| $10^6$ | 200000 | 0.891 | 0.894 | 0.845 | 0.831 | 0.827 | 0.795 | 0.865 | 0.852 | 0.868 | 0.858 | 0.884 | 0.877 | 0.902 | 0.902 | 0.906 | 0.917 |
| | 250000 | 0.893 | 0.896 | 0.800 | 0.752 | 0.867 | 0.856 | 0.870 | 0.861 | 0.873 | 0.862 | 0.896 | 0.895 | 0.901 | 0.905 | - | - |
| | 300000 | 0.897 | 0.899 | 0.862 | 0.849 | 0.879 | 0.875 | 0.859 | 0.848 | 0.895 | 0.896 | 0.897 | 0.902 | - | - | - | - |
| | 350000 | 0.901 | 0.903 | 0.876 | 0.871 | 0.850 | 0.834 | 0.894 | 0.895 | 0.897 | 0.904 | - | - | - | - | - | - |
| | 400000 | 0.910 | 0.913 | 0.835 | 0.812 | 0.885 | 0.883 | 0.885 | 0.890 | - | - | - | - | - | - | - | - |

**Table 2: Running Time and Size of the Tree.**

| $\|S\|$ | VFDT | | | OcVFDT | | |
|---|---|---|---|---|---|---|
| | Time(ms) | #Leaves | #Node | Time(ms) | #Leaves | #Node |
| 20000 | 91.00 | 8.6 | 7.6 | 680.84 | 11.2 | 10.2 |
| 200000 | 691.92 | 64.4 | 63.4 | 8004.48 | 65.8 | 64.8 |
| 2000000 | 7335.68 | 313.0 | 312.0 | 106935.90 | 441.8 | 440.8 |
| 20000000 | 78450.00 | 1529.0 | 1528.0 | 1459911.00 | 3775.0 | 3774.0 |

1 lists the size of data stream $S$; column 2 gives the size of (labeled) positive sample set, $|POS|$; columns 3 and 4 give the experiment results of VFDT, measured in F1 and accuracy respectively; columns 5 to 18 give the experiment results of OcVFDT with different $i$ values. It is shown in Tab.1 that from $i = 1$ to 7, the classification performance of OcVFDT is improving, because the total number of training samples is increasing and the distribution of the training set is becoming more and more uniform.

#### 4.1.4 Experimentation of Tree Selections

In this group of experiments, we examine the ability of our tree selection algorithm for choosing the best trees. When generating data streams, we set $PosLevel = 50\%$, and $UnLevel = 80\%$. Experiment results of $|S| = 10^5$ and $|S| = 10^6$ are reported in Fig.4(A) and Fig.4(B) respectively. In Fig.4, the horizonal axis represents nine OcVFDTs trained with nine different $PosLevel$ values, and the vertical axis represents the classification performance. In Fig.4, "$T_i$ *F1*", "$T_i$ *Accuracy*", "*OcVFDT F1*", and "*OcVFDT Accuracy*" represents the F1 index of tree $T_i$, the accuracy index of $T_i$, the F1 index of the selected tree, and the accuracy index of the selected tree respectively.

It is shown in Fig.4 that our tree selection algorithm is very effective in choosing the best tree from the forest $T$ and in the 20 trails experiments both in $|S| = 10^5$ and $|S| = 10^6$. For $|S| = 10^5$, there are 7 trails which select the top tree, 10 trails select the second tree and 3 trails choose the third tree. For $|S| = 10^6$, all the trails select the top tree. For lack of space in this paper, we can't report the results of other $UnLevel$ values.

#### 4.1.5 Running Time and Size of the Tree

Here, we report the running time, the count of inner nodes, and the count of leaves of VFDT and OcVFDT with different $|S|$ parameters. When generating data streams, we set $PosLevel = 50\%$, $UnLevel = 80\%$. The running time reported here is the time for the leaner to learn from the samples. The time for input/output is not considered.

The experiment results are listed in Tab.2. In Tab.2, column 1 lists the size of the data stream, $|S|$; columns 2, 3, and 4 list the running time, the count of leaf nodes, the count of inner nodes for VFDT respectively; and columns 5,6, and 7 give these information for OcVFDT.

### 4.2 Real-life Data

We perform experiments on one real-life dataset, namely Reuters Corpus Volume 1 (RCV1) [14].

#### 4.2.1 Datasets and Preprocessing

**RCV1.** Reuters Corpus Volume 1 (RCV1) consists of English language stories produced by Reuters journalists between August 20, 1996 and August 19, 1997. To make it easier to use, Lewis *et al.* provided RCV1-V2 [3] after preprocessing on RCV1, which becomes a benchmark dataset [14] for text mining researchers.

There are four main categories in RCV1-V2, i.e., ECAT, GCAT, CCAT, and MCAT. We select CCAT as a positive category in our experiment, and the rest of the documents are taken as negative samples, so as to generate a balanced data stream.

Usually, there are thousands of features in a corpus. We performed feature selection by using WEKA in the experi-

[3] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

ment. As RCV1-V2 is too large for WEKA to process, for CCAT category, 3% of the training samples and 3% of the testing samples are selected randomly to form dataset $D$, and 100 top expressive features are selected by performing Information Gain (IG) [16] algorithm on $D$. The original text corpus is represented by these features.

The details of the dataset after the preprocessing are follows.

**Table 3: Description of the Real Dataset.**

| Dataset | #Att | Training Set | | Testing Set | |
|---|---|---|---|---|---|
| | | #Pos | #Neg | #Pos | #Neg |
| RCV1-V2 | 101 | 370541 | 410724 | 10786 | 12363 |

### 4.2.2 Experiment Result

We simulate data stream on this real-life dataset. For RCV1-V2 dataset, $UnLevel$ ranges from 65% to 80%. For each experiment setting, we generate fifty trails of data streams, and the averaged experiment results of the fifty trails are reported in Tab.4.

In Tab.4, column 1 lists the name of the dataset; columns 2 to 6 give the F1 index, accuracy index, running time, count of leaves, and count of inner nodes of VFDT respectively; column 7 gives the $UnLevel$ parameter; and columns 8 to 12 give the experiment results for OcVFDT.

It is obvious from Tab.4 that even with high percentage of unlabeled samples, the performance of OcVFDT is very competitive to that of VFDT, which is trained on fully labeled data stream. Furthermore, with different $UnLevel$ values, the performance of OcVFDT is stable.

## 5. CONCLUSION AND FUTURE WORK

In this paper, based on VFDT and POSC4.5, we propose OcVFDT algorithm for one-class classification of data streams. Our experiments on both synthetic data and real-life data show that even 80% of the samples in the stream are unlabeled, the classification performance of OcVFDT is still very close to that of VFDT which is trained on fully labeled data stream. For example, in synthetic datasets with 80% unlabeled samples, the decrease accuracy of OcVFDT of $|S| = 10^5, 10^6$ and $10^7$ with $PosLevel = 50\%$ is 0.022, 0.008 and 0.009 respectively.

OcVFDT requires less labeled training samples, which makes it more applicable to real-life applications.

Data streams are characterized by concept drift. However, concept drift is not considered in this paper for simplicity. Based on our OcVFDT algorithm, we plan to study one-class data stream classification algorithm to cope with concept drift as our next effort.

## 6. REFERENCES


[1] B. Calvo, P. Larranaga, and J. A. Lozano. Learning Bayesian classifiers from positive and unlabeled examples. *Pattern Recognition Letters*, 28:2375–2384, 2007.

[2] F. Denis, R. Gilleron, and F. Letouzey. Learning from positive and unlabeled examples. *Theoretical Computer Science*, pages 70–83, 2005.

[3] T. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7):1895–1923, 1998.

[4] P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'00)*, pages 71–80. ACM New York, NY, USA, 2000.

[5] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'08)*.

[6] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[7] G. Fung, J. Yu, H. Lu, and P. Yu. Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):6–20, 2006.

[8] J. Gama, P. Medas, and P. Rodrigues. Learning Decision Trees from Dynamic Data Streams. *Journal of Universal Computer Science*, 11(8):1353–1366, 2005.

[9] J. Gama, R. Rocha, and P. Medas. Accurate decision trees for mining high-speed data streams. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'03)*, pages 523–528. ACM Press New York, NY, USA, 2003.

[10] G. Hulten, P. Domingos, and L. Spencer. Mining massive data streams. In *The Journal of Machine Learning Research*, 2005.

[11] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'01)*, pages 97–106. ACM New York, NY, USA, 2001.

[12] R. Jin and G. Agrawal. Efficient decision tree construction on streaming data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'03)*, pages 571–576. ACM New York, NY, USA, 2003.

[13] W. Lee and B. Liu. Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. In *Proceedings of Twentieth International Conference on Machine Learning. (ICML'03)*, volume 20, page 448, 2003.

[14] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[15] B. Liu, Y. Dai, X. Li, W. Lee, and P. Yu. Building text classifiers using positive and unlabeled examples. In *Proceedings of the Third IEEE International Conference on Data Mining. (ICDM'03)*, pages 179–186, 2003.

[16] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[17] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson. Estimating the Support of a

**Table 4: Result of Experiments on Real Dataset.**

| Dataset | VFDT | | | | | OcVFDT | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | Acc | Time(ms) | #Leaves | #Node | *UnLevel* | F1 | Acc | Time(ms) | #Leaves | #Node |
| RCV1-V2 | 0.812 | 0.837 | 4550.05 | 184 | 183 | 80 | 0.801 | 0.830 | 49152.78 | 199.30 | 198.30 |
| | | | | | | 75 | 0.802 | 0.830 | 49144.54 | 196.52 | 195.52 |
| | | | | | | 70 | 0.802 | 0.827 | 50019.94 | 199.92 | 198.92 |
| | | | | | | 65 | 0.795 | 0.827 | 50009.84 | 197.76 | 196.76 |


High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[18] W. Street and Y. Kim. A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'01)*, pages 377–382. ACM New York, NY, USA, 2001.

[19] H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (SIGKDD'03)*, pages 226–235. ACM New York, NY, USA, 2003.

[20] H. Yu. Single-Class Classification with Mapping Convergence. *Machine Learning*, 61(1):49–69, 2005.

[21] H. Yu, J. Han, and K. Chang. PEBL: web page classification without negative examples. *IEEE Transactions on Knowledge and Data Engineering*, 16(1):70–81, 2004.

[22] Y. Zhang and X. Jin. An automatic construction and organization strategy for ensemble learning on data streams. *ACM SIGMOD Record*, 35(3):28–33, 2006.

[23] Y. Zhang, X. Li, and M. Orlowska. One-Class Classification of Text Streams with Concept Drift. In *Proceedings of the Third IEEE International Conference on Data Mining Workshops. (ICDMW'08)*, pages 116–125, 2008.

[24] X. Zhu, X. Wu, and Y. Yang. Dynamic Classifier Selection for Effective Mining from Noisy Data Streams. In *Proceedings of the Fourth IEEE International Conference on Data Mining. (ICDM'04)*, pages 305–312, 2004.

# A Frequent Pattern Based Framework for Event Detection in Sensor Network Stream Data*

Li Wan[1,2]
[1]Beijing University of Posts and Telecommunications, Beijing, China 100876
[2]EBUPT Information Technology Co., Ltd, Beijing, China 100191
wanli@ebupt.com

Jianxin Liao[1,2]
[1]Beijing University of Posts and Telecommunications, Beijing, China 100876
[2]EBUPT Information Technology Co., Ltd, Beijing, China 100191
liaojx@bupt.edu.cn

Xiaomin Zhu[1,2]
[1]Beijing University of Posts and Telecommunications, Beijing, China 100876
[2]EBUPT Information Technology Co., Ltd, Beijing, China 100191
zhuxm@bupt.edu.cn

## ABSTRACT
In this paper, we presented a frequent pattern based framework for event detection in stream data, it consists of frequent pattern discovery, frequent pattern selection and modeling three phases: In the first phase, a MNOE (Mining Non-Overlapping Episode) algorithm is proposed to find the non-overlapping frequent pattern in time series. In the frequent pattern selection phase, we proposed an EGMAMC (Episode Generated Memory Aggregation Markov Chain) model to help us selecting episodes which can describe stream data significantly. Then we defined feature flows to represent the instances of discovered frequent patterns and categorized the distribution of frequent pattern instances into three categories according to the spectrum of their feature flows. At last, we proposed a clustering algorithm EDPA (Event Detection by Pattern Aggregation) to aggregate strongly correlated frequent patterns together. We argue that strongly correlated frequent patterns form events and frequent patterns in different categories can be aggregated to form different kinds of events. Experiments on real-world sensor network datasets demonstrate that the proposed MNOE algorithm is more efficient than the existing non-overlapping episode mining algorithm and EDPA performs better when the input frequent patterns are maximal, significant and non-overlapping.

## Categories and Subject Descriptors
H.2.8 [**Information Systems**]: Database Management-Data mining

## General Terms
Algorithms

## Keywords
Frequent pattern, temporal data mining, event detection, sensor network

## 1. INTRODUCTION
Stream data refers to object sequences annotated with ordered time stamps. In this paper, "stream data", "temporal data" and "time series" represent the same concept. Frequent patterns are subsequences of objects that appear in a time series with frequency no less than a user-specified threshold (i.e. minimum support). Each occurrence of a frequent pattern is an instance of this frequent pattern in the given time series.

Abundant literature has been dedicated to find frequent patterns in temporal data [1,2,3,4,5,6,7]. Frequent patterns are usually extracted from time series as the features of the time series for different data mining tasks, such as clustering, classification and mining associate rules [8,9,10,11,12,13]. But the methods developed for mining closed frequent patterns often generate a huge number of patterns. People usually need interesting patterns only [14]. In time series, frequent patterns can be categorized into different categories, patterns in different category can represent different features of stream data which might be interesting patterns in different tasks. As shown in Fig.1, different categories of frequent patterns represent different relationships among frequent pattern instances (y, z axis) and their different distributions in time series (x axis). The concepts mentioned in Fig.1 are defined in section 2.

There are different kinds of events, such as burst event, regular event and routine event. For example "burst event" are some events may frequently occur in a special short period of time but not or almost not occur in any other period of time. As events consist of frequent patterns, the features of events can be described by the frequent patterns associated with them. Take the time series in Fig.2 as an example, if we set minimum support as 3 and constraint the length of episode no more than 5, let A, B, C, D denote four episodes consist of 5 objects respectively (see Def. 5). It's obverse that episode A, B, C, D are all frequent, namely appear more than 3 times respectively. But it's clear that episode A and D don't occur frequently in all the scope of the time series. If we split the time series into three pieces, we could say that event A and D burst in the second time frame. Episode A and D are both burst episodes and they form a burst event, namely "AD". We can consider frequent patterns as elements form events in time series.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09*, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7…$5.00.

*This work was jointly supported by: (1) National Science Fund for Distinguished Young Scholars (No. 60525110); (2) National 973 Program (No. 2007CB307100, 2007CB307103); (3) Program for New Century Excellent Talents in University (No. NCET-04-0111); (4) Development Fund Project for Electronic and Information Industry (Mobile Service and Application System Based on 3G).

Different category of frequent patterns form different category of events.

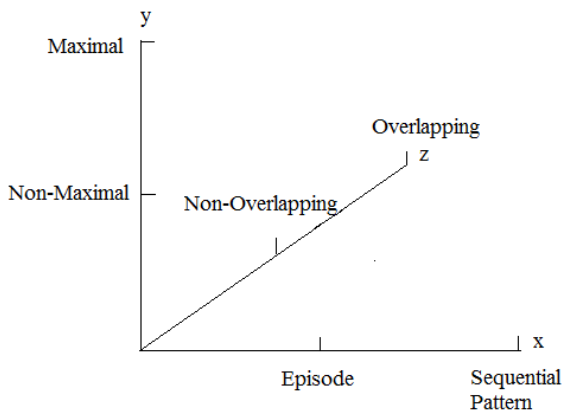



**Fig.1 Frequent patterns are categorized by the three dimensions. i.e. the inclusion relationship (maximal or non-maximal), the occurrences in time series (episode or sequential pattern) and the positions of instances in the time series (overlapping or non-overlapping).**

| | | |
|---|---|---|
| BCCBBC | AADADAAD | BCBBAC |

**Fig.2 Episode "A" and "D" burst in the second piece of time, and they form a burst event "AD".**

In this paper, we focus on discovering non-overlapping episode and selecting frequent pattern in different categories for detecting different kinds of events in temporal data. The contributions of this paper are as follows:

- We categorize frequent patterns in time series by the relationship among their instances and the instance distribution. We defined feature flows to represent frequent patterns and we can get the feature of frequent pattern instance distribution by analyzing the spectral gram of corresponding feature flow.
- We presented MNOE algorithm to discover non-overlapping episode and defined significant non-overlapping episode. Based on the presented EGMAMC model, we induced the minimum support for discovering the defined significant non-overlapping episode (see section 3.2.2).
- We defined different categories of frequent patterns and events in a time series. We connect them by instance distribution of frequent patterns. We presented EDPA algorithm to aggregate frequent patterns in different category to get different category of event.
- Experiments on sensor network datasets demonstrate that MNOE beats the existing algorithm. EDPA can detect events in high accuracy and distinguishing non-overlapping episode from overlapping episode, significant episode from non-significant episode help improving the accuracy of EDPA.

The rest of the paper is organized as follows: Section 2 is preliminaries. Section 3 presents the frequent patterns based framework for event detection. The proposed algorithms (i.e. MNOE, EDPA) and EGMAMC model are presented in three subsections respectively. Section 4 gives related works. Section 5 gives the experiment results and we conclude in section 6.

## 2. PRELIMINARIES

Definition 1. (Time Series) A sequence of objects, $T = \{< o_1, t_1 >, < o_2, t_2 >, ..., < o_i, t_i >\}$ , where $O = \{o_1, o_2, ..., o_i\}$ represents the symbols of different types of objects and $t_i$ the time of occurrence of the $i$ th object.

Definition 2. (Frequent Pattern) A subsequence of object symbols (i.e. $P = \{o_1, o_j, ..., o_i\}$ ) that appear in a time series with frequency no less than a user-specified threshold (i.e. minimum support).

Definition 3. (Maximal Frequent Pattern) Given a frequent pattern $P_i = \{o_1, o_2, ..., o_m\}$ , there do not exist another frequent pattern $P_j = \{o_1, o_2, ..., o_n\}$ includes $P_i$ . If $P_i \subseteq P_j$ , then $\forall P_i(o_i) = P_j(o_j)$ , $\exists o_i = o_j$ , where $P_i(o_i)$ represents the index of symbol $o_i$ in $P_i$ .

By defining different measures of the contribution that the occurrences of objects made to supports, we can get different kinds of frequent patterns, namely sequential pattern and episode.

Definition 4. (Sequential Pattern) Given a set of time segments and a minimal support $\gamma_S$ . A sequence occurs once or more than once in a segment contributes 1 to its support. If a sequence's support is larger than $\gamma_S$ , it's a sequential pattern.

Definition 5. (Episode) Given a time series and a minimal support $\gamma_E$ . A sequence occurs once in this time series contributes 1 to its support. If a sequence's support is larger than $\gamma_E$ , it's an episode.

Laxman etc. [15] is the first one defines non-overlapping episode as follows:

Definition 6. (Non-overlapping episode) Suppose an episode occurs in the time series twice, if any object associated with either occurrence doesn't occur between the objects associated with the other occurrence, then these two instances of the episode are called non-overlapping episode. The frequency of an episode is defined as the maximum number of non-overlapping occurrences of the episode in time series.

To describe the distribution of frequent patterns in a time series, we defined feature flows.

Definition 7 (Feature Flow) Suppose a time series $T = \{< o_1, t_1 >, < o_2, t_2 >, ..., < o_i, t_i >\}$ is segmented into a segmentation $Seg = \{s(tf_1), s(tf_2), ..., s(tf_T)\}$ . A flow of frequent pattern feature can be written as the sequence:

$y_f = \{y_f(tf_1), y_f(tf_2), ..., y_f(tf_T)\}$, $y_f(tf_i)$ is a measure of frequent pattern feature $f$ in time frame $tf_i$, the sequence of time frame $\{tf_1, tf_2, ..., tf_T\}$ is a mapping of the segmentation $Seg$.

$y_f(tf) = \frac{N_p(s(tf))}{N(s(tf))} \times \log(\frac{N}{N_p})$, where $N_p(s(tf))$ is the number of occurrence of pattern $p$ in segment $s(tf)$. $N(s(tf))$ is the total number of occurrence of all the frequent patterns in segment $s(tf)$. $N$ and $N_p$ are the number of occurrence of all the frequent patterns and pattern $p$ in all segments respectively.

Definition 8 (Event) An event in time series is a set of frequent patterns $P = \{p_1, p_2, ..., p_n\}$ and any two patterns ( $p_i$ and $p_j$ ) must satisfy the following necessary conditions:

1. Their feature flows $y_{fi}$ and $y_{fj}$ must be identically distributed.
2. $p_i$ and $p_j$ must occur in the same time segments with high probability.

## 3. FREQUENT PATTERN BASED FRAMEWORK FOR EVENT DETECTION

The presented frequent pattern based event detection framework consists of frequent pattern discovery, frequent pattern selection and modeling. In frequent pattern discovery phase, we can use any existing algorithms to discovery various types of frequent patterns (sequential pattern and episode) and section 3.1 presents the proposed MNOE algorithm which can discovery non-overlapping episodes. For different data mining tasks, there are many methods have been proposed to select interesting frequent patterns from the closed set of frequent patterns. In section 3.2, we present a Markov chain based method to distinguish significant and non-significant episode and it can be used as a method to select frequent pattern. Finally, in modeling phase, different models can be trained from selected frequent patterns to achieve different tasks (e.g. clustering, classification and associate rule). In section 3.3, we present EDPA algorithm which clusters frequent patterns for event detection in stream data.

### 3.1 Finding Non-overlapping Episodes

The concept of non-overlapping episode was firstly proposed by S. Laxman etc. [15] (see Def. 6). But it's regret that S. Laxman etc. [15,16] only proposed an algorithm to find non-overlapping episodes from the complete set of episodes. It's obviously not efficient. So based on the same definition of non-overlapping episode and its frequency, we present a iterative algorithm which directly find non-overlapping episodes from the given time series.

Algorithm MNOE

Input:

1.a set $S$ of time annotated time series projection $P < head, body >$

2.The length of episode to be found in current iterate: $l$

3.The maximum time interval permitted between every two successive events in an episode: maxGap

4. minimum support: $Spt_{\min}$

Output: the set of non-overlapping episodes

1. Sorting $P_i$ increasingly in $S$ by the time stamp annoted to the last element in $P_i.head$
2. for (every projection $P_i$ in S){
3. for every element $e$ in $P_i.body$ {
4. If (substraction of $e.time_stamp$ and the time stamp of the last element in $P_i.head$ is larger than maxGap) then
5. Break;
6. If $e$ is the element belonging to |e| with smallest time stamp in $P_i.body$ and the time stamp of the first element in $P_i.head$ is larger than the time stamp of last element in HashtableFE(|e|), then
7. $e$ contribute 1 to the support of |e|.
8. Project $e$ on $P_i.body$: $P_i'$ =projection( $e$ );
9. $S'.add(P_i')$
10. For every key value (i.e.|e|) in HashtableFE
11. if HashtableFE(|e|).size is larger than $Spt_{\min}$
12. Call MNOE( $S'$ , $l$+1, $\max Gap$, $Spt_{\min}$ )

Operator $|.|$ on a sub-sequence $x$ (i.e. $|x|$ ) means don't consider the annotated time stamps on $x$. For example, $|e|$ represents an event type, $e = <|e|, time_stamp>$ represents an instance of $|e|$ appears in time series at $time_stamp$. So $|P_i.head|$ represents the prefix of episodes in current iterate, $P_i.head$ represents the instances. $P_i.body$ represents the set of all events occur after $P_i.head$ in time series. The sorting of $S = \{P_1, P_2, ..., P_n\}$ in line 1 of MNOE guarantees us getting the maxmum number of non-overlapping occurrentces of the episode. HashtableFE is a hash table got |e| as the keys, and HashtableFE(|e|) represent the set of event instances following the current prefix $|P_i.head|$. Finally,

projection( $e$ ) follows the rule: if $P_i'$ is the new projection, then $P_i'.head = P_i.head + e$ , $P_i'.body$ is a sub-set of event instances in $P_i.body$ whose time stamps are larger than $e$ .

## 3.2 Significant Frequent Episodes

### 3.2.1 EGMAMC Model

EGMAMC is a combination of memory Markov chain and aggregation Markov chain:

Suppose $X = \{x_1, x_2, ..., x_t, ..., x_T\}$ , where $t \in \{1, ..., T\}$ , $x_t \in E = \{e_1, e_2, ..., e_n\}$ is the state at current time point, $E$ represents state space. Then

$$p(x_t \mid x_{t-1}, ..., x_{t-L}) = \sum_{l=1}^{L} p(l) p^l(x_t \mid x_{t-l}) = \sum_{l=1}^{L} p(l) \sum_{k=1}^{K} p(x_t \mid k) p^l(k \mid x_{t-l}) \quad (1)$$

where $L$ is the maximum time delay permitted (i.e. the maximum time interval between every successive events permitted in an episode), the state space is categorized into $K$ clusters, $p(l)$ represents the probability that the state at current time point is determined by the state at the $l$ before time points. $p^l(k \mid x_{t-l})$ denotes the probability in which the $l$ before time points state determines the current time state belonging to the cluster $k$ . $p(x_t \mid k)$ denotes the probability the state at current time point is $x_t$ on condition that the current time point state belongs to cluster $k$ .

The states in EGMAMC's state space are categorized into episode event $K_e$ and noise $K_n$ . $K_e$ denotes the events associated with the episode $\alpha = \{S_e^1, S_e^2, ..., S_e^{N-1}, S_e^N\}$ , where $S_e^i$ represents the event appears in the $i'th$ slot of $\alpha$ and $N$ is the length of $\alpha$ . $K_n$ denotes the event not associated with $\alpha$ . If set the maximum interval between every two successive events as $L$ , we get the $\alpha$ based EGMAMC model $\Lambda_\alpha$ :

$$\begin{aligned} & P(x_t \mid x_{t-1}, \ldots, x_{t-L}) \\ &= \sum_{l=1}^{L} P(l \mid x_{t-1}, \ldots, x_{t-L}) P^l(x_t \mid x_{t-l}) \\ &= \sum_{l=1}^{L} P(l \mid x_{t-1}, \ldots, x_{t-L}) \sum_{k=1}^{K} P(x_t \mid k, x_{t-l}) P^l(k \mid x_{t-l}) \end{aligned} \quad (2)$$

When $\exists x_{t-l} \in K_e$ and $\forall l' < l, x_{t-l'} \in K_n$ , $P(l' \mid x_{t-1}, ..., x_{t-L}) = 1$ otherwise, $P(l = 1 \mid x_{t-1} \in K_n, ..., x_{t-L} \in K_n) = 1$.

Set $P(x_{t-1} \in K_n, ..., x_{t-L} \in K_n) = \lambda$ which indicates the probability that the time interval of two successive instances of episode $\alpha$ is larger than $L$ . $\lambda$ is larger means $\alpha$ distribute more sparsely in the time series, vice versus.

Set $P^l(k = K_n \mid x_{t-l}) = \eta$ as the probability that the current state belongs to noise (i.e. $1-\eta$ is the probability that the current state belongs to episode event).

When $k = K_e$ , if $x_{t-l} = S_e^{n-1}$ , then $P(x_t = S_e^n \mid K_e, x_{t-l}) = 1$ , $P(x_t \neq S_e^n \mid K_e, x_{t-l}) = 0$.

When $k = K_n$ , $P(x_t \mid K_n, x_{t-l}) = \frac{1}{M}$ , where $M$ is the number of event type which could appear in the given time series. We suppose the noise obeys an independent identical distribution.

Based on the definition above, given an observed time series $o$ , it's generated by $\Lambda_\alpha$ in the probability:

$$\begin{aligned} & P(o \mid \Lambda_\alpha) \\ &= \left(\frac{\lambda\eta}{M}\right)^{q_n^n(o)} (\lambda(1-\eta))^{q_n^e(o)} \left(\frac{(1-\lambda)\eta}{M}\right)^{q_e^n(o)} ((1-\lambda)(1-\eta))^{q_e^{e'}(o)} \\ &= \left(\frac{\lambda}{1-\lambda}\right)^{q_n^n(o)+q_n^e(o)} \left(\frac{(1-\lambda)\eta}{M}\right)^{T} \left(\frac{M(1-\eta)}{\eta}\right)^{q_e^e(o)} \\ &= \left(\frac{\lambda}{1-\lambda}\right)^{\lambda T} \left(\frac{(1-\lambda)\eta}{M}\right)^{T} \left(\frac{M(1-\eta)}{\eta}\right)^{q_e^e(o)} \\ &= \left(\frac{\lambda^{\lambda}(1-\lambda)^{(1-\lambda)}\eta}{M}\right)^{T} \left(\frac{M(1-\eta)}{\eta}\right)^{q_e^e(o)} \end{aligned} \quad (3)$$

Where $T = |o|$ denotes the length of the observed time series, $q_n^n(o)$ is the times noise states transit to noise states in $o$ . Similarly, $q_n^e(o)$ represents the times noise states transit to the first event in $\alpha$ (i.e. $S_e^1$ ). $q_e^n(o)$ denotes the times episode states transit to noise state, $q_e^{e'}(o)$ represents the times episode states transit to all the episode states "except" $S_e^1$ , $q_e^e(o)$ represents the times episode states transit to all the episode states.

We could approximately consider $q_e^e(o) = Nf_\alpha$ , so we get equation (4) from (3) as follows:

$$P(o \mid \Lambda_\alpha) = \left(\frac{\lambda^{\lambda}(1-\lambda)^{(1-\lambda)}\eta}{M}\right)^{T} \left(\frac{M(1-\eta)}{\eta}\right)^{Nf_\alpha} \quad (4)$$

Informally speaking, if $\eta$ is larger, $\alpha$ appears less frequently in $o$ , if $\lambda$ is larger, $\alpha$ distributes more sparsely in $o$ . Next section introduces the method to test whether $\Lambda_\alpha$ is proper to be utilized to describe a given time series $o$ .

### 3.2.2 Significance Testing of Frequent Episodes

This study compare EGMAMC model $\Lambda_\alpha$ with an i.i.d model by Likelihood ratio testing. If the hypothesis of $\Lambda_\alpha$ is more proper than i.i.d model to be considered as the generative model of the

observed time series $o$ is accepted, $\alpha$ is called significant episode.

We test the alternate hypothesis $H_1$: time series $o$ is generated by an EGMAMC model $\Lambda_\alpha$, against the null hypothesis $H_0$: time series $o$ is generated by an independent identical distribution model $\Lambda_{i.i.d.}$ If inequation (5) holds then $H_0$ is rejected (i.e. $\alpha$ is significant)

$$L(o) = \frac{P(o \mid H_1)}{P(o \mid H_0)} > \gamma > 0 \quad (\gamma \text{ is determined by fixing the probability of Type I error}) \tag{5}$$

According to equation (4), when $\frac{M}{M+1} < \eta < 1$ or $\frac{1}{2} < \lambda < 1$, there is too much noise in time series $o$, so that $H_1$ is equal to $H_0$. So it's only necessary to test the two hypothesis under condition $\eta < \frac{M}{M+1}$ and $\frac{1}{2} < \lambda < 1$.

Since $P(o \mid H_0) = (\frac{1}{M})^T$ is a constant, inequation (5) is equal to $P(o \mid H_1) > \gamma' = \gamma P(o \mid H_0)$, and according to equation (3), $P(o \mid H_1) > \gamma'$ is equal to $q_e^e(o) > \Gamma$, so inequation (5) is equal to

$$L'(o) = q_e^e(o) > \Gamma \quad (\gamma \text{ is determined by fixing the probability of Type I error}) \tag{6}$$

Type I error is the probability of incorrectly accepting $H_0$ under the condition that $L'(o) > \Gamma$.

$$P_{f\alpha} = P(L'(o) > \Gamma; H_0) = (\frac{1}{M})^T Q(\Gamma) \tag{7}$$

Where $Q(\Gamma) = |\{o; q_e^e(o) > \Gamma\}|$ represents the number of $T$-length time series which let $L'(o) > \Gamma$ hold.

$Q(\Gamma)$ has an upper bound as

$$Q(\Gamma) \le \sum_{k>\Gamma} C_T^k M^{\lambda T} (M-1)^{(1-\lambda)T-k} \tag{8}$$

$C_T^k$ is the combination number of choosing $k$ position in a $T$-length time series to place episode states. There are $\lambda T$ positions are filled by states which are determined by the noise at the time point before, the combination number of these states is $M^{\lambda T}$. The remaining positions in the $T$-length time series should be filled by the states are determined by episode states, and each of these position has $(M-1)$ choice of the state, because it could not be chosen as the next episode state. So the combination number is $(M-1)^{(1-\lambda)T-k}$. After all we get

$$\begin{aligned} P_{f\alpha} &\le (\frac{1}{M})^T \sum_{k>\Gamma} C_T^k M^{\lambda T} (M-1)^{(1-\lambda)T-k} \\ &< 1 - (\frac{M}{M-1})^{T\lambda_{\min}(k\le\Gamma)} \sum_{k\le\Gamma} C_T^k (1-\frac{1}{M})^{T-k} (\frac{1}{M})^k \\ &\approx 1 - (\frac{M}{M-1})^{T\lambda_{\min}(k\le\Gamma)} \Phi\left(\frac{\Gamma - \frac{T}{M}}{\sqrt{T(\frac{1}{M})(1-\frac{1}{M})}}\right) \\ &= 1 - c\Phi\left(\frac{\Gamma - \frac{T}{M}}{\sqrt{T(\frac{1}{M})(1-\frac{1}{M})}}\right) \end{aligned} \tag{9}$$

where $\Phi(.)$ is the cumulative distribution function of a standard normal random variable and the approximation holds for large $T$ due to the central limit theorem. $c = (\frac{M}{M-1})^{T\lambda_{\min}(k\le\Gamma)}$ is a constant, $\lambda_{\min}(k \le \Gamma)$ represents the minimum $\lambda$ appears in the time series in which $k \le \Gamma$. So if given the upper bound of Type I error $\varepsilon$ (e.g. $\varepsilon = 0.5$) then

$$\Gamma = \frac{T}{M} + \sqrt{(\frac{T}{M})(1-\frac{1}{M})}\,\Phi^{-1}(\frac{1-\varepsilon}{c}) \tag{10}$$

When $T$ is larger, $\Gamma \approx \frac{T}{M}$, and $q_e^e(o) = Nf_\alpha$, then according to inequation (6), when $f_\alpha > \frac{T}{MN}$, $H_0$ is rejected. Since in a time series $o$, $\eta = \frac{T - q_e^e(o)}{T} = \frac{T - Nf_\alpha}{T}$, in order to satisfy $\eta < \frac{M}{M+1}$, $f_\alpha > \frac{T}{N(M+1)}$ must hold.

Summarizing above, when $f_\alpha > \frac{T}{MN}$ and $\lambda < \frac{1}{2}$, $\alpha$ is significant episode in time series $o$.

## 3.3 Frequent Pattern Clustering for Event Detection

### *3.3.1 Frequent Pattern Spectral Analysis*

Intuitively, an event could be represented by a set of correlated frequent patterns. Instances of different types of frequent pattern have different distributions in a time series segmentation. For example, if a subsequence of objects is an episode in a time series segment, its instances must densely distribute in the segment, otherwise the distribution is sparse. If a subsequence of objects is a sequential pattern in a time series segmentation, it must periodically distribute in the original time series. It's obverse that a subsequence of objects can both be episode and sequential pattern.

Feature flows can be used to describe the distribution of frequent patterns (see Def. 7). Given a feature flow $y_f$, we can decompose it into frequency components by discrete Fourier transform (DFT):

$$X_k = \sum_{t=1}^{T} y_f(t) e^{-\frac{2\pi i}{T}(k-1)^t}, k = 1, 2, ..., T$$

$X_k$ denotes the amplitude of the sinusoid with frequency $k/T$. The dominant period of the feature flow is represented by the frequency component which has the maximum signal power. We call this maximum frequency component as dominant power spectrum, i.e. $S_f = \max \| X_k \|^2$, $k = 1, 2, ..., \lceil T/2 \rceil$, $\| X_k \|^2$ indicates the signal power at frequency $k/T$. And the corresponding dominant period is $P_f = T / \arg\max_k \| X_k \|^2$. As shown in table 1, according to the values of $S_f$ and $P_f$, we categorize frequent patterns into four categories.

**Table 1 Categories of frequent patterns and corresponding event types.**

| Pattern Categories | Feature flow description | Event Categories | Examples in offices |
|---|---|---|---|
| Burst Pattern (episode but not sequential pattern) | High $S_f$, aperiodic or long-term periodic ($P_f > \lceil T/2 \rceil$) | Burst Event: Just happen in a short time frame with large volume. | A fire alarm or an annual conference |
| Regular Pattern (episode and sequential pattern) | High $S_f$, short-term periodic ($P_f \le \lceil T/2 \rceil$) | Regular Event: Daily Happen events | Work day Meetings |
| Rare Pattern (neither episode nor sequential pattern, namely noise) | Low $S_f$, aperiodic or long-term periodic | Rare Event: Just happen in several short time frames, sometimes this kind of event is very important, sometimes it is only noise | Somebody fell down |
| Routine Pattern (not episode but sequential pattern) | Low $S_f$, short-term periodic ($P_f \le \lceil T/2 \rceil$) | Routine Event: Happen periodically but not so often comparatively with regular events | Every day cleaning work or biweekly meetings |

As defined in Table 1, different kinds of events can be represented by corresponding categories of frequent patterns, each category of frequent pattern can be interpreted as different combinations of episode and sequential pattern. We should note that the detection of rare event is a very important and challenging topic [17], but it's not proper to represent it by frequent patterns. So in this paper, we don't discriminate rare event and noise and the clustering algorithm for event detection only considered the other three categories of events.

According to the definition of sequential pattern and episode (see Def.4 and 5), if a frequent pattern is episode, it can approximately indicate its high $S_f$ feature. If a frequent pattern is sequential pattern, it can approximately indicate its short term periodic feature. So we can map categories of frequent patterns to the combinations of episode and sequential pattern. In this way, we needn't to do DFT on feature flows when we get all the frequent patterns. It promotes the computing efficiency in practice.

### *3.3.2 EDPA Algorithm*

If two patterns ($p_i$ and $p_j$) are representative of the same event, they must satisfy the following necessary conditions:

1. Their feature flows $y_{fi}$ and $y_{fj}$ must be identically distributed.
2. $p_i$ and $p_j$ must occur in the same time segments with high probability.

To measure the distribution similarity of feature flows, we introduce mutual information (MI). In information Theory, MI is usually used to measure the distance between two distributions (e.g. $D(y_{fi})$ and $D(y_{fj})$). $MI(D(y_{fi}), D(y_{fj}))$ can get a value between $[0,1]$, if $MI(D(y_{fi}), D(y_{fj})) = 1$, $D(y_{fi})$ and $D(y_{fj})$ are identical. The value of $MI(D(y_{fi}), D(y_{fj}))$ gets smaller when the distance of $D(y_{fi})$ and $D(y_{fj})$ becomes larger.

Let $TS_i$ denotes the set of time segments containing pattern $p_i$, to measure the probability of pattern $p_i$ and $p_j$ occur in the same time segment, we defined time segment occurrence overlap as follows:

$$d(f_i, f_j) = \frac{|TS_i \cap TS_j|}{\min(|TS_i|, |TS_j|)}$$

For a set of pattern feature $A_i$, we define its mutual information distance and time segment occurrence overlap as follows:

$$MI(A_i) = \min_{\forall fi, fj \in A_i} (MI(f_i, f_j))$$

$$d(A_i) = \min_{\forall fi, fj \in A_i} (d(f_i, f_j))$$

To detect events occur in time series is to find a set of pattern feature $A_i$ that minimizes the following cost function:

$$C(A_i) = \frac{1}{d(A_i) \times MI(A_i) \sum_{fj \in A_i} S_{fj}} \quad (11)$$

The underlying event $e$ associated with frequent patterns in $A_i$ can be represented as:

$$y(e) = \sum_{fj \in A_i} \frac{S_{fj}}{\sum_{fu \in A_i} S_{fu}} y_{fj} \quad (12)$$

The proposed EDPA (Event Detection by Pattern Aggregation) is a greedy algorithm. It initializes the pattern feature set $A_i = \{f_i\}$, in each iteration it merges two feature sets together if the merged result get a smaller value of the cost function, i.e. $C(A_i \bigcup A_j) < C(A_i)$ and $C(A_i \bigcup A_j) < C(A_j)$. Intuitively, frequent pattern features in the same category correlated stronger than those in different categories and it could also be verified by the cost function (equation (11)). So each time, we only input the feature flows of the frequent patterns in the same category to find a special kind of event as defined in Table 1, namely we use the feature flows of burst pattern to detect burst event, the regular pattern to detect regular event etc.

Algorithm EDPA

Input: feature flows $F = \{f_1, f_2, ..., f_n\}$ represent patterns in $P = \{p_1, p_2, ..., p_n\}$

Output: events $E = \{e_1, e_2, ..., e_m\}$

1. for every $f_i$ in $F$
2. init $A_i = \frac{1}{S_{fi}}$
3. sort $A = \{A_1, A_2, ...A_i, ..., A_n\}$ in ascendant order
4. while $A$ is not empty
5. $A_i = A.next$
6. //aggregate patterns set together until the aggregation increase $C(A_i)$
7. for every $A_j$ in $A$ where $i \neq j$
8. if $C(A_i \bigcup A_j) < C(A_i)$ and $C(A_i \bigcup A_j) < C(A_j)$
9. $A_i = A_i \bigcup A_j$ and $A = A - A_j$
10. $C_{\min} = C(A_i)$
11. else
12. output $A_i$ as $e_k$ into $E$ //represent $e_k$ as equation (12)
13. $A = A - A_i$
14. break for loop
15. end if
16. end for
17. end while
18. output $E$

## 4. RELATED WORK

S. Laxman etc. [15] defined non-overlapping episode and proposed the first algorithm to discovery non-overlapping episodes. But unfortunately, the algorithm proposed by S. Laxman etc. can only discover non-overlapping episode from the set of all episodes (including overlapping and non-overlapping ones) , which means this algorithm must based on the result output by a episode discovering algorithm such as WinMiner [3]. MNOE proposed in this paper doesn't traverse all the episode space, we only pick out our interested ones, namely non-overlapping episodes.

Meger and C. Rigotti. [3] proposes a complete algorithm (i.e. WinMiner) to find frequent episode pattern in a single long sequence. WinMiner only constrains the maximum time interval between every two successive events in episodes, so the length of sequential pattern is unbound. But WinMiner does not consider the time interval and position of the objects in the pattern instances. J. Pei and J. Han [2] presented the first algorithm discovering sequential patterns.

Giannella et al. [5] proposed a frequent itemsets mining algorithm over data stream which utilizes tilted windows to calculate the frequent patterns for the most recent transactions and maintains a tree data structure (i.e. FP-stream) to represent the frequent itemsets.

Manku and Motwani [7] proposed an approximate frequency counts in data streams. Cormode and Muthukrishnan [4] proposed an algorithm to find the hottest k items by group testing. The algorithm is used with the turnstile data stream model which allows addition as well as deletion of data items.

Gaber et al. [6] proposed an AOG-based algorithm: Lightweight frequency counting LWF. It can find an approximate solution to the most frequent items on-line using adaptation and releasing the least frequent items regularly in order to count the more frequent ones.

There is a series of research projects worked on event detection and correlation of events especially in the database community. TelegraphCQ [21] focuses on meeting the challenges that in handling large streams of continuous queries over high-volume, highly-variable data streams. Aurora [22, 23] manages continuous data stream for monitoring applications; it focuses on DBMS-like support for applications. Borealis [24] provide advanced capabilities that are commonly required by newly-emerging stream processing applications. The STREAM project [25] views stream processing as the running of continuous queries expressed in a query language (CQL) that includes sliding windows and sampling over the data stream. Unfortunately, the mentioned projects do not consider the frequent patterns as an important structure in data streams. In this paper, authors argue that frequent patterns form events in data stream and different types of frequent patterns have different features which also represent different features of events. EDPA presented in this paper cluster frequent patterns into events based on the feature follow of frequent patterns.

## 5. EXPERIMENTAL EVALUATION

We evaluated the presented framework on two public datasets (MERL and IL dataset) and a dataset come from a smart building project (Smart-Building dataset). All the datasets are generated by

sensor networks. We use relative minimum supports in our experiments and represent them as ($\min Spt_S$, $\min Spt_E$), $\min Spt_S, \min Spt_E \in (0,1)$, where $\min Spt_S$ is the relative minimum support for sequential pattern and $\min Spt_E$ is the relative minimum support for episode.

We compared efficiency performance of the proposed MNOE algorithm with NOC algorithm [15] on all the three datasets. Because S. Laxman etc. [15] declare any episode discovering algorithm could be used at the step of generating all the episodes (non-overlapping and overlapping episode) in NOC. So we implemented the most efficient episode discovering algorithm WinMiner [3] proposed in existing literatures and generated all episodes for NOC. We implemented PrefixSpan [2] to discover sequential patterns.

We evaluated the proposed EDPA algorithm on MERL dataset and Smart-Building dataset, because these two datasets include event calendars of the spots where the sensor networks were deployed. We can measure the performance of EDPA by accuracy. For an output event of EDPA, if its duration is overlapped with an event recorded in the event calendar, we consider this event is correctly detected. The accuracy of EDPA is the percentage of correctly detected events in all the events output by EDPA. All the frequent patterns (both sequential pattern and episode) input into EDPA are maximal and we evaluated EDPA when the input episodes are overlapping or non-overlapping and are significant or non-significant respectively (we don't distinguish overlapping sequential patterns between non-overlapping ones or significant sequential patterns between non-significant ones).

All the algorithms are implemented in Java, and run on a window server platform with CPU 3.5G, Memory 2G.

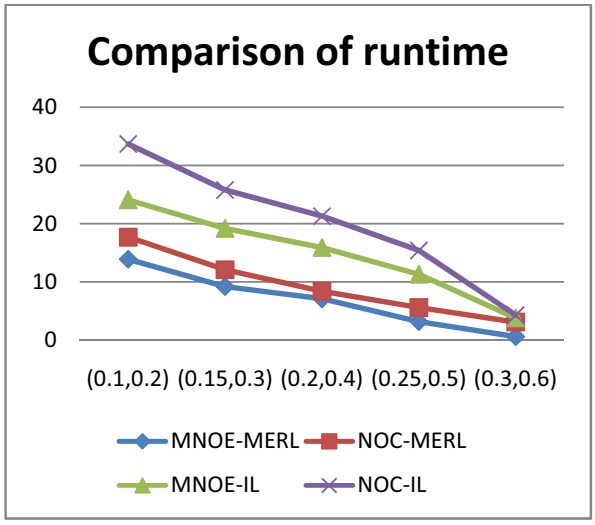
## 5.1 Experiments on MERL and IL Datasets

MERL Motion Detector Dataset [18] [1](using "MERL dataset" for short) has been collected by Mitsubishi Electric Research Labs (MERL) from a network of over 200 sensors for two years between March 2006 and March 2008. This sensor network was deployed in an office building and each sensor only records the motion signals caused by individuals. We picked a subset of the data which records the data generated by 4 sensors in a meet room, namely "nitta" and got a 4 dimension time series last from 2006-03-23 03:10 to 2007-07-04 06:18. We aggregated the raw data into 25229 records by adding the values of records in the same minutes together and took the reservation records of this meeting room provided in MERL dataset as the ground true of events happened in this room.

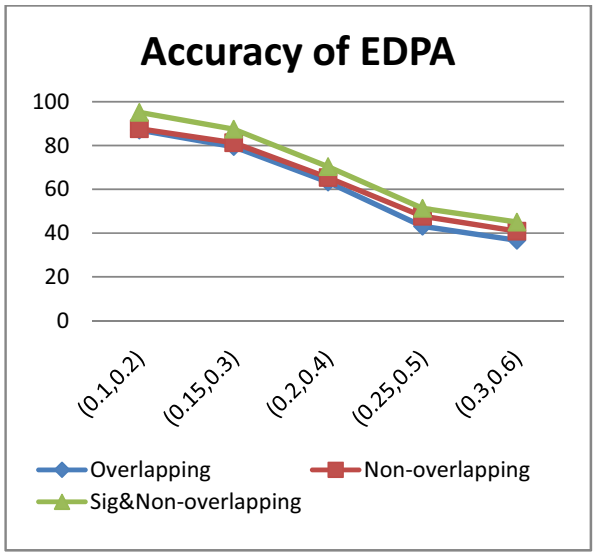
Intel Lab Dataset [19][2](using "IL dataset" for short) has been collected from 54 sensors from February 28th to April 5th , 2004. We selected the data generated by sensors from No. 1 to No. 10, which forms a 40 dimensional time series (every record fired by a sensor includes 4 values) and discretized the time series by algorithm proposed in Ref.[20]. As illustrated in Fig.3 (a), MNOE beats NOC on runtime. The proposed MNOE is more efficient than NOC under any minimum support set. Fig.3 (b) shows that EDPA achieves better accuracy when maximal sequential patterns and maximal significant non-overlapping episodes are input.

[1] http://www.merl.com/wmd/

[2] db.csail.mit.edu/labdata/labdata.html




**(a)x-axis represents different minimum support sets, y-axis represents the runtime of algorithms**




**(b) x-axis represents different minimum support sets, y-axis represents accuracy of EDPA.**

**Fig.3 (a) Compare the runtime of MNOE and NOC under different minimum support sets on MERL and IL datasets. (b) accuracy of EDPA playing on MERL dataset under different minimum support sets and different input episodes: overlapping, non-overlapping, significant and non-overlapping.**

## 5.2 Experiments on Smart-Building Datasets

We also apply the proposed algorithms in a sensor network data analyzing system, namely EventMiner. Smart-Building dataset has been collected from a sensor network in a smart building. The sensor network we studied consists of six sensors and is deployed

in a meeting room. The sensors could throw out events simultaneously. The event types include temperature, motion, illumination, acoustic, humidity and carbon dioxide. The finest time granularity is minute. As listed in table 2, we split five month's data into three data sets. The event calendars in these datasets are the reservation records of this meeting room. For space limitation, we plot average runtimes of MNOE and NOC under different minimum support sets in Fig. 4 and it shows that MNOE also beats NOC. As the results are similar, we only plot the results of EDPA playing on Dataset1 in Fig.5. The maximal significant non-overlapping episodes also improve the accuracy of EDPA.

**Table 2. Smart-Building Datasets**

| Data sets | Number of records | Month |
|---|---|---|
| Dataset1 | 43359 | 6 |
| Dataset2 | 89125 | 7-8 |
| Dataset3 | 116993 | 9-11 |

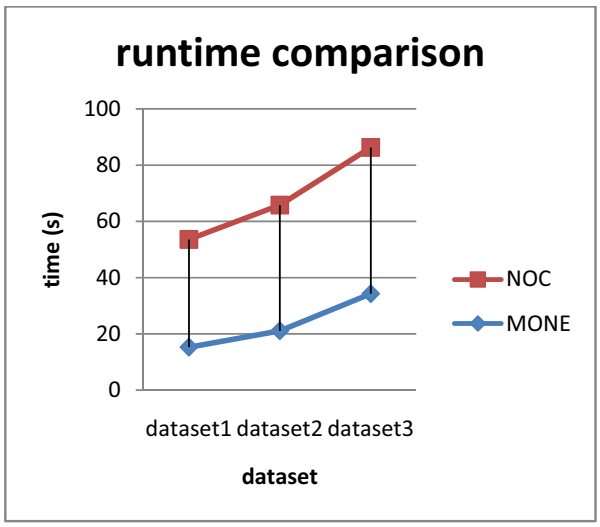



**Fig. 4 Compare the runtime of MNOE and NOC.**

Fig. 4 illustrates the average runtimes of the two algorithms under different minimum support sets (i.e. (0.16,0.2), (0.17,0.3), (0.18,0.4), (0.19,0.5), (0.2,0.6)). The result shows our MNOE is more efficient than NOC and scalable to large datasets. It demonstrate that an algorithm directly discover non-overlapping episode is necessary.

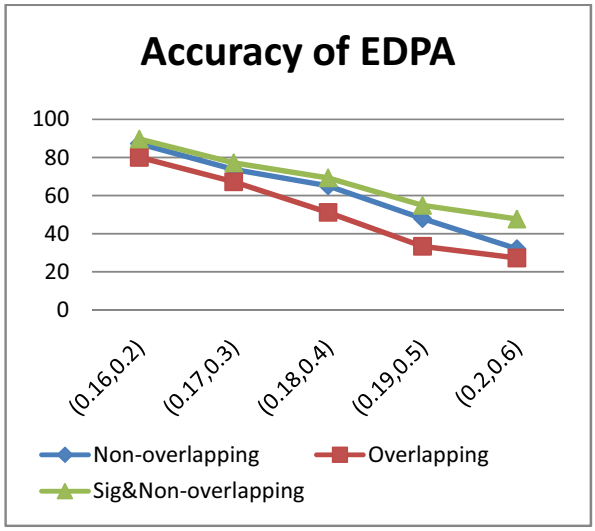



**Fig. 5 accuracy of EDPA under different minimum support sets in Dataset1. X-axis represents different minimum support sets, y-axis represents accuracy of EDPA.**

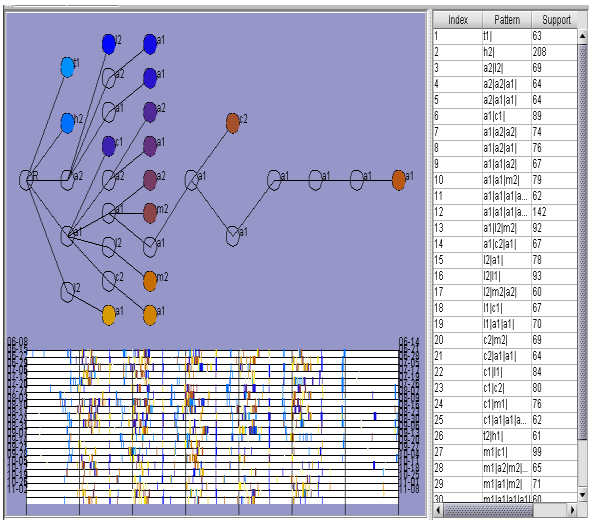



**Fig.6 a screen shot of EventMiner illustrates discovered frequent patterns and detected events**

Fig.6 shows the visualizing interface of frequent patterns and events. In the right panel, a table list all the discovered frequent patterns and their supports. In the left top panel, the frequent patterns are organized in a tree which help users to find out outlier or novel patterns intuitively. In the left bottom panel, the events formed by the frequent patterns are plot on the time line.

## 6. CONCLUSIONS

This paper proposed a frequent pattern based event detection framework. In the phase of frequent pattern discovery, non-overlapping episode is discovered by the proposed MNOE

algorithm. Every non-overlapping episode constructs a corresponding EGMAMC model. In the frequent pattern selection phase, we utilize likelihood ratio testing (set Type I error as 0.5) to measure whether EGMAMC is more proper than i.i.d. model to describe a given time series. If an EGMAMC model is accepted in the likelihood ratio testing, the corresponding episode is said to be the significant episode in the given time series. Finally in the modeling phase, we defined feature flow to represent frequent pattern instances in time series and presented EDPA clustering algorithm which can aggregate input frequent patterns into events. Experiments on sensor network datasets demonstrate that MNOE outperforms the existing algorithm and selecting maximal significant non-overlapping episode as features improves the performance of EDPA.

## 7. REFERENCES


[1] Li Wan, Jianxin Liao, Xiaomin Zhu, "Mining Sequential Episodes from Segmentation of Multivariate Time Series", the proceedings of DMIN 2009 (to be appear).

[2] J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," Proc. 17th Int'l Conf. Data Eng., pp. 215-224, 2001.

[3] N. Meger and C. Rigotti. Constraint-based mining of episode rules and optimal window sizes. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04), pages 313–324. Springer, 2004.

[4] G. Cormode, S. Muthukrishnan What's hot and what's not: tracking most frequent items dynamically. PODS 2003: 296-306

[5] C. Giannella, J. Han, J. Pei, X. Yan, and P.S. Yu, Mining Frequent Patterns in Data Streams at Multiple Time Granularities, in H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), Next Generation Data Mining, AAAI/MIT, 2003.

[6] Gaber, M, M., Krishnaswamy, S., and Zaslavsky, A., On-board Mining of Data Streams in Sensor Networks, Accepted as a chapter in the forthcoming book Advanced Methods of Knowledge Discovery from Complex Data, (Eds.) Sanghamitra Badhyopadhyay, Ujjwal Maulik, Lawrence Holder and Diane Cook.

[7] G. S. Manku and R. Motwani. Approximate frequency counts over data streams. In Proceedings of the 28th International Conference on Very Large Data Bases, Hong Kong, China, August 2002.

[8] Joao Gama, Pedro Pereira Rodrigues, Raquel Sebastiao, Evaluating Algorithms that Learn from Data Streams. The proceeding of SAC'09, 2009

[9] Eduardo J. Spinosa, Joao Gama, Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks, the proceeding of SAC'08, 2008.

[10] Yixin Chen, Li Tu, Density-Based Clustering for Real-Time Stream Data, the proceedings of kdd'07, 2007.

[11] Joao Gama, Mohamed Medhat Gaber, Learning from data streams: processing techniques in sensor networks. Springer 2007.

[12] Auroop R. Ganguly , Joao Gama, Olufemi A. Omitaomu, Mohamed Medhat Gaber, Ranga Raju Vatsavai (Editor), Knowledge Discovery from Sensor Data (Industrial Innovation) , CRC, 2008

[13] Mohamed Medhat Gaber, Arkady Zaslavsky, Shonali Krishnaswamy, Mining Data Streams: A Review, SIGMOD Record, Vol. 34, No. 2, Page:18-26, 2005

[14] Han J, Kamber M 2001 Data mining: Concepts and techniques (San Fransisco, CA: Morgan Kauffmann)

[15] S. Laxman, P. S. Sastry, and K. P. Unnikrishnan. Discovering frequent episodes and learning Hidden Markov Models: A formal connection. IEEE Transactions on Knowledge and Data Engineering, 17(11):1505–1517, Nov. 2005.

[16] S. Laxman, Stream Prediction Using A Generative Model Based On Frequent Episodes In Event Sequences Proceeding of KDD 2008

[17] G. M. Weiss and H. Hirsh. Learning to predict rare events in event sequences. In proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD 98), pages 359–363, New York City, NY, USA, Aug. 27–31 1998.

[18] Wren, C.R.; Ivanov, Y.A.; Leigh, D.; Westhues, J., "The MERL Motion Detector Dataset", Workshop on Massive Datasets (MD), ISBN: 978-1-50593-981-8, pp. 10-14, November 2007

[19] db.csail.mit.edu/labdata/labdata.html.

[20] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Bill Chiu. A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, In Proceeding of DMKD, 2003

[21] Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, etc TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. Proceedings of CIDR 2003 Conference

[22] D. Abadi, D. Carney, U. Cetintemel, etc. Aurora: A New Model and Architecture for Data Stream Management. In VLDB Journal (12)2: 120-139, 2003

[23] D. Abadi, D. Carney, U. Cetintemel, etc. Aurora: A Data Stream Management System (Demonstration). In proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'03), San Diego, CA, June 2003.

[24] Daniel J. Abadi, Yanif Ahmad, Magdalena Balazinska, etc, The Design of the Borealis Stream Processing Engine, 2nd Biennial Conference on Innovative Data Systems Research (CIDR'05), Asilomar, CA, January 2005.

[25] R. Motwani, J. Widom, A. Arasu, etc. Query Processing, Resource Management, and Approximation in a Data Stream Management System. In Proc. of CIDR 2003, Jan. 2003

# Supervised Clustering via Principal Component Analysis in a Retrieval Application *

Esteban Garcia-Cuesta
Physics Department
Av. Universidad 30, 28911
Leganes (Madrid), Spain
esteban.garcia@uc3m.es

Ines M. Galvan
Computer Science Department
Av. Universidad 30, 28911
Leganes (Madrid), Spain
igalvan@inf.uc3m.es

Antonio J. de Castro
Physics Department
Av. Universidad 30, 28911
Leganes (Madrid), Spain
decastro@fis.uc3m.es

## ABSTRACT

In regression problems where the number of predictors exceeds the number of observations and the correlation between the predictors is high, a dimensionality reduction or a variable selection approach is demanded. In this paper we deal with a real application where we want to retrieve the physical characteristics of a combustion process from the measurements obtained with a spectroscopic sensor. This application shows up a multicollinearity problem but furthermore it is considered an ill-posed problem.

Guided by this application scenario, we propose a clustering approach to find out homogeneous subsets of data which are embedded in arbitrary oriented linear manifold. This model is developed under certain assumptions guided by a priori problem knowledge. The resulting division preserves both, the priori assumptions and the homogeneity in the models. Thereby we break the whole problem in n subproblems improving its individual prediction accuracy versus a global solution. We show the obtained improvements in a real application scenario related with estimating the temperature from spectroscopic data in a remote sensing framework.

## General Terms

Dimensionality reduction, supervised learning, clustering, regression, remote sensing, principal components analysis

*Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. *SensorKDD'09*, June 28, 2009, Paris, France. Copyright 2009 ACM 978-1-60558-668-7...$5.00.

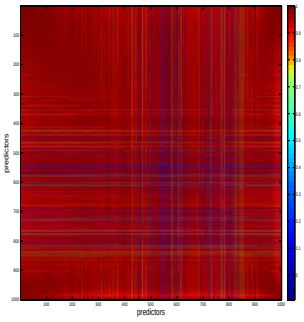



Figure 1: Predictors correlation matrix.

## 1. INTRODUCTION

In this paper we study a method to transform a regression problem in a classification + n regression problems throughout a supervised clustering technique that makes used of the data principal components.

The scenario that has motivated our work concerns with a method for the retrieval of physical properties in a combustion process, and specifically with the estimation of the temperature which appears among others as a very important parameter to be monitored [14, 15, 19, 21]. The physics of this process is governed by the Radiative Transfer Equation (RTE)[8], and its a challenging problem due to its globally non-linear behavior and its high dimensionality data i.e. spectral data.

Furthermore the predictors or input variables are highly correlated to each other which could lead to a degradation of the prediction accuracy. That degradation is due to an overkill adjustment of the model parameters leading to overfitting, and reducing the prediction performance on new data sets. This phenomena is called multicollinearity, and its influence depends on the data and the methodology applied. Regarding the data influence, we want to note that the predictors redundancy it is not as important as its non homogeneity. In [2] a comparative study of two popular methods (neural networks NN and multivariate adaptive regression spline MARS), is done when multicollinearity is present. The study concludes that NN outperforms MARS facing multicollinearity problems. Also in [4], the authors proposed a truncated

total least squares method discarding the smallest singular values to face the computational problem due to singular values that are close to zero.

An important knowledge related with the RTE equation is that different variables of the spectral data provides different information on the physical properties. It is also known that every predictor contains information related with all responses or outcome variables, although there are physical reasons to acknowledge the fact that the relationship should be stronger between the above mentioned predictors-responses. Due to the fact that the most energetic predictors are more related with every response that any other, it is not a trivial process to uncover the physical relationships, specially because sometimes these relationships can involve many variables and the already exposed global non-linearity.

The figure 1 illustrates the correlation matrix of our predictors data. The red color represents predictors which are highly correlated. Most of the matrix is highly correlated and it suggests that exists multicollinearity in the data.

A general approach to solve these problems is to consider its principal components (PC) as predictors instead than the original variables [12]. These new variables are the orthogonal linear combination of the original predictors in decreasing order of variance.

The extension of principal component analysis (PCA) to a regression problem is named principal component regression (PCR) which performs a regression using the principal components instead of the raw predictors. Although this intuitive and appealing approach is optimal from a preserving energy point of view, it does not necessarily provide a meaningful representation of the data; i.e. it does not correspond to physical quantities.

This approach has been studied in previous works [6, 7] without getting the desired accuracy and robustness. This lack of success can be assign to the next two reasons. First, we are dealing with a non-linear problem and therefore it might not find out the truly relationships between variables since we are using a linear method. The second reason is related with the fact that we are trying to reduce the whole set of predictors at once. This implies that the PCA makes an optimal reduction (from the least square error point of view) without checking if the variation is due to a physical parameter or another, and therefore mixing the information.

In [7] an alternative approach to solve this problem was proposed. The proposal is based on a feature selection of the original predictors using the eigenvectors coefficients, and applying a maximum local peak selector to force a spreading predictors selection. Despite of the results improvements, the mean variance error was still high.

Although an optimal variable selection approach is desirable in our context application, it can be very expensive. Even knowing the number of desired features $t$ from a set of $u$ there are $q = \binom{u}{t} = \frac{u!}{(u-t)!t!}$ subsets which is impractical for high dimensionality data. There are other methods computationally more practical but they lack of optimality or assume monotonicity as the branch-and-bound feature selection algorithm [5].

Thereby due to the multicollinearity problems and computationally cost , we address the problem from a clustering point of view. We propose to use a supervised clustering method based on principal components analysis to find out linear and homogeneous data subsets, desiring that these new data subsets will be easier to estimate. We define homogeneity as energy or variance preserving and we suppose that it exists in some locality. With this approach we pursue two main goals: find out a lower dimensional linear manifold where the predictors and the responses lay on, and decouple (at least in some degree) the non-linear relation between them.

Our proposal algorithm can be regarded as a two-step algorithm. The first step finds a linear manifold that is going to extract the common information (correlated) between the predictors and the responses, while the second step is going to find out the clusters of the projected data in that manifold obtaining the data with a common behavior.

The paper is organized as follows. Section 2 introduces the general regression framework and the underlying latent model. In Section 3, the algorithm is described and explained, including the density clustering process. Section 4 introduces the combustion temperature estimation problem and we provide with the results obtained. We conclude with a discussion of the method and the results in Section 5.

# 2. REGRESSION MODELS

## 2.1 General Framework

The application that has motivated our work is a multivariate prediction in a regression setting problem. We assume there is a database $\mathbf{D}$ of $d$-dimensional feature vectors ($\mathbf{D} \in \Re^{d \times n}$) with $n$ samples, and each one of the samples $\mathbf{d}_i \in \mathbf{D}$ is composed by a set of predictors $\mathbf{x}_i \in \Re^{p \times 1}$, and a set of responses denoted by $\mathbf{y}_i \in \Re^{s \times 1}$, where $p + s = d$ (see notation[1]).

Then we can consider the following form:

$$\mathbf{Y} = \mathbf{B}\mathbf{X} + \mathbf{E} \tag{1}$$

being $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_s]^T$, $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_p]^T$, and $\mathbf{B} = [\mathbf{b}_1 \dots \mathbf{b}_p]$ is the $s \times p$ matrix of coefficients, and $\mathbf{E}$ is the $s \times n$ error matrix.

An optimal $\mathbf{B}$ which minimizes $\| \mathbf{Y} - \mathbf{B}\mathbf{X} \|_F$ is given by

$$\mathbf{B} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} \tag{2}$$

and the prediction $\hat{\mathbf{y}}_0$ over a new element $\mathbf{x}_0$ is obtained by

[1]Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the $j^{th}$ column of the matrix $\mathbf{D}$. $d_{ij}$ denotes the scalar in the row $i$ and column $j$ of the matrix $\mathbf{D}$ and the scalar $i$-th element of a column vector $\mathbf{d}_j$. All non-bold letters represent scalar variables. $||\mathbf{x}||_2 = \sqrt{\mathbf{x}^T\mathbf{x}}$ designates Euclidean norm of $\mathbf{x}$. $||\mathbf{A}||_F = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of a matrix.

$$\hat{\mathbf{y}}_0 = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{x}_0 \tag{3}$$

When the operation $(\mathbf{X}\mathbf{X}^T)^{-1}$ in not possible because of its singularity due to the multicollinearity, then a regularization can be applied. The methods ridged regression or the more recent lasso, make used of the regularization penalizing the large regression coefficients. They try to minimize the following expression,

$$\min_{\mathbf{B}} ||\mathbf{Y} - \mathbf{B}\mathbf{X}||_F + \lambda||\mathbf{B}||_2^2 \tag{4}$$

where the second term shrinks the coefficients towards zero. The regularization parameter $\lambda$ controls the amount of shrinkage, and allows to solve $(\mathbf{X}\mathbf{X}^T)^{-1}$ when it is singular. Then, the estimation is obtained by $\hat{\mathbf{Y}} = \overline{\mathbf{Y}} + \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{X}$ - see [11, 17, 10] and the references therein-. These approaches have been tested and despite its generalization improvements, the obtained overall accuracy is not acceptable.

Another alternative to avoid the multicollinearity is to reduce the data in order to making the new predictors independent each one from each other. It does not leave out any variable, but uses a weighted sum off all of them to make this reduction. The weights are allocated according to its contribution to the variance of the variables.

Let

$$\mathbf{P} = \mathbf{\Gamma}^T\mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_k]^T. \tag{5}$$

where $\mathbf{P}$ is a $k \times n$ matrix with $k \leq p$, $\mathbf{\Gamma}$ is a $p \times p$ matrix with the first $k$ eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ on its first $k$ columns, and $p_{ki}$ is the projection of the $i^{th}$ observation into the first $k$ bases of $\mathbf{\Gamma}$.

Therefore in PCR setting the predictor $\hat{\mathbf{y}}_i$ is given by

$$\begin{aligned}\hat{\mathbf{y}}_i &= \mathbf{Y}\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{p}_i = \\ &= \mathbf{Y}\mathbf{X}^T\mathbf{\Gamma}(\mathbf{\Gamma}^T\mathbf{X}\mathbf{X}^T\mathbf{\Gamma})^{-1}\mathbf{\Gamma}^T\mathbf{x}_i.\end{aligned} \tag{6}$$

The regression estimation can be expressed in parametric from by: $\hat{y}_{ij} = b_{i0} + b_{i1}p_{1j} + \ldots + b_{jk}p_{kj} + \epsilon$, where $p_{kj}$ is the $k^{th}$ principal component of the $j^{th}$ sample. Then, it is hoped that the projections will still retain good predictive behavior, and that the $\mathbf{p}_i$ will be interpretable. This approach is desired whenever we are obtaining a weighted combination of a subset of variables which are correlated and have common information. Then, it is not desirable to drop any predictor out and using principal component analysis would be helpful. In our case that is not very desirable because a few predictors are going to be used to estimate most of the responses reducing the flexibility of the model facing new observations, leading to overfitting and bias towards a subset of the data set.

In the next subsection we introduce the underlying model base on latent models that allows to find the correlated clusters in the data.

## 2.2 Underlying model

Different techniques can be considered to relate two sets of variables. In regression problems, one of these approaches is to calculate the PCs of the two groups, followed by the calculation of the regression equation.

The algorithm can be summarized as:

1. Compute the principal components of the predictors and responses matrix variables.

2. Compute the projection $\mathbf{P} = [\mathbf{p}_1 \dots \mathbf{p}_k]^T$ of each data set using the first $k$ eigenvectors $[\gamma_1 \dots \gamma_k]^T$.

3. Calculate the regression equation to predict each $\mathbf{Px}$ from one set from the other $\mathbf{Py}$ in the other set.

For our purpose of finding clusters of linearly homogeneous data, the algorithm fails because two input and output subspaces associated to different local homogeneity (for instance the first input eigenbase and the second output eigenbase) created by the PCA analysis can be related and they do not from an orthogonal base which is necessary to preserve homogeneity.

To overcome this, we consider the following model to support the supervised clustering via principal component method. Suppose that we have a response variable $\mathbf{Y}$ which is related to an underlying latent model $\mathbf{Hy} \in \Re^{s\times k}$ being $k \ll s$ by a linear relation where,

$$\mathbf{Y} = \mathbf{BHy} + \mathbf{E} \tag{7}$$

and in addition, we have a measurement on a set of spectral data $\mathbf{X}$ related to the underlying model $\mathbf{Hx} \in \Re^{p\times l}$ being $l \ll p$,

$$\mathbf{X} = \mathbf{AHx} + \mathbf{F} \tag{8}$$

where $\mathbf{Hx_i} \perp \mathbf{Hy_j}\ \forall\ i, j : i \neq j$.

The underlying models $\mathbf{Hy}$ and $\mathbf{Hx}$ are reduced subspaces of the original data variables which preserves their information from the least squares error reconstruction point of view.

Therefore we are seeking for those elements which share common linear subspaces which corresponds with lines, planes or hyperplanes formed by the projection into the models $\mathbf{Hy}$ and $\mathbf{Hx}$. Thus, $\mathbf{Y}$ can be predicted by $\mathbf{Hy}$, $\mathbf{Hy}$ by $\mathbf{Hx}$, and $\mathbf{Hx}$ by $\mathbf{X}$.

For a given number of predictors, the latent structure in (7) and (8) is an special structure case of an error-in-variables model, where $\mathbf{H} = [\mathbf{Hx} + \mathbf{Hy}]$ and it is subject to $\| \mathbf{H} \| = 1$.

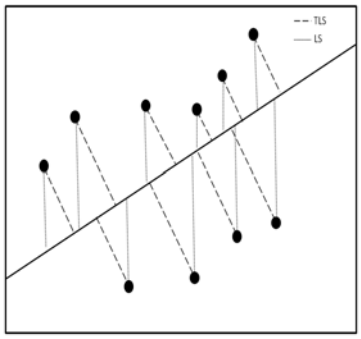



Figure 2: Lines of Total Least Squares (TLS) vs. Least Squares(LS) error.

We could set up an optimization criterion as follows

$$\min_{\mathbf{B},\mathbf{A},\mathbf{H},\|\mathbf{H}\|=1} \frac{\sum_{i=1}^{n}(\mathbf{y}_i - \mathbf{B}\mathbf{H}\mathbf{y}_i)^2}{\sigma_y^2} + \frac{\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{A}\mathbf{H}\mathbf{x}_i)^2}{\sigma_X^2} \tag{9}$$

where we try to minimize the distance between the outcome and the latent model $\mathbf{Hy}$ and simultaneously minimizing the distance from predictors to the latent model $\mathbf{Hx}$. This optimization problem amounts with finding the linear subspace that relates the two underlying models, preserving the information of both. and can be solve as an eigen-problem(see appendix A).

There is also a geometric interpretation to this underlying model, as locating the line, plane or hyperplane nearest to a collection of points where the distance is measured not only in the "y" axis but using the true euclidean distance. This is refereed in literature as total least squares error (TLS). The figure 2 shows the difference between the typical least squares error (LS) where the error is measure along the y-axis, and TLS that calculates the true euclidean distance between the point and the line. This problem can be solved using a singular value decomposition (SVD) on the augmented matrix $\mathbf{C}(x, y)$. A further description of the method and its solution can be find in [9], and [16].

A similar geometric interpretation but constrained to the problem of characterizing affine subspaces in $\Re^p$ passing through specified affine subspaces of lower dimension, is developed as a constrained principal component analysis in [18]. It provides an interesting related parametric as well as Cartesian systems of equations formulation.

We want to point out that the solution of (9) can also be explain as maximizing the next objective function,

$$tr\{\mathbf{XHxHx}^T\mathbf{X}^T\} + tr\{\mathbf{YHyHy}^T\mathbf{Y}^T\} \tag{10}$$

hence the set of variables with higher variance will be more influential. To control that influence, we propose a weighted version of the solution adding the parameter $\alpha$ which range is $0 \leq \alpha \leq 1$. The resulting objective function is:

$$(1-\alpha)tr\{\mathbf{XHxHx}^T\mathbf{X}^T\} + (\alpha)tr\{\mathbf{YHyHy}^T\mathbf{Y}^T\}. \tag{11}$$

This can be solved by the augmented matrix $\mathbf{Z} = (\mathbf{X}, \mathbf{Y})$ as we did before (see appendix A), but multiplying the predictors $\mathbf{X}$ by $(1-\alpha)$, and the responses $\mathbf{Y}$ by $\alpha$ [2].

In [1] Bair et. al. mention a similar idea as a natural supervised modification that encourages the leading eigenvector to align with $\mathbf{Y}$. They do not pursue the idea further, because the estimate requires of the responses as well as the predictors and it can not be used directly with the test data.

In the next, we introduce the algorithm used to find the different linear models contained in the data, and its description.

## 3. SUBSPACE CLUSTERING ALGORITHM PROPOSAL

Firstly we want to introduce the physical motivation that has guided this work. In the Section 1, we introduced the multicollinearity problem and how different techniques try to solve it, having different advantages and drawbacks. We propose a supervised clustering approach to split the problem which not only helps to reduce the multicollinearity problems in a regression framework, but furthermore it fits a priori application knowledge.

In our remote sensing application problem we can detect two types of samples upon their behaviors. We can affirm that:

1. in general, whenever there is a big variation in the predictors (i.e.spectral data) it also imply a big variation in the responses (i.e. temperature profiles), but
2. in some cases, a small variation of the predictors can also imply big variation in the responses.

In both situations, every predictor is affected by the variations, and both reinforce the multicollinearity. However, considering those affirmations we can assume that locally there exists a group of situations that behaves as explained in the first item (almost linearly), but further away of this locality the behavior shifts to be a non-linear one as explained at item two.

The proposed algorithm tries to exploit this problem knowledge associated to physical properties, and find out different models that have homogeneous local behavior. Therefore, we look for those samples whose behavior follow the above mentioned problem knowledge, and we separate them into different models according to its common behavior from a linear point of view.

Whether such locality exists, then the samples that belongs to that locality will build up a cluster once the original data is projected into the linear embedding subspace obtained by

[2]Without any lost of generality, we need to notice that the $\alpha$ parameter is squared during the optimization problem (see appendix A) and then the weighting process will be squared.

**Algorithm 1**

1: Input:
$\mathbf{X} \in \Re^{nxs}$ % It ts centered and scaled
$\mathbf{Y} \in \Re^{nxp}$ % It is centered and scaled
$\rho :=$ density clustering algorithm parameters
$\alpha := 0$
$step_\alpha := 0.1$
2: Output:
Set of clusters indexes $idx_clusters$
3: **for** $\alpha := 0 : step_\alpha : 1$ **do**
4: $\mathbf{X} := (1-\alpha)\mathbf{X}$
5: $\mathbf{Y} := (\alpha)\mathbf{Y}$
6: $\mathbf{Z} := [\mathbf{X}|\mathbf{Y}]$
7: $r := rank(\mathbf{Z})$
8: $[\mathbf{U}\ \mathbf{S}\ \mathbf{V}\ ] := \mathrm{SVD}(\mathbf{Z})$
9: $\mathbf{P} := \mathbf{U}^T\mathbf{Z}$
10: **for** $i := 1 : 1 : r$ **do**
11: **if** (density_clustering_algorithm($\mathbf{P}_i$, $\rho$) $\geq 2$ ) **then**
12: return idx_clusters
13: **end if**
14: $i := i + 1$
15: **end for**
16: $\alpha := \alpha + step_\alpha$
17: **end for**

solving (9). That model tries to exploit the first affirmation from above capturing the maximum redundancy between the predictors and responses, reducing simultaneously the residuals $\mathbf{E}$ and $\mathbf{F}$ from (7), and(8). Then, we can discover the different clusters iteratively assuming locality behavior.

The proposal algorithm is described in algorithm 1. It is composed by two main for loops which use a greedy approach to adjust the parameters $\alpha$ and $r$. At steps 4, 5 and 6, we composed the augmented matrix $\mathbf{Z}$ and we weight it by $\alpha$. Initially, the value of $\alpha$ is set to 0 and therefore $\mathbf{Z} = [\mathbf{X}|\mathbf{0}]$ which corresponds with an unsupervised mode.

Afterwards, we solve (9), and we transform the raw predictors into projected predictors $\mathbf{P}$ using the eigenvectors $\mathbf{U}$.

To calculate the eigenvectors of the augmented matrix $\mathbf{Z}$ we use the singular value decomposition (SVD) (algorithm step 8). Then the matrix $\mathbf{Z}$ can be decomposed as: $\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, where the columns of $\mathbf{U} \in \Re^{d\times d}$ give the eigenvectors of $\mathbf{Z}\mathbf{Z}^T$, the columns of $\mathbf{V} \in \Re^{n\times n}$ give the eigenvectors of $\mathbf{Z}^T\mathbf{Z}$. Being $r$ the rank of $\mathbf{Z}$, the linear embedding is obtained by its $\lambda \leq r$ first eigenvectors which corresponds with a $\lambda$ dimensional space related with the amount of linearity between $\mathbf{X}$ and $\mathbf{Y}$.

Then, we apply a density clustering algorithm to detect the homogeneous data in the projected space. The density clustering algorithm discovers arbitrary shaped clusters based on the density of the data, therefore finding the different models in the projected space.

Afterwards, if the algorithm does not find more than one cluster during the first main loop, the $\alpha$ parameter described in (11), is modified to weight the $\mathbf{X}$ and the $\mathbf{Y}$ encouraging the leading eigenvector to align with $\mathbf{Y}$. We repeat the process until we find a division.

During the execution, $\alpha$ is going to range from 0 to 1, or in other words from unsupervised clustering or equivalently PCA clustering, to a fully supervised clustering.

We want to point out that we are weighting uniformly both predictors and responses, and a more specific weighting function could be applied if we desire to introduce a priori information related with blocks of data.

Once the algorithm has found the different clusters we apply a learning algorithm to obtain the estimations of the new models. We use a wrapper approach [13] to analyze these results, and if one of the models does not have successful results we apply again the algorithm to its data and so on, until we find successful results or we can not find any other cluster, meaning that there is not any other linear relation.

# 4. APPLICATION TO A COMBUSTION TEMPERATURE ESTIMATION SCENARIO

Here we are going to apply the above explained algorithm to the combustion scenario which is related with the inversion of the radiative transfer equation (RTE) [8] which is defined as,

$$L_i = \int_{z_0}^{z_1} B_i\{T(z), C(z)\}K_i(z)dz \qquad (12)$$
$$K_i = \frac{d\tau_i}{dz}$$

where the parameter $L_i$ expresses the amount of energy emitted by each wavenumber $i$, $B_i$ is Planck's law (standard black body emission), $\tau_i$ is the transmittance, $T(z)$ is the temperature profile, $C(z)$ is the gas concentration profile, and $K_i$ is the so-called temperature weighting function that gives the relative contribution of the radiance coming from each region $dz$. The forward problem is to obtain $L_i$ knowing the physical parameters. But we would like to invert this equation, that is: from the radiance measure $L_i$ provided by the spectrometer, we would like to know the temperature profile. Given the measurements of energy represented by $\mathbf{x}_i \in \Re^{p\times 1}$, we would like to predict the flame constituents such as temperature, and length, represented by $\mathbf{y}_i \in \Re^{s\times 1}$. We would like to learn a mapping $f$ such that $\mathbf{Y} = f(\mathbf{X})$, where $\mathbf{X} \in \Re^{p\times n}$ represents the energy emitted at different wavenumbers and $\mathbf{Y} \in \Re^{s\times n}$ indicates the temperature profiles and length. Recall that each column corresponds to a different observation. For instance, each sample $\mathbf{x}_i$ is a spectrum of radiance in the infrared range of $2110cm^{-1} - 2410cm^{-1}$ with $p = 2341$. Likewise, each sample $\mathbf{y}_i$ is the corresponding temperature, and length with $q = 200$.

## 4.1 Experimental Design

The synthetic dataset used in this study has been generated with a simulator based on the well known experimental database HITRAN [20]. The parameter ranges used to generate this dataset are based on typical combustion environment conditions. The temperature ranges from 296 K to 1100 K with several different profiles, the length range covers from 0.05 to 0.85 meters, and the concentration values for $CO_2$ and $H_2O$ have been selected as typical ones from combustion of fossil fuels at different temperatures.

## 4.2 Results

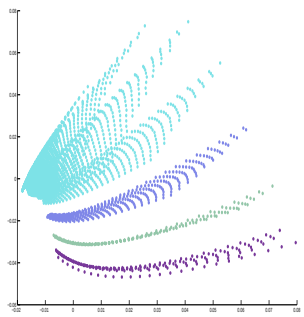


Figure 3: Clusters formed under the first two projected dimensions.

In this section, we report the results obtained from applying the algorithm 1 to the temperature estimation scenario.

We have centered and scaled the dataset (either $\mathbf{X}$ and $\mathbf{Y}$) and we have begun with an unsupervised mode, therefore $\alpha = 0$. After a few iterations, the algorithm stops with an $\alpha$ value of 0.4 which means that we are in somewhere between unsupervised and totally supervised mode.

Ideally these clusters should appear as lines, planes or hyperplanes (depending of the reduction dimensionality), but eventually the projected data does not match exactly with this patterns. The figure 3 illustrates the projected data $[\mathbf{p}_1, \mathbf{p}_2]^T$ into its first two components. It shows the different geometry of the projected data.Therefore instead of looking for lines, planes or hyperplanes, we need to use a density clustering algorithm to identify different density areas related with different models. For this purpose we have used the well known density clustering algorithm DBSCAN algorithm [3].The same figure 3 shows with different colours the four clusters discovered by DBSCAN.

The clusters have different structure showing different linearity. The wider is the structure, the less linear it is. We explained above that ideally the structures should be linear, but it depends on the data. Therefore, we can estimate with a linear regressor three of the models and we can use a non-linear model to estimate the fourth.

To show the improvements between the clustering approach and a global model we have done four different proofs. Each proof corresponds with different data sets retrieval associated to each one of the clusters discovered.

In order to make a fair comparison for each one of the proofs, we have fixed the parameters used in the regression model for every proof. We have used a multilayer perceptron (MLP) as non-linear regressor, and the predictors have been reduced to 80 using a PCA reduction. We also fixed the multilayer perceptron (MLP) architecture to a unique hidden layer with 40 hidden neurons. For the training process, the data is splitted into a training dataset (70%) and a testing dataset (30%). Whether a MLP or a linear regressor has been used, it is indicated in the results (see table 1).

The table is divided in two rows, and each one in another four which corresponds to the number of clusters discovered by our algorithm. The first row from above shows the global model results per each one of the clusters, and the second row shows separately the error for each one of the models.

To give a better insight into the quality of the results we have included the figure 4. For instance, the picture a) shows an example where the global model does not estimate well the temperature profile but the cluster model does. Moreover, the global model has some oscillations which are non desire because in our physical problem the hottest temperature is always at center. In the other hand, the picture b) shows an example where both models achieve successful results, and the estimation error is very low.

Because using a full rank search (see line 10, algorithm 1) can be computationally very expensive, we have decide to lower the value of $r$. One option would be to choose an $r$ equal to the number of eigenvalues which accounts for a relative percentage of the total variance. We have chosen 90% of total variance. This reduces significantly its value due to the multicollinearity of the data.

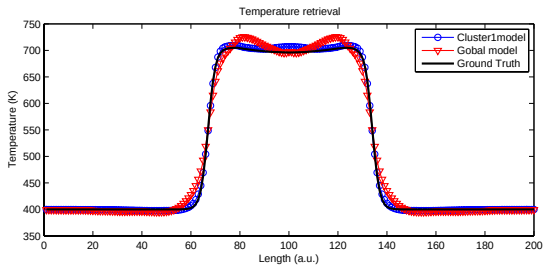
As can bee seen in table 1, and figure 4, the results obtained with our clustering approach improves the results of a global approach.

In section 3, we introduced the idea of iteratively applying the algorithm to the non-linear cluster or the "widest" from a geometrical view. This is needed whenever we still do not reach successful results. In our application we did not exploit this idea because the results obtained with this four clusters were successful. Furthermore, we do not want to subdivide the problem into many clusters because we are in a semi-supervised solution, and an overkill division could lead to a misinterpretation, and classification problems.

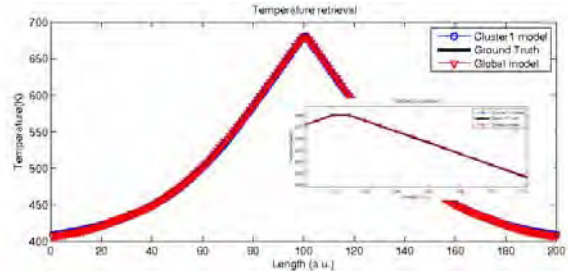
Because we are working in a semi-supervised learning due to the $\alpha$ parameter, we need to answer the next question: how to choose between the four different models associated to the different clusters?.

In this study, the most direct answer is related with the physical properties of the application problem. The four discovered clusters are highly related with different lengths of the temperature profile. Hence, each one of the clusters is related with a local length area where exists a linearity, and the data is homogeneous. Therefore we can make a priori observation of the length of the temperature profile (using other sensoring technique), and then apply the appropriate model.

Another via to solve that problem, deals with learning the function that classifies the data into these four clusters. Depending on the degree of supervised learning where we are working on, the classification will be more or less difficult to learn. We have done some naive proofs using support vector machines to learn the classification and we have obtained a classification error around 6% which is very promising although the proofs have not been extensive enough to consider it with reliability.




(a) Bad performance for a global model but good for the clustering model.




(b) Good performance for both models.

Figure 4: Comparative behavior between the global and cluster approaches.

Table 1: Mean Absolute Error per sample(MAEs) of temperature, and its standard deviation (SD).

| Method | Cluster No. | Temperature **Test** (MAEs/SD)K | Temperature **Train** (MAEs/SD)K |
|---|---|---|---|
| Global Model | Cluster 1(MLP) | 2.04/2.84 | 2.32/3.36 |
| | Cluster 2(MLP) | 3.47/3.32 | 2.38/3.24 |
| | Cluster 3(MLP) | 0.90/1.12 | 0.35/0.49 |
| | Cluster 4(MLP) | 0.97/1.32 | 0.41/0.51 |
| Model 1 | Cluster 1(MLP) | 0.61/1.29 | 0.52/0.60 |
| Model 2 | Cluster 2(linear) | 0.40/0.45 | 0.31/0.26 |
| Model 3 | Cluster 3(linear) | 0.39/0.86 | 0.19/0.16 |
| Model 4 | Cluster 4(linear) | 0.42/0.46 | 0.24/0.19 |

## 5. CONCLUSIONS

In this paper, we have presented a clustering approach based on principal componets to extract the overall redundancy between two sets of high dimensional data in a regression framework. The method is guided by a physical motivation in a remote sensing scenario where we want to predict the temperature profile from the spectrum of energy. We developed a new algorithm, which exploits the redundancy between two data sets to find out different clusters of linearly correlated data between them. Our algorithm reduce the dimensionality using principal components and moreover helps to solve the multicollinearity problem searching for homogeneous correlated clusters.

We evaluated our methodology on a real remote sensing scenario, showing how our algorithm outperforms the results obtained by a single regressor model, and furthermore gives an insight into the physical interpretation of the data which is very valuable.

One question that needs to be addressed is how to choose between the different models due to we are using some information of the response variables to discover linear structures in the data. This is not a main drawback in our remote sensing scenario, because we have discovered that the different clusters are highly related with different lengths of the temperature profile, and therefore we can make a rough but useful classification, based on this parameter.

We believe that the classification can be learned and therefore making a complete automatic retrieval process. In a more general framework, it is clear that the classification learning process success depends on the value of $\alpha$ parameter which controls the shift from unsupervised towards supervised learning. A more deeper study of this parameter and its influence into the classification learning process is needed to obtain more extensive conclusions.

## 6. ACKNOWLEDGMENTS

The authors wish to acknowledge the Spanish Ministry of Education and Science for financial support under the projects TRA2005-08892-C02-01, and MSTAR::UC3M Ref:TIN2008-06491-C04-03. The author E. Garcia-Cuesta also wants to thank Dr. Fernando de la Torre for its valuable guiding towards this work, and Mr. Jacobo Varela for many discussions in the lab.

## 7. REFERENCES

[1] E. Bair, T. Hastie, D. Paul, and R. Tibshirani. Prediction by supervised principal components. *J. Am. Stat. Assoc.*, 101:119–137, 2006.

[2] H. L. De Veaux RD. Multicollinearity: a tale of two nonparametric regressions. *Selecting Models from Data: AI and Statistics IV, (ed.Cheeseman P, Oldford R.W.)*, pages 293–302, 1994.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.

[4] R. D. Fierro, G. H. Golub, P. C. Hansen, and D. P. O'Leary. Regularization by truncated total least squares. *SIAM Journal on Scientific Computing*, 18(4):1223–1241, 1997.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 1990.

[6] E. Garcia-Cuesta, F. de la Torre, and A. J. de Castro. Advances in computational algorithms and data analysis: Machine learning approaches for the

inversion of the radiative transfer equation. *Lectures Notes in Electrical Engineering*, 14:319–333, 2008.

[7] E. Garcia-Cuesta, I. M. Galvan, and A. J. de Castro. Multilayer perceptron as inverse model in a ground-based remote sensing temperature retrieval problem. *J. Eng. Appl. Artif. Intell.*, 21(1):26–34, February 2008.

[8] R. M. Goody and Y. Yung. *Atmospheric Radiation. Theoretical basis (Chap. 2)*.

[9] P. D. Groen. An introduction to total least squares. *Nieuw Archief Voor Wiskunde*, 2:237–254, 1996.

[10] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, pages 1–29, 2007.

[11] A. E. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

[12] I. T. Jollife. *Principal Component Analysis (2nd Ed.)*. Springer Series in Statistics Springer-Verlag (Chap.8), New York, 2002.

[13] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence, special issue on relevance*, 97(1-2):273–324, 1997.

[14] L. H. Liu and J. Jiang. Inverse radiation problem for reconstruction of temperature profile in axisymmetric free flames. *J. Quant. Spectrosc. Radit. Transfer*, 70:207–215, 2001.

[15] G. Lu, Y. Yan, and M. Colechin. A digital imaging based multifuncional flame monitoring system. *IEEE T. Instrum. Meas.*, 53:1152–1158, 2004.

[16] I. Markovsky and S. V. Huffel. Overview of total least-squares methods. *Signal Processing*, 10:2283–2302, 2007.

[17] L. Meier and P. Buhlmann. Smoothing l1-penalized estimators for high.dimensional time-course data. *Electronic Journal of Statistics*, pages 597–615, 2007.

[18] J.-D. Rolle. Optimal subspaces and constrained principal component analysis. *Electronic Journal of Linear Algebra*, 10:201–211, July 2003.

[19] C. Romero, K. S. Li, X., and R. Rossow. Spectrometer-based combustion monitoring for flame stoichiometry and temperature control. *Appl. Therm. Eng.*, 25:659–676, 2005.

[20] L. S. Rothman and et al. The hitran molecular spectroscopic database: edition of 2000 including updates through 2001. *J. Quant. Spectrosc. Radiat. Transfer*, 2003.

[21] M. Thakur, A. Vyas, and C. Shakher. Measurement of temperature and temperature profile of an axisymmetric gaseous flames using lau phase interferometer with linear gratings. *Opt. Laser Eng.*, 36:373–380, 2001.

# APPENDIX

## A. UNDERLYING MODEL OPTIMIZATION

We assume that the data is centered and scaled, then

$$\min_{\mathbf{B},\mathbf{A},\mathbf{H}_x,\mathbf{H}_y} \sum_{i=1}^{n}(\mathbf{Y_i} - \mathbf{BHy_i}) + \sum_{i=1}^{n}(\mathbf{X_i} - \mathbf{AHx}_i) =$$
$$\min_{\mathbf{B},\mathbf{A},\mathbf{Hx},\mathbf{Hy}} \parallel \mathbf{Y} - \mathbf{BHy} \parallel_F + \parallel \mathbf{X} - \mathbf{AHx} \parallel_F$$

subject to

$$\begin{cases} \mathbf{Hx}^T\mathbf{Hx} = \mathbf{I}, \\ \mathbf{Hy}^T\mathbf{Hy} = \mathbf{I}, \\ \mathbf{Hx} \perp \mathbf{Hy} \ \forall\ i,j : i \neq j. \end{cases}$$

then we have to minimize simultaneously both terms to minimize the expression. To do this we can aggregate the two sets of data and their models as

$$\begin{cases} \mathbf{Z} = (\mathbf{Y}, \mathbf{X}) \\ \mathbf{H} = (\mathbf{Hy}, \mathbf{Hx}) \end{cases}$$

then,

$$\min_{\mathbf{H}} \parallel \mathbf{Z} - \mathbf{H}^T\mathbf{Z} \parallel_F =$$
$$\min_{\mathbf{H}} tr\{\mathbf{Z}^T\mathbf{Z} - \mathbf{Z}^T\mathbf{HH}^T\mathbf{Z}\}$$

and including the constraints using the Lagrangian multipliers into the cost function $\ell(H) = \mathbf{Z}^T\mathbf{Z} - \mathbf{Z}^T\mathbf{HH}^T\mathbf{Z} - \lambda\mathbf{H}^T\mathbf{H} - \lambda$, and equating the derivatives to zero the equation is formuled as an eigenvector problem,

$$\mathbf{ZZ}^T\mathbf{H} = \lambda\mathbf{H}$$

# A Novel Measure for Validating Clustering Results Applied to Road Traffic

Yosr Naïja
LIP2 - Faculty of Science of Tunis
Campus Universitaire
2092 El-Manar Tunis, Tunisia,
Tel : +216 71 741 964, Fax : +216 748 843
yosr.naija@gmail.com

Kaouther Blibech Sinaoui
LIP2 - Université 7 Novembre - Carthage
Avenue de la république
BP 77 1054 Amilcar, Tunisia
Tel : +216 71 741 964, Fax : +216 748 843
kaouther.blibech@gmail.com

## ABSTRACT

The clustering validation and clustering interpretation are the two last steps of clustering process. The validation step permits to evaluate the goodness of clustering results using some measures. Valid results are then generally interpreted and used in cluster analysis. The validity measures are classified into three categories: unsupervised measures, supervised measures and relative measures. Several supervised measures have been proposed to perform supervised evaluation such as *entropy*, *purity*, *F-measure*, *Jaccard coefficient* and *Rand statistic*. Generally, these measures evaluate results according to class labels. However, they are not always able to distinguish interpretable clusters because most of them depends on the number of labels. This paper proposes a new supervised evaluation measure - called "homogeneity degree"- that permits to merge the steps of validation and interpretation. Our measure is applied to a real traffic data set and is used to interpret some traffic situations. Comparison with other evaluation measures shows the performance of our proposal.

## Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Data mining*; I.5.3 [**PATTERN RECOGNITION**]: Clustering

## General Terms

Performance

## Keywords

clustering, validity, supervised measure, external criteria

## 1. INTRODUCTION

In domains such as road traffic, there is often a need to extract data measuring the evolution of some parameters. The values of these parameters are generally recorded by sensors placed on roads and are useful for road traffic monitoring systems to supervise the traffic state. Nowadays, these systems face every day a lot of troubles to maintain the traffic fluidity. Solutions have to be found and used to solve the problem of congestion. When a traffic situation is consistent and repetitive such as congestions preceding companies opening hours, traffic engineers can make actions to decrease -or to solve- the problem. For example, they can modify traffic lights synchronization, change real-time traveller's guidance ... These situations are called 'typical' ones and are generally very dependent on human activity and so on academic and administrative calendar. So, it would be judicious to begin by identifying the different types of days - and so the factors - that influence each situation and then to use these factors to guarantee suitable solutions in the future.

Clustering is a widely used technique in data mining applications to achieve this goal. In fact if we consider the values recorded by a sensor along a day as the features of an object and if we associate to each object class labels representing the characteristics of the day associated to this object (holiday, working day, the day of the week, the month day, ...), it will be interesting to use one clustering method to group objects according to traffic values and then to use an evaluation measure based on class labels in order to view how labels are distributed on clusters. This distribution must reveal if every cluster is described by unique labels that "well-represent" the cluster and consequently if the results of clustering are interpretable with respect to the type of days. These labels will then represent the factors that influence a traffic situation.

In literature, there have been several proposals to deal with evaluation measures [1, 2, 4, 5, 6, 7, 8, 10, 11, 14, 13]. The authors of [13] classify these measures into three categories : unsupervised measures, supervised measures and relative measures. The first category uses internal criteria such as distances between objects to verify the compactness of clusters and/or the separation between the clusters. The second category is based on external criteria like class labels and permits to compute the degree of correspondence between cluster labels and class labels. Relative measures permit to compare clustering results obtained with different values of input parameters. In this paper we are interested particularly in supervised measures because they are closely related to our problem .

Several supervised measures have been proposed to per-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SensorKDD'09, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

form supervised evaluation such as *entropy* [12], *purity* [13], *precision*, *recall*, *F-measure* [3, 15], *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index* [6]. However, these measure are not always able to detect interpretable clusters because of the way that these measures handle labels and also the dependance of some of them on the number of labels.

In this paper we propose a new supervised measure called "homogeneity degree" where the validation of results leads to a possible interpretation of them. The basic idea of our measure is to compute the extent to which a cluster contains all objects of a specific label and then for the highest extents to verify that the corresponding objects represent a great proportion of the cluster. This permits us to verify if each cluster is described by *representative labels* and so to be able to interpret the clustering results.

This paper is organized as follows. Section 2 discusses some related works. Section 3 details the proposed approach. Section 4 presents the algorithm illustrating our approach. Section 5 introduces the experiment study and discusses the roles of some parameters used in our approach. Section 6 concludes the paper.

## 2. RELATED WORKS

According to [6], 4 steps are necessary to group objects into clusters: (i) *feature selection* where characteristics of objects are identified; (ii) *clustering algorithm selection* where a clustering method is first chosen and then used to create clusters, (iii) *clustering validation or evaluation* that aims to verify the correctness of clustering results basing on a predefined criteria and (iiii) *results interpretation* which aims to verify that clusters are meaningful and useful.

Clustering evaluation permits to evaluate the individual "goodness" of each cluster or the overall "goodness" of clustering results using some measures. Evaluation measures are generally classified into three categories: unsupervised measures, supervised measures and relative measures[13]. We will only focus in this paper on supervised measures because we are looking to propose a novel measure belonging to this category.

Supervised measures are based on external criteria. They are generally used when objects are described by external attributes corresponding to class labels. For example, credit approval data sets may be described by 2 class labels (approved and rejected) corresponding to the acceptance of the credit or not. It exists 2 different definitions of supervised measures : According to Halkidi, Batistakis and Vazirgiannis, supervised measures aim to evaluate the clustering results by comparing them to a predefined partition which represents the user intuition about the clustering structure [6]. For Tan, Steinbach and Kumar in [13], supervised measures aim to compute the degree of correspondence between cluster labels and class labels. With this definition, there is no need for a pre-specified partition. In this paper, we will use the latter definition because we think that it is more intuitive and more general.

Several measures have been proposed to express this degree such as *entropy* [12], *purity* [13], *precision*, *recall*, *F-measure* [3, 15], *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index*[6]. The following sections present these measures and their limits.

### 2.1 Supervised measures

*Entropy* aims to distinguish the clusters described by a single class label. Given $p_{ij}$ the proportion of objects of cluster $i$ described by class label $j$ and $L$ the number of class labels, the *entropy* of a cluster $i$ is computed as $-\sum_{j=1}^{L} p_{ij} \log_2 p_{ij}$ [12]. *Purity* has the same role than *entropy*. The *purity* of a cluster $i$ is equal to $max_j(p_{ij})$ [13]. The overall *entropy* of clusters (resp. overall *purity*) is equal to a weighted sum of all *entropies* (resp. *purities*). An *entropy* (resp. *purity*) value close to 0 (resp. close to 1) means that the cluster is well described.

*Precision* expresses the proportion of objects of cluster $i$ described by a class label $j$ ($p_{ij}$). *Recall* reflects the proportion of objects of a class label $j$ in a cluster $i$ ($m_{ij}$) in respect to the total number of objects of class label $j$ ($m_j$) and is equal to $m_{ij}/m_j$. *F-measure* combines the two measures *precision* and *recall*. The general formula of *F-measure* is $F(i,j) = \frac{(1+\beta^2).precision(i,j).recall(i,j)}{\beta^2.precision(i,j)+recall(i,j)}$ [3, 15]. Traditionally, it is computed using a $\beta$ value equal to 1. The other values of $\beta$ lead to favouring either precision or recall. To compute the overall *F-measure* of the clustering result, we pick out for every class label, the highest value of *F-measure* in all clusters and then we compute the weighted average of these *F-measure* values. A *F-measure* value close to 1 means a *good* clusters structure .

Concerning *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index* , given the partition $P_1$ generated by clustering method and a partition $P_2$ defined such that the objects having the same label are grouped into the same cluster, the 3 measures are based on computing the correspondence degree between $P_1$ and $P_2$ and precisely on the computation of the following values:

- **MM** : the number of couples of objects that belong to the same cluster in $P_1$ and to the same cluster in $P_2$.
- **MD** : the number of couples of objects that belong to the same cluster in $P_1$ but to different clusters in $P_2$.
- **DM** : the number of couples of objects that belong to different clusters in $P_1$ and to the same cluster in $P_2$.
- **DD** : the number of couples of objects that belong to different clusters in $P_1$ and to different clusters in $P_2$.

The sum of values MM, MD, DM et DD is noted $M$. Given these $M$ couples, *Jaccard coefficient* (J), *Rand statistic* (R) and *Folkes and Mallows index* (FM) are defined as [6]:

$$J = \frac{MM}{MM + MD + DM} \tag{1}$$

$$R = \frac{MM + DD}{M} \tag{2}$$

$$FM = \sqrt{\frac{MM}{MM + MD} \cdot \frac{MM}{MM + DM}} \tag{3}$$

The values of these measures vary between 0 and 1. Good results are obtained when these values are near to 1.

### 2.2 Limits of the existing supervised measures

As mentioned in the introduction, we intend to use a supervised evaluation measure that permits the detection of interpretable clusters. However the measures presented above are not always able to reflect this situation. This is due to

three reasons: the inability to reflect the overall validity of a cluster and/or the partition, the way that the measures handle labels and the dependance on the number of labels.

The first type of reasons concerns *Precision* and *Recall*. These measures evaluate cluster only according to a given label. They are not able to reflect the overall validity of a cluster and they can't be used to interpret the clustering results.

The second type concerns *Entropy* and *Purity*. If we consider a cluster that contains the majority of objects of class label 1 and the majority of objects of class label 2 and where the proportion of each category of objects represent the half of the cluster cardinality, the cluster can be considered as interpretable since it is described by unique labels. However, *entropy* and *purity* are not able to reflect this situation because they are only based on the proportion of objects of a cluster described by a class label $j$ in respect to the total number of cluster'objects and not in respect to the total number of objects of $j$ in the data set.

The dependance on the number of labels concerns *F-measure*, *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index*. To illustrate this dependance, consider the example given in figure 1 illustrating the distribution of 7 labels on 2 clusters $C_1$ and $C_2$. Cluster $C_1$ is composed of all objects having label $l_1$ or $l_2$ (the number of labels is equal to 2). Cluster $C_2$ contains all objects described by the remaining labels $l_3$, $l_4$, $l_5$, $l_6$ and $l_7$ (the number of labels is equal to 5). So, the cluster $C_1$ and $C_2$ are interpretable since they are described by unique labels ($l_1$ or $l_2$ for the cluster $C_1$ and $l_3$, $l_4$, $l_5$, $l_6$ for the cluster $C_2$). However, when we compute the values of *F-measure*, *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index*, we have:

- $F(C_1, l_1) = 0.66$
- $F(C_1, l_2) = 0.66$
- $F(C_2, l_3) = 0.33$
- $F(C_2, l_4) = 0.33$
- $F(C_2, l_5) = 0.33$
- $F(C_2, l_6) = 0.33$
- $F(C_2, l_7) = 0.33$
- *Overall F-measure* $= \frac{2\times(3\times0.66)+5\times(3\times0.33)}{3\times7} = 0.42$
- $J = 0.1$
- $R = 0.52$
- $FM = 0.175$

*F-measure* values of $l_3$, $l_4$, $l_5$, $l_6$ and $l_7$ in cluster $C_2$ are very low in contrary to ones of $l_1$ and $l_2$ in cluster $C_1$ due to the increase of the number of labels in cluster $C_2$. This leads to a low value of the overall F-measure ($= 0.42$) and to the invalidity of clustering results. We note also the low values of *Jaccard coefficient*, *Rand statistic* and *Folkes and Mallows index*. Consequently, even if each cluster contains all objects of a given class label and is interpretable, all these measures are not able to express it. This confirms the sensibility of these measures to the number of labels.

The following section presents our approach based on a new evaluation measure called "homogeneity degree" that permits to take into account these limits.

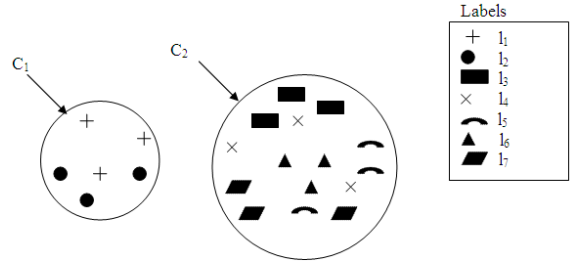



**Figure 1: Distribution of 7 labels on 2 clusters**

## 3. PROPOSED HOMOGENEITY DEGREE-BASED EVALUATION MEASURE

The basic idea of our evaluation approach is to merge the steps of validation and interpretation. Indeed, the criteria that we use to validate the correctness of clustering results is the fact that these results must be interpretable. In other words, we consider that a partition is valid if every cluster of the partition is described by unique labels and if, for each cluster, the proportion of objects having the associated labels represent more than the half of the cluster cardinality. These labels will be called *representative labels*. To reach this goal, we compute for every label the extent to which a cluster contains all objects of this specific label and then for the highest extents we verify that the corresponding objects represent a great proportion of the cluster. We call the measure that illustrates this idea "Partition Homogeneity degree". Algorithm 1 in section 4 illustrates this idea.

The homogeneity degree of a partition depends on the homogeneity degrees of its clusters. In this section, we present these two degrees and then we give an illustrative example and some propositions to explain better the new concepts.

### 3.1 Cluster homogeneity based on class labels

Naturally, a cluster is homogeneous according to its labels if its objects have the same label. However, a crisp definition of the notion of homogeneity may be very restrictive in practice and makes homogeneity largely depending on the application's domain and on the data sets. The homogeneity degree concept as introduced in the rest of this section provides a fuzzy measure of homogeneity and is called *overall homogeneity degree* of a cluster.

The homogeneity of a cluster depends on two hypothesis. First of all, the number of objects of a cluster described by a given label must represent at least a proportion greater than 50% of the objects of the domain associated to the same label (the majority rule). The degree that illustrates this idea is called *homogeneity degree in respect to a domain*. Secondly, the number of objects verifying the first hypothesis must represent a proportion greater than the half of cluster cardinality. This hypothesis is illustrated by *partial homogeneity degree*. A cluster verifying these two hypothesis is called *valid-interpretable* cluster. The higher the homogeneity degree in respect to a domain and the partial homogeneity degree are, the better the interpretability of the cluster is.

Overall homogeneity degree of a cluster takes its values in the range [0,1], where 1 indicates that the cluster is a fully *valid-interpretable* cluster and 0 indicates that the cluster

is not a *valid-interpretable* one. A value between 0 and 1 indicates the degree of *validity-interpretability* of the cluster.

The definition of overall homogeneity degree requires the use of two thresholds $\alpha \in [0.5, 1]$ and $\beta \in [0.5, 1]$ assuring the coherence of results. The practical utility of these parameters will be better explained later.

Next, some useful notations are introduced.

- $X = \{O_1, O_2, \ldots, O_n\}$: a set of $n$ objects, also called *the domain*.
- $L = \{l_1, l_2, \ldots, l_m\}$: a set of $m$ labels.
- $P_k(C_1, C_2, \ldots, C_k)$: a partition of $k$ clusters.
- $SC_{ij} = \{O_r \in C_i : Label(O_r) = l_j\}$: the set of objects of cluster $C_i$ having the label $l_j$.
- $S_j = \{O_r \in X : Label(O_r) = l_j\}$: the set of objects of $X$ having the label $l_j$.
- $\| \cdot \|$: set cardinality symbol.

### 3.1.1 Membership degree of a label to the domain

The membership degree $MD_\alpha(l_j, C_i)$ of a label $l_j$ to the domain $X$ in respect to a cluster $C_i$ expresses the proportion of objects of $C_i$ described by label $l_j$ in respect to the total number of objects of $X$ having the label $l_j$. $MD_\alpha(l_j, C_i)$ is defined as follows:

$$MD_\alpha(l_j, C_i) = \begin{cases} \frac{\|SC_{ij}\|}{\|S_j\|}, & \text{if } \frac{\|SC_{ij}\|}{\|S_j\|} \geq \alpha \text{ and } \| S_j \| \neq 0; \\ 0, & \text{Otherwise} \end{cases} \quad (4)$$

As it is shown in Eq. 4, $MD_\alpha(l_j, C_i)$ will be equal to 0 any time the proportion of objects described by label $l_j$ is strictly less than a given threshold $\alpha$. This threshold represents a sensibility indicator assuring that $C_i$ contains at least $(100 * \alpha)\%$ of objects of $X$ described by the label $l_j$.

### 3.1.2 Homogeneity degree of a cluster in respect to a domain

Homogeneity degree of $C_i$ in respect to the domain, denoted $HD_\alpha(C_i)$, represents the mean of the non-null membership degree of the different labels to the domain. It is computed as follows:

$$HD_\alpha(C_i) = \begin{cases} \frac{1}{B} \cdot \sum_{j=1}^{m} MD_\alpha(l_j, C_i), \text{if } B \neq 0; \\ 0, \text{Otherwise.} \end{cases} \quad (5)$$

where $B = \| \{l_k : l_k \in L \wedge MD_\alpha(l_k, C_i) > 0\} \|$, that is, the number of labels $l_k$ such that $MD_\alpha(l_k, C_i)$ is strictly positive.

### 3.1.3 Membership degree of a label to a cluster

The membership degree $MC_\alpha(l_j, C_i)$ of a label $l_j$ to a cluster $C_i$ reflects the proportion of objects of $C_i$ described by $l_j$. In other words, $MC_\alpha(l_j, C_i)$ permits to measure the importance of label $l_j$ in cluster $C_i$. Formally, $MC_\alpha(l_j, C_i)$ is computed as follows:

$$MC_\alpha(l_j, C_i) = \begin{cases} \frac{\|SC_{ij}\|}{\|C_i\|}, & \text{if } MD_\alpha(l_j, C_i) \geq \alpha; \\ 0, & \text{Otherwise} \end{cases} \quad (6)$$

As it is shown in Eq. 6, only labels for which $MD_\alpha \geq \alpha$ are included in the definition of $MC_\alpha(l_j, C_i)$. This ensures that only the most important labels in respect to the domain are considered to compute $MC_\alpha(l_j, C_i)$.

### 3.1.4 Partial homogeneity degree of a cluster

The partial homogeneity degree of a cluster $C_i$, denoted $HC_{\alpha,\beta}(C_i)$, computes the proportion of objects of $C_i$ described by labels that have a certain importance in respect to the domain ($MD_\alpha(l_j, C_i) \geq \alpha$). Like for $MD_\alpha$, we use a threshold $\beta$ to force the cluster to contain at least a proportion greater than $\beta$. Formally, the partial homogeneity degree $HC_{\alpha,\beta}(C_i)$ is given by Eq. 7:

$$HC_{\alpha,\beta}(C_i) = \begin{cases} \sum_{j=1}^{m} MC_\alpha(l_j, C_i), \\ \text{if } \sum_{j=1}^{m} MC_\alpha(l_j, C_i) \geq \beta; \\ 0, \\ \text{Otherwise} \end{cases} \quad (7)$$

### 3.1.5 Overall homogeneity degree of a cluster

The overall homogeneity degree of a cluster $C_i$ takes into account the homogeneity degree of $C_i$ in respect to the domain and the partial homogeneity of $C_i$. It is denoted by $D_{\alpha,\beta}(C_i)$ and is computed through Eq. 8 hereafter:

$$D_{\alpha,\beta}(C_i) = HD_\alpha(C_i) \cdot HC_{\alpha,\beta}(C_i) \quad (8)$$

For a cluster $C_i$, a value of $D_{\alpha,\beta}$ different from 0 means that the cluster is a *valid-interpretable* one. In this case, the labels that have a $MD_\alpha(l_j, C_i) \neq 0$ are called *representative labels* of the cluster $C_i$.

## 3.2 Partition homogeneity degree

For a partition $P_k(C_1, C_2, \ldots, C_k)$, a cluster $C_i$ may have a maximal homogeneity for a given $\alpha$ and $\beta$. However, the homogeneity degree of another cluster $C_j$ may not be maximal for these values $\alpha$ and $\beta$. To overcome this problem, we introduce the notion of *homogeneity degree of a partition*.

Let $P_k(C_1, C_2, \ldots, C_k)$ be a partition of $k$ clusters. The homogeneity degree of $P_k$ is defined by the function $DP_{\alpha,\beta}(P_k)$ and is computed as follows

$$DP_{\alpha,\beta}(P_k) = \begin{cases} \frac{\sum_{i=1}^{i=k} D_{\alpha,\beta}(C_i)}{k}, \\ \text{if } \forall i, D_{\alpha,\beta}(C_i) \neq 0; \\ 0, \\ \text{otherwise} \end{cases} \quad (9)$$

The value of $DP_{\alpha,\beta}(P_k)$ varies between 0 and 1. When this value is different from 0, the partition is said a *valid-interpretable* partition.

## 3.3 Illustrative example

Consider a set $X$ of objects described by 4 class labels :

- Label $l_1$ indicates that the day is between Monday and Friday and is a holiday.
- Label $l_2$ indicates that the day is between Monday and Friday and is not a holiday.
- Label $l_3$ indicates that the day is a Saturday or a Sunday and is a holiday.
- Label $l_4$ indicates that the day is a Saturday or a Sunday and is not a holiday.

Let $L = \{l_1, l_2, l_3, l_4\}$ be the set of labels. Suppose that $\| S_1 \| = 5$, $\| S_2 \| = 10$, $\| S_3 \| = 6$, and $\| S_4 \| = 4$. Let $C_1$ and $C_2$ be two clusters defined on $X$. The content of these two clusters is shown in Figure 2. Cluster $C_1$ contains 12 objects (three are described by label $l_1$, one by $l_2$, six by $l_3$ and two by $l_4$). Cluster $C_2$ contains 13 objects (two are described by label $l_1$, nine by $l_2$ and two by $l_4$).

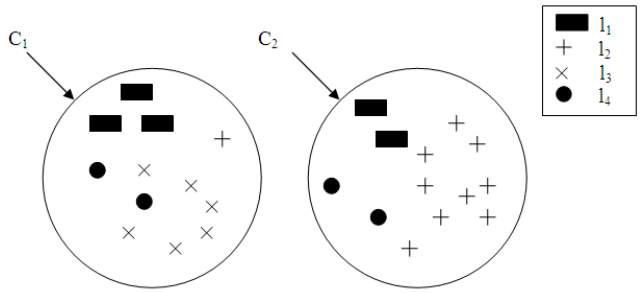



**Figure 2: clusters $C_1$ and $C_2$**

Consider now cluster $C_1$ and suppose that $\alpha = 0.6$ and $\beta = 0.5$. Using Eq. 4 and Eq. 6, we get:

- $MD_{0.6}(l_1, C_1) = 0.6$; $MC_{0.6}(l_1, C_1) = 0.25$,
- $MD_{0.6}(l_2, C_1) = 0$; $MC_{0.6}(l_2, C_1) = 0$,
- $MD_{0.6}(l_3, C_1) = 1$; $MC_{0.6}(l_3, C_1) = 0.5$,
- $MD_{0.6}(l_4, C_1) = 0$; $MC_{0.6}(l_4, C_1) = 0$.

For instance: $MD_{0.6}(l_1, C_1) = 0.6$ since $\frac{||SC_{1,1}||}{||S_1||} = \frac{3}{5} = 0.6 \geq \alpha$ and $MD_{0.6}(l_2, C_1) = 0$ since $\frac{||SC_{1,2}||}{||S_2||} = \frac{1}{10} = 0.1 < \alpha$.

Then, by using Eq. 5 and Eq. 7, we get:

- $HD_{0.6}(C_1) = \frac{1}{2}(MD_{0.6}(l_1, C_1) + MD_{0.6}(l_3, C_1)) = 0.8$,
- $HC_{0.6,0.5}(C_1) = MC_{0.6}(l_1, C_1) + MC_{0.6}(l_2, C_1) + MC_{0.6}(l_3, C_1) + MC_{0.6}(l_4, C_1) = 0.75$.

Next, after the computation of $HD_{0.6}(C_2)$ and $HC_{0.6,0.5}(C_2)$, the homogeneity degrees of clusters $C_1$ and $C_2$ can be calculated by Eq. 8. This leads to:

$$\begin{aligned} D_{0.6,0.5}(C_1) &= HD_{0.6}(C_1) \cdot HC_{0.6,0.5}(C_1) \\ &= 0.8 \cdot 0.75 \\ &= 0.6 \end{aligned}$$

$$\begin{aligned} D_{0.6,0.5}(C_2) &= HD_{0.6}(C_2) \cdot HC_{0.6,0.5}(C_2) \\ &= 0.62 \end{aligned}$$

Finally, the homogeneity degree of partition $P_2(C_1, C_2)$ is computed by Eq. 9:

$$\begin{aligned} DP_{0.6,0.5}(P_2) &= \frac{1}{2}(D_{0.6,0.5}(C_1) + D_{0.6,0.5}(C_2)) \\ &= \frac{1}{2}(0.6 + 0.62) \\ &= 0.61 \end{aligned}$$

Tables 1, 2 and 3 hereafter give the values of $D_{\alpha,\beta}(C_1)$, $D_{\alpha,\beta}(C_2)$ and $DP_{\alpha,\beta}(P_2)$ for some values of $\alpha$ and $\beta$. It is easy to see that for $\alpha = 1$ and $\beta = 0.5$, cluster $C_1$ is more homogeneous than cluster $C_2$, while for $\alpha = 0.6$ and $\beta = 0.5$, $C_2$ is more homogeneous than $C_1$.

**Table 1: Values of $HD_\alpha(C_1)$, $HC_{\alpha,\beta}(C_1)$ and $D_{\alpha,\beta}(C_1)$**

| $\alpha$ | $\beta$ | $HD_\alpha(C_1)$ | $HC_{\alpha,\beta}(C_1)$ | $D_{\alpha,\beta}(C_1)$ |
|---|---|---|---|---|
| 0.6 | 0.5 | 0.8 | 0.75 | 0.6 |
| | 0.7 | | 0.75 | 0.6 |
| | 0.9 | | 0 | 0 |
| 0.8 | 0.5 | 1 | 0.5 | 0.5 |
| | 0.7 | | 0 | 0 |
| | 0.9 | | 0 | 0 |
| 1 | 0.5 | 1 | 0.5 | 0.5 |
| | 0.7 | | 0 | 0 |
| | 0.9 | | 0 | 0 |

**Table 2: Values of $HD_\alpha(C_2)$, $HC_{\alpha,\beta}(C_2)$ and $D_{\alpha,\beta}(C_2)$**

| $\alpha$ | $\beta$ | $HD_\alpha(C_2)$ | $HC_{\alpha,\beta}(C_2)$ | $D_{\alpha,\beta}(C_2)$ |
|---|---|---|---|---|
| 0.6 | 0.5 | 0.9 | 0.69 | 0.62 |
| | 0.7 | | 0 | 0 |
| | 0.9 | | 0 | 0 |
| 0.8 | 0.5 | 0.9 | 0.69 | 0.62 |
| | 0.7 | | 0 | 0 |
| | 0.9 | | 0 | 0 |
| 1 | 0.5 | 0 | 0 | 0 |
| | 0.7 | | 0 | 0 |
| | 0.9 | | 0 | 0 |

**Table 3: Values of $DP_{\alpha,\beta}(P_2)$ for different values of $\alpha$ and $\beta$**

| $\alpha$ | $\beta$ | $D_{\alpha,\beta}(C_1)$ | $D_{\alpha,\beta}(C_2)$ | $DP_{\alpha,\beta}(P_2)$ |
|---|---|---|---|---|
| 0.6 | 0.5 | 0.6 | 0.62 | 0.61 |
| | 0.7 | 0.6 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 |
| 0.8 | 0.5 | 0.5 | 0.62 | 0.56 |
| | 0.7 | 0 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 |
| 1 | 0.5 | 0.5 | 0 | 0 |
| | 0.7 | 0 | 0 | 0 |
| | 0.9 | 0 | 0 | 0 |

## 3.4 Propositions

This section presents a series of propositions concerning the relationship between $HD_\alpha$, $HC_{\alpha,\beta}$, $MD_\alpha$ and $MC_\alpha$.

PROPOSITION 1. *Giving a cluster $C_i$,*

- *Part 1: if $\max_j(\frac{\|SC_{ij}\|}{\|S_j\|}) < \alpha$, then $HD_\alpha(C_i) = 0$*
- *Part 2: given $\alpha = 1$, if $\nexists\, l_j$ such that $\frac{\|SC_{ij}\|}{\|S_j\|} = 1$, then, $HD_\alpha(C_i) = 0$, otherwise $HD_\alpha(C_i) = 1$*
- *Part 3: if there is at least one label $l_j$ such that $\frac{\|SC_{ij}\|}{\|S_j\|} = 1$, then $\lim_{\alpha \to 1}(HD_\alpha(C_i)) = 1$*

PROOF. :

- Part 1: given a cluster $C_i$, $\max_j(\frac{\|SC_{ij}\|}{\|S_j\|}) < \alpha$ means that for every label $l_j$ $(j = 1, \ldots, m)$, $\frac{\|SC_{ij}\|}{\|S_j\|} < \alpha$, then for every label $l_j$ $(j = 1, \ldots, m)$, $MD_\alpha(l_j, C_i) = 0$. Consequently, according to Eq. 5, $HD_\alpha(C_i) = 0$.

- Part 2: as consequence of the Part 1, given $\alpha = 1$, if $\nexists$ $l_j$ such that $\frac{\|SC_{ij}\|}{\|S_j\|} = 1$, then, it is trivial to conclude that $HD_\alpha(C_i) = 0$. Otherwise, given a cluster $C_i$, $HD_\alpha(C_i)$ is equal to the mean of the $MD_\alpha(l_j, C_i)$, $\forall l_j$, that are different from 0. When $\alpha = 1$, we take into account only the values of $\frac{\|SC_{ij}\|}{\|S_j\|}$, $\forall l_j$, that are equal to 1 to compute $HD_\alpha(C_i)$ (the maximum value of the proportion $\frac{\|SC_{ij}\|}{\|S_j\|}$ is 1). So, if there is at least one label $l_j$ such that $\frac{\|SC_{ij}\|}{\|S_j\|} = 1$, then $HD_\alpha(C_i) = 1$.
- Part 3: as mentioned above, $HD_\alpha(C_i)$ is equal to the mean of the $MD_\alpha(l_j, C_i)$ that are different from 0. If $\alpha$ is close to 1, so for the computation of $HD_\alpha(C_i)$, we take into account only the values of $\frac{\|SC_{ij}\|}{\|S_j\|}$, $\forall l_j$, that are greater than $\alpha$. Then, the mean will be greater than $\alpha$ which is close to 1.

□

PROPOSITION 2. *Let $Pl_j$ be a set of $r$ clusters containing objects having the label $l_j$. The following holds:*

- *Part 1: if $\forall$ $C_i \in Pl_j$, $MD_\alpha(l_j, C_i) \neq 0$, then $\sum_{i=1}^{r} MD_\alpha(l_j, C_i) = 1$ .*
- *Part 2: If for a given cluster $C_i \in PL_j$, $MD_\alpha(l_j, C_i) > 0.5$, then $\forall$ $C_m \neq C_i$ and $\in PL_j$, $MD_\alpha(l_j, C_m) < 0.5$*

PROOF. :

- Part 1: according to section 3.1.1, $MD_\alpha(l_j, C_i)$ is equal to the proportion of objects of a cluster $C_i$ described by a label $l_j$ in respect to the total number of the objects of $X$ having the label $l_j$ when this proportion is greater than $\alpha$. Next, if $\forall$ $C_i \in Pl_j$, $MD_\alpha(l_j, C_i) \neq 0$, then:

$$\begin{aligned}\sum_{i=1}^{r} MD_\alpha(l_j, C_i) &= \frac{\| SC_{1j} \|}{\| S_j \|} + \frac{\| SC_{2j} \|}{\| S_j \|} + \cdots + \frac{\| SC_{rj} \|}{\| S_j \|} \\ &= \frac{\| SC_{1j} \| + \| SC_{2j} \| + \cdots + \| SC_{rj} \|}{\| S_j \|} \\ &= 1\end{aligned}$$

  Otherwise, if $\exists$ $C_i \in Pl_j$ such that $MD_\alpha(l_j, C_i) = 0$, we will have some proportions that will not be taken into account and so $\sum_{i=1}^{r} MD_\alpha(l_j, C_i)$ will be strictly less than 1.

- Part 2: Basing on the part 1 of the proposition 2, if for a given cluster $C_i \in PL_j$, $MD_\alpha(l_j, C_i) > 0.5$ so: $\sum_{m=1, m\neq i}^{r} MD_\alpha(l_j, C_m) < 0.5$. Consequently, $\forall$ $C_m \neq C_i$ and $\in PL_j$, $MD_\alpha(l_j, C_m) < 0.5$.

□

PROPOSITION 3. *Let $C_i$ be a cluster with labels list $Labels(C_i)$. If $\forall l_j \in Labels(C_i)$ $MD_\alpha(l_j, C_i) \geq \alpha$, then $HC_{\alpha,\beta}(C_i) = 1$.*

PROOF. $MD_\alpha(l_j, C_i) \geq \alpha$ means that $MC_\alpha(l_j, C_i) = \frac{\|SC_{ij}\|}{\|C_i\|}$. Consequently, if $\forall l_j \in Labels(C_i)$ we have $MD_\alpha(l_j, C_i) \geq \alpha$, then,

$$\begin{aligned}HC_{\alpha,\beta}(C_i) &= \frac{\| SC_{i1} \|}{\| C_i \|} + \frac{\| SC_{i2} \|}{\| C_i \|} + \cdots + \frac{\| SC_{ij} \|}{\| C_i \|} \\ &= \frac{\| SC_{i1} \| + \| SC_{i2} \| + \cdots + \| SC_{ij} \|}{\| C_i \|} \\ &= 1\end{aligned}$$

□

PROPOSITION 4. *If $HD_\alpha(C_i) = 0$, then $HC_{\alpha,\beta}(C_i) = 0$.*

PROOF. $HD_\alpha(C_i) = 0$ means that $\forall l_j$, $MD_\alpha(l_j, C_i) = 0$. Thus, according to Eq. 6, $MC_\alpha(l_j, C_i) = 0$, $\forall l_j$. Consequently, $HD_{\alpha,\beta}(C_i) = 0$. □

# 4. HOMOGENEITY DEGREE-BASED EVALUATION ALGORITHM

The basic idea of the proposed algorithm is to run one of the clustering methods (`CLUSTERING` algorithm) and compute the partition homogeneity degree $DP_{\alpha,\beta}$ for different values of $\alpha$ and $\beta$. The values of $\alpha$ and $\beta$ that maximize $DP_{\alpha,\beta}$ are stored and the value of $DP_{\alpha,\beta}$ is hold.

```
Procedure homogeneity_degree( X : Set of objects)
    Pk← CLUSTERING(X)
    DPmax ← 0
    αmax ← 0
    βmax ← 0
    For α from 0.5 to 1 by step_α do
        For β from 0.5 to 1 by step_β do
            compute DP_{α,β}(Pk)
            If (DP_{α,β}(Pk) > DPmax) then
                DPmax ← DP_{α,β}(Pk)
                αmax ← α
                βmax ← β
            end If
        end For
    end For
    return (αmax, βmax, DPmax)
End
```

**Algorithm 1: Homogeneity degree-based evaluation algorithm**

Thresholds $\alpha$ and $\beta$ vary from 0.5 to 1 by step $step_\alpha$ and $step_\beta$, respectively. The parameters $step_\alpha$ and $step_\beta$ are chosen by the user before running the algorithm. Both $step_\alpha$ and $step_\beta$ must be in the range $]0, 0.5]$. Values of these steps can be 0.01, 0.1, 0.2 ...

The complexity of the algorithm is $O(CP + \frac{1}{step_\alpha}\frac{1}{setp_\beta})$ where $CP$ is the complexity of the used clustering algorithm.

# 5. EXPERIMENTS AND DISCUSSIONS

This section presents an application of the proposed algorithms on a real road traffic data set. Since our objective is not the comparison of clustering algorithms, we choose the well-known K-means [9] to test our algorithm.

## 5.1 Road Traffic data set

Road traffic measures, also called fundamental data, are generally recorded by sensors. They permit to describe the state of the traffic in space and time. In this paper, we are interested in flow measures. A flow measure corresponds to the number of vehicles that pass through point x in a road

network during a time interval I. It is expressed in vehicles per unit of time (generally hours or minutes).

More than 300 sensors are placed on a road of a french city. Every day, each sensor records 480 values of flow measure (one value every 3 minutes). An object of 480 attributes $\{A_1, A_2, \ldots, A_{480}\}$ is associated to each sensor where $A_1$ is the number of vehicles on 00h00, $A_2$ is the number of vehicles on 00h03, and so on. Along the year 2003, about 365 objects were collected for every sensor. We focus on the data set given by one of these sensors to experiment our approach. The data set contains 365 objects. Among them, 132 objects have some missing values of numeric attributes. Therefore, since the proposed algorithm do not deal with missing values, we use only 233 objects. We use also two sets of labels ($L_1$ and $L_2$) to test our approach. The labels associated to these sets are built from 3 attributes :

- Attribute $A_{481}$ which indicates if the flow values are recorded in a working day (between Monday and Friday) or not. The values domain of $A_{481}$ is $\{Y, N\}$.
- Attribute $A_{482}$ which is associated to the day of a week. Its values domain is $\{1, 2, \ldots, 7\}$.
- Attribute $A_{483}$ indicating if a day is a holiday or not. The values domain of $A_{483}$ is $\{Y, N\}$.

The first set $L_1$ is built from the attributes $A_{481}$ and $A_{483}$. Each label of the second set $L_2$ is composed of the two attributes $A_{482}$ and $A_{483}$. Instead of creating labels using cartesian product of attributes , we will look through objects and extract only the values of the attributes that appear together. For instance, for the second set $L_2$, some labels such that $3Y$ (a wednesday holiday) or $7Y$ ( a sunday holiday) are not associated to any object in data set for the year 2003. So, it is not necessary to include them in the set of labels. According to that, $L_1$ will contain 4 labels ($\{YY, YN, NY, NN\}$) and $L_2$ will be composed of 12 labels ($\{1N, 2N, 3N, 4N, 5N, 6N, 7N, 1Y, 2Y, 4Y, 5Y, 6Y\}$).

## 5.2 Application of the homogeneity degree-based evaluation algorithm

In this section, we present the clustering results obtained by homogeneity degree-based evaluation algorithm and compare the proposed degree of homogeneity used by the algorithm to the other supervised measures such that entropy, purity, F-measure, Jaccard, Rand and Folkes and Mallows. First, the tests are made on the set $L_1$ that contains 4 labels. Then we will use the set $L_2$ of 12 labels to show the performance of our proposed measure in comparison to the other ones when the number of labels increases (see section 2.2).

Table 4 (resp. table 5) shows the K-means clustering results obtained with the set of labels $L_1$ (resp. set $L_2$) and $k = 2$. For every cluster, we compute the cluster homogeneity degree (D), the entropy (E) and the purity (P) and then we give the values of the partition homogeneity degree (DP), the overall entropy (OE), the overall purity (OP), the overall F-measure (OF), Jaccard coefficient (J), Rand statistic (R) and Folkes and Mallows index (FM) to evaluate the final clustering results.

As it is shown in table 4, the values of all the measures reflect the goodness of clustering results. Indeed, we have $DP_{0.9,0.9} = 0.97$, $OE = 0.2$, $OP = 0.96$, $OF = 0.96$, $J = 0.93$, $R = 0.96$ and $FM = 0.96$. Nevertheless, if we concentrate on cluster $C_1$, we can note that the value of cluster homogeneity $D_{0.9,0.9}(C_1)$ is particulary better than entropy. This confirms that entropy is not always able to express the fact that a cluster is composed by *representative labels* even if objects described by one label represent a great proportion of the cluster (see label $NN$ in cluster $C_1$).

**Table 4: Results of the application of Algorithm 1 with 4 labels**

| | $C_1$ | $C_2$ | **Tot.** |
|---|---|---|---|
| **YY** | 4 | 0 | 4 |
| **YN** | 3 | 155 | 158 |
| **NY** | 1 | 0 | 1 |
| **NN** | 70 | 0 | 70 |
| **Tot.** | 78 | 155 | 233 |
| $\mathbf{D}_{0.9,0.9}$ | 0.96 | 0.98 | $\mathbf{DP}_{0.9,0.9}$=0.97 |
| **E** | 0.62 | 0 | **OE**=0.2 |
| **P** | 0.89 | 1 | **OP**=0.96 |
| **OF** | - | - | 0.96 |
| **J** | - | - | 0.93 |
| **R** | - | - | 0.96 |
| **FM** | - | - | 0.96 |

Table 5 shows the distribution of labels into clusters and the values of the supervised measures when the number of labels increases and become equal to 12 (labels of set $L_2$). If we look at the distribution of labels of clusters $C_1$ and $C_2$, we can note that $C_1$ contains all objects of labels $1N$, $2N$, $3N$, $4N$ and $5N$, and the majority of objects of label $6N$. The second cluster contains all objects associated to label $7N$, $1Y$, $2Y$, $4Y$, $5Y$ and $6Y$ and 10 objects out of 25 of label $6N$. According to the point of view explained in section 3, the clusters are *valid-interpretables*, and evaluation measures must reflect this idea. The results of table 5 show that whether we look on the goodness of each cluster separably or on the goodness of the overall clustering results, only our measure of homogeneity degree is able to valid the results. For example, if we concentrate on the validation of the overall clustering results, we have:

- $DP_{0.7,0.8}(P_2) = 0.87$
- $OE = 2.32$
- $OP = 0.3$
- $OF = 0.36$
- $J = 0.18$
- $R = 0.45$
- $FM = 0.41$

These values confirm that the measures of entropy, purity, F-measure, Jaccard, Rand and Folkes and Mallows are dependent on the number of labels and cannot always validate interpretable clusters.

## 5.3 Identification of factors associated to typical situations

As mentioned in section 1, a typical traffic situation is a consistent and repetitive situation. Concretely, if we consider that each cluster is represented by its centroid, each typical situation will be represented by the centroid of a

**Table 5: Results of the application of Algorithm 1 with 12 labels**

| | $C_1$ | $C_2$ | **Tot.** |
|---|---|---|---|
| **1N** | 32 | 0 | 32 |
| **2N** | 29 | 0 | 29 |
| **3N** | 34 | 0 | 34 |
| **4N** | 28 | 0 | 28 |
| **5N** | 35 | 0 | 35 |
| **6N** | 25 | 10 | 35 |
| **7N** | 0 | 35 | 35 |
| **1Y** | 0 | 1 | 1 |
| **2Y** | 0 | 1 | 1 |
| **4Y** | 0 | 1 | 1 |
| **5Y** | 0 | 1 | 1 |
| **6Y** | 0 | 1 | 1 |
| **Tot.** | 183 | 50 | 233 |
| $\mathbf{D}_{0.7,0.8}$ | 0.95 | 0.8 | $\mathbf{DP}_{0.7,0.8}$=0.87 |
| **E** | 2.57 | 1.38 | **OE**=2.32 |
| **P** | 0.19 | 0.7 | **OP**=0.3 |
| **OF** | - | - | 0.36 |
| **J** | - | - | 0.18 |
| **R** | - | - | 0.45 |
| **FM** | - | - | 0.41 |

cluster (a set of objects having similar behaviours $\simeq$ repetitive behaviours) and the factors influencing this situation are the *representative labels* of the cluster associated to it.

According to section 3.1.5 and if we consider the repartition of labels described in table 5, we have:

– For $C_1$:

- $MD_{0.7}(1N, C_1) = 1$
- $MD_{0.7}(2N, C_1) = 1$
- $MD_{0.7}(3N, C_1) = 1$
- $MD_{0.7}(4N, C_1) = 1$
- $MD_{0.7}(5N, C_1) = 1$
- $MD_{0.7}(6N, C_1) = 0.71$
- $D_{0.7,0.8}(C_1) = 0.95$.

Then, $1N$, $2N$, $3N$, $4N$, $5N$ and $6N$ are the representative labels of cluster $C_1$ and the factors influencing the traffic situation associated to $C_1$.

– For $C_2$:

- $MD_{0.7}(6N, C_2) = 0$
- $MD_{0.7}(7N, C_2) = 1$
- $MD_{0.7}(1Y, C_3) = 1$
- $MD_{0.7}(2Y, C_3) = 1$
- $MD_{0.7}(4Y, C_3) = 1$
- $MD_{0.7}(5Y, C_3) = 1$
- $MD_{0.7}(6Y, C_2) = 1$
- $D_{0.7,0.8}(C_2) = 0.8$.

Then, $7N$, $1Y$, $2Y$, $4Y$, $5Y$ and $6Y$ are the representative labels of the cluster $C_2$ and the factors influencing the traffic situation associated to $C_2$.

The identification of the factors influencing a traffic situation allow experts to make decisions such as:

- Forecasting: the fact that each cluster is described with distinct labels permits to conclude that each object having a label $l_j$ must be included in a cluster for which label $l_j$ is representative. This permits to predict the behaviour of any object.

- Detection of outliers and atypical situations: this operation is the consequence of the forecasting step. Indeed, if we have a new object $O_r$ described by a label $l_j$ that is more close to the centroid of a cluster $C_i$ than the other centroids and if the label $l_j$ is not representative of cluster $C_i$, then the object $O_r$ is considered as an outlier or an atypical traffic situation. For instance, considering the results of table 5, if the flow values of a new object are recorded a sunday and if this object is more close to the centroid of the cluster $C_1$ than to the one of $C_2$, this means that there was a traffic problem like an accident that occurred near the sensor that collected these flow values. Such situations can also be detected in real time if we consider the values of an objects as a stream data and treat them as soon as we collect them.

The precision of forecasting and/or detection of outliers are very related to the values of $\alpha$ and $\beta$ since these parameters reflect the quality degree of clustering interpretation the user wants to have. A high degree of precision is so assured by high values of $\alpha$ and $\beta$ (values close to 1).

## 6. CONCLUSION

In this paper, we presented a new evaluation measure called Homogeneity degree that permit to identify the factors influencing road traffic situations. The measure is based on class labels and aims to validate only interpretable clusters. We showed that our measure is better than the existing ones such as F-measure or Jaccard coefficient since the latter are not always able to identify interpretable results. As future directions, we intend to apply our approach to other domains like biology or medicine and to propose a new algorithm allowing us to use the homogeneity degree as relative measure to choose the best clustering results.

## 7. REFERENCES


[1] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal Process.*, 83(4):825–833, 2003.

[2] F. Boutin and M. Hascoet. Cluster validity indices for graph partitioning. In *Proc. of the Conference on Information Visualization*, pages 376–381, Londin, UK, 2004.

[3] N. Chinchor. Muc-4 evaluation metrics. In *Proceedings of the 4th conference on Message understanding (MUC4 '92)*, pages 22–29, 1992.

[4] D.-L. Davies and D.-W. Bouldin. Cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(2):224–227, 1979.

[5] J.-C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetica*, 4:95–104, 1974.

[6] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[7] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Cluster validity methods: part i. *SIGMOD Rec.*, 31(2):40–45, 2002.

[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity checking methods: Part ii. *SIGMOD Record*, 31(3):19–27, 2002.

[9] J. Mcqueen. some methods for classification and analysis of multivariate observations. In *5th Berkeley Symp. on Math. Statistics and Probability*, pages 281–298, Berkley, USA, 1967.

[10] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20(1):53–65, 1987.

[11] S. Saitta, B. Raphael, and I. F. C. Smith. A Bounded Index for Cluster Validity. In *Machine Learning and Data Mining in Pattern Recognition*, pages 174–187, 2007.

[12] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[13] P.-N. Tan, M. Steinbach, and K. Kumar. *Introduction to Data Mining*. Pearson Addison Wesley, 2005.

[14] R. Tibshirani and G. Walther. Cluster validation by prediction strength. *Journal of Computational & Graphical Statistics*, 14(3):511–528, September 2005.

[15] C. J. van Rijsbergen. *Information Retrieval (2nd ed.)*. Butterworth, 1979.

# SkyTree: Scalable Skyline Computation for Sensor Data

Jongwuk Lee, Seung-won Hwang
Pohang University of Science and Technology
Pohang, Republic of Korea
{julee, swhwang}@postech.ac.kr

## ABSTRACT

Skyline queries have gained attention for supporting multi-criteria analysis of large-scale datasets. While a lot of skyline algorithms have been proposed, most of the algorithms build upon pre-computed index structures that cannot generally be supported over sensor data of dynamically changing attribute values. We aim to design a scalable non-index skyline computation algorithm for sensor data. More specifically, we propose Algorithm SkyTree constructing a *dynamic lattice* that divides a specific region into several subregions based on a *pivot point* maximizing *dominance region*. Such structure enables to perform region-wise dominance tests, which eliminates unnecessary point-wise dominance tests. In addition, we ensure the progressiveness that has not been supported by any non-index algorithm, where we can identify $k$ points maximizing the sum of dominance regions as the greedy approximation method. The $k$ points are used to reduce *communication cost* between sensors in computing global skyline. Our evaluation results validate the efficiency of Algorithm SkyTree, both in terms of response time and communication overhead, over existing algorithms.

## 1. INTRODUCTION

Skyline queries have gained attention as an alternative operator for multi-criteria analysis of large-scale datasets, due to intuitive query formulation. For a multi-dimensional dataset, the skyline queries return a subset of "interesting" points that are no worse than, or not dominated by, any other points. If a point $p$ is better than another point $q$ in at least one dimension and not worse than $q$ in all other dimensions, it is said that $p$ *dominates* $q$.

To illustrate traditional skyline queries with static data, we describe an example with a new SQL expression, *i.e.*, `SKYLINE OF` [5, 6].

Example 1 (Skyline queries) Consider a hotel retrieval system using a database such that Hotel(*hno*, *name*, *price*, *star*, *city*). To find interesting hotels with cheap price and high star on city 'Paris', a user can formulate a skyline query as follows:

```
SELECT * FROM Hotel
WHERE city = 'Paris'
SKYLINE OF price MIN, star MAX;                    (Q1)
```

This paper studies skyline computation for sensor data in which the query processing consists of two parts. First, each sensor computes *local skyline* itself, where data are horizontally distributed. Second, we then take into account how to merge local skyline to identify *global skyline* over entire sensors. Each challenging issue has been mainly studied in *centralized* and *distributed* skyline computation respectively. We thus aim to optimize two key costs– (1) the *response time* in computing the local skyline at each sensor, and (2) the *communication cost* between sensors in computing the global skyline.

We first consider how to compute local skyline for each sensor. Existing solutions have mainly focused on improving skyline computation using pre-computed auxiliary structures such as a *bitmap* [20], *sorted list* [7, 8], *R-tree* [12, 18], and a new index to access points in *Z order* [14]. These structures are essential in ensuring the efficiency of computation, by pre-partitioning data into regions and enabling block-level dominance tests. For instance, as *R-tree* partitions datasets into minimum-bounding rectangles, multiple data points in each rectangle can be pruned in batch, from a single dominance test (*e.g.*, between a skyline point and a bounding data point of the rectangle). Without such pre-partitioning, dominance tests are done on point pairs and at most one point can be pruned from each test. However, it is infeasible to assume that such structures are maintained in the sensors, as values of data are constantly and dynamically changing, *e.g.*, *humidity* and *temperature*.

Our goal is thus to study skyline computation algorithms requiring no index, yet enabling batch pruning. While non-index algorithms have been studied [5, 10, 19], all these algorithms require to compare a skyline point with all other skyline points to ensure the correctness, *i.e.*, incurring $O(s)$ tests for each skyline point, when the number of skyline points is $s$. However, since $s$ tends to increase exponentially over dimensionality, *i.e.*, "curse of dimensionality", such cost critically hurts the scalability of algorithms.

In a clear contrast, we propose a novel algorithm, not requir-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09,* June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00

ing index, yet achieving efficiency by reducing point-based dominance tests between skylines. To achieve this goal, *pivot points* are dynamically selected, with respect to which the entire region are partitioned into disjoint subregions. The partitioned structure is abstracted as a *dynamic lattice* on which we define the *partial dominance* or *incomparability* notions. We perform *block-level* dominance tests on the lattice that reduce pair-wise dominance tests. Our evaluation results show that Algorithm SkyTree is significantly more efficient than existing non-index algorithms over all experimental settings.

An additional advantage of Algorithm SkyTree is to identify the skyline in a *progressive* manner. Progressiveness is studied as a desirable property in [12, 18], requiring to ensure the following two properties:

1. **Online skyline qualification:** The first skyline can be returned before all the other skylines are identified.
2. **Sublinear skyline qualification:** Returning a skyline point does not require to compare it with all other skyline points.

All existing non-index algorithms compare a skyline point with all other skyline points before qualifying it as a skyline and leave optimizing such skyline qualification cost as an open problem. As a result, no such algorithm is progressive, failing to ensure both [5, 10] or the first property [19] listed above.

In a clear contrast, Algorithm SkyTree does block-level tests on the lattice. It enables to instantly qualify a skyline point by only comparing with some skyline points in "promising" regions. This property allows Algorithm SkyTree to progressively produce more results early on, without compromising correctness.

This progressiveness of our proposed algorithm, in addition to reduce *response time* by decreasing pairwise dominance comparisons as described above, also contributes to reduce *communication cost* between sensors. Specifically, we propose a *pivoting strategy* that greedily find $k$ *representative skyline points* [15], to reduce both response time and communication cost, by maximizing dominated local points in each sensor. However, as the problem of finding the exact $k$ representative skyline points is known to be NP-hard in 3 or higher dimensional space [15], our proposed solution can be considered as a good practical approximation, as we also empirically validate in Section 5.

To summarize, we believe that this paper has the following contributions:

- We study the problem of supporting skyline computation over sensor data in scalable and progressive manners.
- We design Algorithm SkyTree, enabling block-level dominance tests on a lattice structure dynamically generated from a selective pivot point.
- We validate the efficiency of Algorithm SkyTree in terms of both response time and communication cost, over existing non-index skyline algorithms.

The rest of this paper is organized as follows. Section 2 reviews existing skyline algorithms and Section 3 discusses preliminaries on general skyline problem. Section 4 proposes Algorithm SkyTree based on a dynamic lattice. Section 5 shows experimental results and Section 6 presents our remarkable conclusions.

# 2. RELATED WORK

This section surveys existing skyline computation algorithms. Skyline queries have been first studied as *maximal vectors* in [13, 3, 4] which developed algorithms with better time complexity based on theoretical analysis. To estimate the cost of skyline computation, [3, 9, 6] has been formulated the size of skyline as a probabilistic equation. Later, starting from [5], many skyline algorithms have been studied in database community.

In particular, we present related algorithms in two categories:

- Centralized algorithms: These algorithms assume data are centrally stored and aim to minimize computation costs.
- Distributed algorithms: These algorithms assume data are distributed over nodes and aim to minimize communication cost across nodes.

## 2.1 Centralized Skyline Algorithms

We first survey existing centralized algorithms, which can be divided into those using indices and those not. Though our problem assumes distributed sensor environments, existing centralized algorithms, aiming to minimize response time, can inspire an efficient local skyline computation within sensor. In particular, sensor environments, with values constantly and dynamically changing, are more relevant to non-index algorithms.

### 2.1.1 Algorithms with Indices

This section presents existing skyline algorithm using auxiliary structures. Specifically, Tan et al. [20] first proposed Bitmap and Index using bitmap and B-tree indices to support progressive skyline computation. Chomicki et al. [7, 8] proposed SFS by using a *sorted list* of points that helps to reduce point-wise dominance comparisons. Kossmann et al. [12] and Papadias et al. [18] employed an *R-tree* index that facilitates *block-level* dominance comparisons without direct point-level access on the disk. Recently, Lee et al. [14] proposed a new index to access points in *Z order*. These algorithms, by leveraging pre-materialized structures, have limited applicability to (1) queries with complex Boolean constraints such as "join" or "group by" or (2) dynamically changing data as in sensors.

### 2.1.2 Algorithms without Indices

This section presents existing non-index skyline algorithms. Specifically, Börzsönyi et al. [5] proposed basic skyline computation methods, BNL and D&C. Later, Godfrey et al.

[10] developed LESS with attractive average-case asymptotic complexity in external memory environment. Recently, Park et al. [19] proposed SSkyline, reducing the dominance comparisons between non-skyline points by sequentially rearranging the order of accessing points. For a special case of dealing with low cardinality dataset, Morse et al. [17] proposed skyline computation using a static lattice structure. Since these algorithms do not essentially require any indices, skyline queries can support sensor data with dynamically changing values. However, typically lack efficiency and scalability, as we will further discuss in Section 4.

## 2.2 Distributed Skyline Algorithms

Centralized algorithms, not considering communication overhead across sensors, incur prohibitive communication cost when naively adopted to our problem. Recent research efforts studied algorithms for minimizing communication cost over vertically or horizontally distributed data sources, as we survey as follows.

### 2.2.1 Algorithms on Vertically Distributed Data

In typical Web scenarios, data are often vertically distributed, *e.g.*, *price* information residing in hotels.com and *distance* in maps.com, as typically assumed in [1, 16]. Balke et al. [1] first proposed distributed skyline computation reducing unnecessary data access cost. Further, Lo et al. [16] proposed progressive skyline computation under the same model which is based on a good "terminating object" using a *linear regression* technique for early termination.

### 2.2.2 Algorithms on Horizontally Distributed Data

Sensors can be viewed as dynamically changing horizontally distributed data. Existing algorithms, assuming static data, typically leverage indices [11, 2] to identify local skylines. Specifically, Hose et al. [11] proposed an algorithm finding relaxed skyline results on Peer Data Management Systems (PDMS). Although these initial results are not guaranteed to be global skyline, these reduce the communication cost across sensors, by pruning out dominated objects from local sensors. Recently, Bartolini et al. [2] proposed a faster algorithm SaLSa using a minC sorted list, which outperforms SFS by communicating promising result subset to prune out local objects and avoid fetching unnecessary data. With the same goal, our proposed algorithm approximately generates $k$ representative skyline points that is used to eliminate unnecessary fetches across sensors.

# 3. PRELIMINARIES

We first introduce basic notations to present the skyline problem. Let $\mathcal{D}$ be a finite $d$-dimensional data space such as $\{d_1, d_2, ..., d_d\}$, where each dimension has a domain of real numbers. Let $dom(\mathcal{D})$ be a domain of total data space, *i.e.*, $dom(\mathcal{D}) = dom(d_1) \times \ldots \times dom(d_d)$. Let $\mathcal{S}$ be an entire dataset with finite $n$ points, which is a subset of $dom(\mathcal{D})$, *i.e.*, $\mathcal{S} \subseteq dom(\mathcal{D})$. A point $p$ in $\mathcal{S}$ is represented as $(p_1, \ldots p_d)$, where $\forall i (1 \leq i \leq n) : p_i \in dom(d_i)$.

Based on these notations, we formally state *dominance* and *skyline*, respectively. Throughout this paper, we use *min* operator for skyline queries. These definitions are consistent with the skyline literature.

Definition 1 (Dominance) Given $p$, $q \in \mathcal{S}$, $p$ *dominates* $q$ on $\mathcal{D}$, denoted as $p \prec_{\mathcal{D}} q$, if and only if $\forall\, i \in [1, d]\ p_i \leq q_i$ and $\exists\, j \in [1, d] : p_j < q_j$.

Definition 2 (Skyline) A point $p \in \mathcal{S}$ is a *skyline point* on $\mathcal{D}$ if and only if any other point $\forall q (\neq p) \in \mathcal{S}$ does not dominate $p$ on $\mathcal{D}$. The skyline is a set of skyline points on $\mathcal{D}$ in $\mathcal{S}$, denoted as $\text{SKY}_{\mathcal{D}}(\mathcal{S})$.

The cost of skyline algorithms heavily depends on the number of dominance comparisons between points. Given a dataset $\mathcal{S}$ with $n$ points, skyline algorithms can perform up to $\frac{n(n-1)}{2}$ pairwise dominance tests. To design an efficient algorithm, we have to study how to reduce such dominance computations as much as possible.

Toward the goal, we use the following properties of skyline computation in [14].

Proposition 1 (Transitivity) Given, $p$, $q$, $r \in \mathcal{S}$ on $\mathcal{D}$, if $p \prec_{\mathcal{D}} q$ and $q \prec_{\mathcal{D}} r$, then $p \prec_{\mathcal{D}} r$.

Proposition 2 (Incomparability) Given $p$, $q \in \mathcal{S}$, if $p \not\prec_{\mathcal{D}} q$ and $q \not\prec_{\mathcal{D}} p$, $p$ and $q$ are *incomparable* on $\mathcal{D}$, denoted as $p \sim_{\mathcal{D}} q$.

For simplicity, we replace notations $\prec_{\mathcal{D}}$, $\not\prec_{\mathcal{D}}$, $\sim_{\mathcal{D}}$, and $\text{SKY}_{\mathcal{D}}(\mathcal{S})$ into $\prec$, $\not\prec$, $\sim$, and $\text{SKY}(\mathcal{S})$ if it is clear from the context.

We now present how existing algorithms exploit these properties for reducing dominance tests.

- **Reducing tests on non-skyline using dominance:** The number of dominance tests on a non-skyline point $p$ can vary from 1 (when $p$ is dominated from the first test) to $n-1$ (when dominated from the last tests). It is thus important to order dominance tests such that non-skylines are first compared with some point $q$ that is highly likely to dominate all non-skyline points. This inspires the intuition used in a sorting-based algorithm [7, 8] ordering data points in *topological order* by a monotonic scoring function, ensuring that if $q \prec p$, $q$ is guaranteed to rank higher in the topological order. By performing dominance tests with the highest ranking point $q$ with non-skylines, we can expect to prune non-skylines with very few tests. All existing non-index algorithms focus on optimizing these costs.
- **Reducing tests on skylines using incomparability:** However, incomparability is also helpful for reducing dominance tests between skylines. Typically, index-based skyline algorithms exploit them, by prepartitioning datasets and detecting incomparability relationships between partitions. Suppose that a hyper-rectangle subregion $\mathcal{R}$ on $\mathcal{D}$ is represented as $[u_1, v_1) \times \ldots \times [u_d, v_d)$, where the virtual best point and the virtual worst points are $\mathcal{R}^{best} = (u_1, \ldots, u_d)$ and $\mathcal{R}^{worst} = (v_1, \ldots, v_d)$ respectively. Given two subregions $\mathcal{R}_i$ and $\mathcal{R}_j$, they are *incomparable*, if $\mathcal{R}_i$ and $\mathcal{R}_j$ are not overlapped, $\mathcal{R}_i^{best} \not\prec \mathcal{R}_j^{worst}$, and $\mathcal{R}_j^{best} \not\prec \mathcal{R}_i^{worst}$. In such case, the points corresponding to each subregion are

also incomparable, *i.e.*, if $\mathcal{R}_i \sim \mathcal{R}_j$, then $\forall p \in \mathcal{R}_i, \forall q \in \mathcal{R}_j : p \sim q$. The property has inspired a divide-and-conquer approach [12, 18, 14] which first partitions an entire region $\mathcal{D}$ into several subregions, then computes the skyline for each subregion independently. As this reduction requires pre-partitioning, existing non-index algorithms do not optimize for this cost, while our proposed algorithm, through effective dynamic partitioning, exploits this reduction as well.

# 4. ALGORITHM SkyTree

This section presents our proposed algorithm not requiring pre-computed indices. In this work, we assume memory size is large enough to keep local skylines and focus on minimizing CPU cost. We first present how to partition a specific region into disjoint subregions in which *region-level* relationships are defined with respect to a point $p$. Observe that the partitioned subregions are abstracted as a *lattice* with the relationships such as *partial dominance* or *incomparability*. This structure enables *block-level* tests to reduce point-wise comparisons. We then explain how to select such desirable pivot point. Specifically, a *pivot point* $p^V$ dynamically constructs the lattice maximizing the size of *dominance regions*, *i.e.*, the volume of regions dominated by the pivot point. As recursively extending the structure, we sequentially identify skyline points, where the total sum of dominance regions are greedily maximized. In particular, the procedure generates the skyline as a tree structure, where each node is a pivot point selected from the subregion. In this aspect, we call our Algorithm SkyTree.

## 4.1 Constructing a Dynamic Lattice

Given a point $p$, each point $q$ in $\mathcal{S}$ can be mapped into a $d$-dimensional binary vector. Suppose that the value on $d_i$ corresponds to the $i^{th}$ least significant bit. If point $q$ is smaller than $p$ on $d_i$, the binary value is 0. Otherwise, the binary value is 1. Let $\mathcal{B}$ be a set of all possible $d$-dimensional binary vectors mapped from points, where the size of $\mathcal{B}$ is $2^d$, denoted as $\mathcal{B} = \{B_0, \ldots, B_{2^d-1}\}$.

Observe that each vector corresponds to a disjoint region partitioned from point $p$. Suppose that the size of dimensions $d$ is 3, where $p$ divides 3-dimensional space into 8 regions such that $\mathcal{B} = \{B_0 = 000, B_1 = 001, \ldots, B_7 = 111\}$. When the binary value on $d_i$ is 0, the possible values of points in a region are represented as the range $[0, p_i)$. Otherwise, the range is $[p_i, 1]$. For example, vectors $B_0$ and $B_1$ correspond to regions such as $[0, p_3) \times [0, p_2) \times [0, p_1)$ and $[0, p_3] \times [0, p_2) \times [p_1, 1]$ respectively. For the sake of representation, we denote a set of regions $\mathcal{R}$ mapped into vectors, *i.e.*, $\mathcal{R} = \{R_0, \ldots R_{2^d-1}\}$ such that $\forall B_i \in \mathcal{B} : B_i \rightarrow R_i$. A set of points mapped into the same vectors can be thus shown as a *block* presenting a region.

We now present the relationship between binary values, which in turn defines the relationships of the corresponding regions. For example, since '000' dominates '111' on all dimensions, points in $R_7$ can be safely pruned out if there exist any other points in $R_0$, *i.e.*, '000' *dominates* '111'. As the another case, since '001' and '011' are identical on $\{d_1, d_3\}$, points in $R_1$ may dominate points in $R_3$ on $\{d_1, d_3\}$, and vice versa. Meanwhile, '001' dominates '011' on $\{d_2\}$. In such case, we say that '001' *partially dominates* '011' on

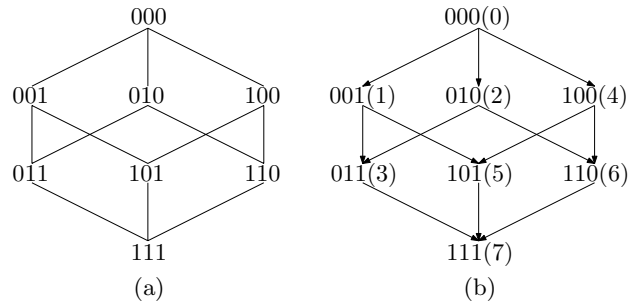



**Figure 1: (a) A binary lattice when $d$=3, (b) An encoded binary lattice with the partial dominance relationship**

$\{d_1, d_2, d_3\}$, which indicates that there may exist a point in $R_1$ dominating another point in $R_3$. As the other case, if two binary vectors exist dominating values at each side, *e.g.*, '001' and '010', we say that they are *incomparable*, which indicates that points with incomparable relationship can skip dominance tests without loss of correctness.

We formally state region-level relationships between binary vectors $\mathcal{B}$. Let $B.d_i$ is a binary value on $d_i$ of vector $B$.

Definition 3 (Dominance in $\mathcal{B}$) Given two vectors $B$ and $B'$, $B$ *dominates* $B'$ on $\mathcal{D}$, denoted as $B \prec B'$, if and only if $\forall d_i \in \mathcal{D} : B.d_i < B'.d_i$.

Definition 4 (Partial dominance in $\mathcal{B}$) Given two vectors $B$ and $B'$, $B$ *partially dominates* $B'$ on $\mathcal{D}$, denoted as $B \prec_{Par} B'$, if and only if $\forall d_i \in \mathcal{D} : B.d_i \leq B'.d_i$.

Definition 5 (Incomparability in $\mathcal{B}$) Given two vectors $B$ and $B'$, $B$ is *incomparable* with $B'$ on $\mathcal{D}$, if and only if $\exists d_i \in \mathcal{D} : B.d_i < B'.d_i$ and $\exists d_j \in \mathcal{D} : B'.d_j < B.d_j$.

The vectors in $\mathcal{B}$ can thus be organized as a *partially ordered set*. To illustrate, when $d = 3$, Figure 1(a) describes a binary *lattice*. The partial ordered set is based on the partial dominance relationships between vectors, described as arrows. We can exploit these relationships to reduce dominance tests as follows:

- **Dominance tests on dominance regions**: If there exist any other points in $B_0$, we can prune out all points in $B_{2^d-1}$. Observe that, when a point $p$ divides the region into $2^d$ subregions, $p$ is inherently contained in $B_0$. This implies that point $p$ dominates all points in $B_{2^d-1}$, requiring only 1 dominance test.
- **Dominance tests on partial dominance regions**: If there exists an arrow from $B_i$ to $B_j$, $B_i$ partially dominates $B_j$. This relationship indicates the relationships between point sets in the corresponding regions $R_i$ and $R_j$, which we denote as $S_i$ and $S_j$ respectively. As points in $R_i$ are likely to dominate those in $R_j$, we need to perform dominance tests between $S_i$ and $S_j$, *i.e.*, $\forall p \in S_i, q \in S_j : p \prec q$. To reduce unnecessary dominance tests, we first identify skyline in $S_i$, *i.e.*,

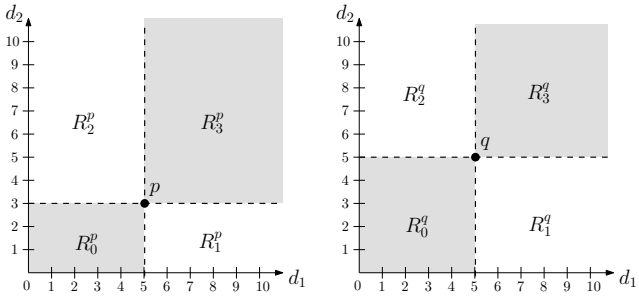



**Figure 2: Dominance regions on $p$ and $q$**

SKY($S_i$), based on which we reduce dominated points in $S_j$. After these tests, $S_j$ is reduced to $S'_j \subseteq S_j$, where we identify skyline SKY($S'_j$) for $S_j$.

Recall that, by the transitivity, partial dominance holds not only on directly connected pairs with an arrow but also on all "reachable" pairs in the lattice. That is, if there exists an arrow from $B_i$ to $B_j$ and also from $B_j$ to $B_k$, *i.e.*, if $B_i$ is reachable to $B_k$, $B_i$ also partially dominates $B_k$. we can thus use the same procedures to reduce dominance tests on the points in $R_i$ and $R_k$.

- **Dominance tests on incomparable regions**: For all the remaining pairs, $B_i$ and $B_j$ are incomparable. This implies that point sets $S_i$ and $S_j$ in corresponding regions $R_i$ and $R_j$ are also incomparable, *i.e.*, $\forall p \in S_i, q \in S_j : p \sim q$. We can thus skip these tests with known outcomes and focus on performing dominance tests on point pairs within $S_i$ (and $S_j$).

Summing up, the block-level tests described above enable to exploit dominance and incompatibility, to avoid unnecessary point-level dominance comparisons for both skylines and non-skyline points. In particular, by dynamically generating the lattice on-the-fly, Algorithm SkyTree can effectively support local skyline computation for sensor data with dynamically changing values, unlike existing work building upon pre-materialized indices.

## 4.2 Determining a Pivot Point

We now discuss how we can effectively divide space into regions, by presenting how we select a *pivot point* $p^V$ that determines the space division into $2^d$ subregions. Desirably, we should select an ideal pivot point reducing overall cost. More specifically, (1) How do we select $p^V$ to minimize response time to compute local skyline for each sensor? (2) How do we select $p^V$ to minimize communication cost to compute global skyline?

We first address the first question. To illustrate, consider two pivot points $p$ and $q$ in Figure 2. Figure 2(a) illustrates two regions with dominance with respect to $p$, *i.e.*, $R_0^p$ and $R_3^p$ (gray colors). Figure 2(b) similarly illustrates $R_0^q$ and $R_3^q$. Observe that, points in $R_3^p$ and $R_3^q$ are guaranteed to be pruned out after dominance tests with any point in $R_0^p$ and $R_0^q$ respectively. Assuming data points are uniformly distributed, it is desirable to select a point maximizing the dominated region. Assuming uniform distribution and attribute values normalized to (0, 1), the number of points in dominated region for dataset of size $n$ is $V = \prod_{i=1}^{n} (1 - p_i) \times n$. We can then select a pivot point $p^V$ as a point maximizing $V$. Note, by maximizing $V$, we consider all skylines as pivots first, before any non-skyline point, because when $p \prec q$, $R_3^p$ is guaranteed to include $R_3^q$. Due to this property, many prior literatures [7, 8, 10] similarly leveraged $V$ (or its variants) for optimization, though our work is the first to use it for space division.

**Algorithm 1 SelectPivotPoint**($\mathcal{S}$, $\mathcal{R}$, *tail*)

**Input**: A dataset $\mathcal{S} = \{ p^1, \ldots, p^n \}$
**Output**: A pivot point $p^V$

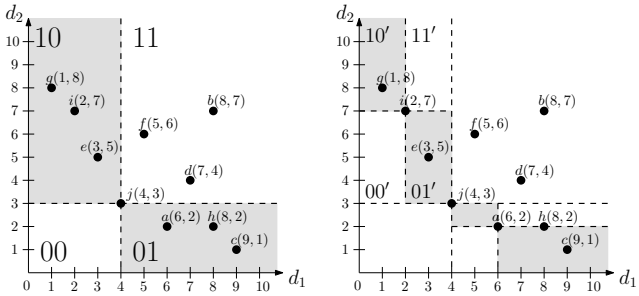
```
head ← 1, cur ← 2. // Initialize variables.
maxDG ← CalculateDG(S[head], R).
while cur ≤ tail do
   if S[head] ≺ S[cur] then
      S[cur] ← S[tail].
      tail ← tail − 1. // Move away a dominated point.
   else if S[cur] ≺ S[head] then
      S[head] ← S[cur]. // Change a dominated head point.
      S[cur] ← S[tail].
      tail ← tail − 1.
      maxDG ← CalculateDG(S[head], R).
   else
      // Compare maxDG with curDG of S[cur].
      curDG ← CalculateDG(S[cur], R).
      if curDG > maxDG then
         Swap between S[cur] and S[head].
         curDG ← maxDG.
      end if
      cur ← cur + 1.
   end if
end while
return S[head].
```

We then address the second question of minimizing communication cost in aggregating local skyline results from sensors. Toward the goal, existing work [2] studies sharing initial results across sensors, to eliminate dominated points from other sensors early on, for saving overall communication cost. More specifically, for such initial results, $k$ initial results that dominate as many points as possible from each sensor are desirable, where $k < |\text{SKY}(\mathcal{S})|$. In a different problem context, Lin et al. [15] studied to identify $k$ *representative skyline*, maximizing the number of dominating points as follows:

Problem 1 ($k$ representative skyline) Given a dataset $\mathcal{S}$ and an integer $k$, identify a set $RS$ of $k$ skyline points maximizing the number of dominating points in $\mathcal{S}$ such that $|\{ (p, q) \mid p \prec q, p \in RS, q \in \mathcal{S}\}|$.

While we can adopt the same intuition to identify the best initial results for eliminating dominated points from other sensors, the exact solution of identifying $k$ representative skyline points is known as NP-hard [15] in 3 and higher dimensional space.

We thus consider a greedy approximation for this problem. Fortunately, our pivot selection, greedily selecting a point $p$ maximizing the dominance region in $R$, can be viewed as a greedy approximation. In other words, we can communicate the first $k$ pivots across sensors, to maximally eliminate dominated points in other sensors.




(a) $1^{st}$ division (b) $2^{nd}$ and $3^{rd}$ divisions

**Figure 3: A toy dataset on two dimensional space**

Algorithm 1 describes the pseudo code of our pivot selection for optimizing both response time and communication cost, where *tail* is a variable presenting the last position in $\mathcal{S}$. Specifically, we perform pairwise dominance tests and swapping, in order to put (1) skyline with (2) the biggest dominance region as the first point $p$. When we sequentially access points in $\mathcal{S}$, we perform dominance tests between $p$ and a current accessed point $q$. If $q$ is dominated, we replace $q$ into a point to be placed to *tail*. As the another case, if (1) $q$ dominates $p$ or (2) $q$ has bigger volume size than $p$, we replace a dominated point into a point in *tail* or swap a pair ($p$, $q$). The function **CalculateDG**($p$, $\mathcal{R}$) is to calculate a dominance region for $p$, *i.e.*, $\sum_{i=1}^{n} log(p_i^{max} - p_i)$ such that $p^{max}$ is a virtual point presenting maximum values in $R$.

## 4.3 Computing the Skyline

This section discusses how we use the lattice structure $\mathcal{B}$ (Figure 1) for the skyline computation. Essentially, our proposed skyline computation accesses nodes in a lattice in a topological order preserving partial dominance relationships. For points in each node, we can skip dominance tests against the points in other nodes with incomparable relationships, and focus on dominance tests with the points in nodes with dominance relationships.

More specifically, we describe this procedure as a pseudo code in Algorithm 2. In lines 4-5, we determines $p^V$ from the specific region $\mathcal{R}$. In lines 6-14, we compute the vector for each point, and allocate it into list $\mathcal{H}$ presenting the corresponding vectors. In an exceptional case, we store the points that are identical to $p^V$ into an auxiliary list $\mathcal{I}$ which is contained into the skyline result $\mathcal{L}$. In lines 15-27, we perform point-level dominance tests for blocks with the partial dominance relationship, eliminate dominated points for partially dominated blocks, and recursively call a function for points in the subregion.

To illustrate SkyTree, we show an example with 10 points on two dimensional space described as Figure 3. First, after scanning whole points, a pivot point is determined as $j$. As illustrated Figure 3(a), we generate 4 subregions, where $j$ is added into the skyline, and 3 points $b$, $d$, and $f$ are simultaneously eliminated. For the other points mapped into '01' and '10', we recursively find pivot points. For '01', pivot points are sequentially selected as $a$ and $c$, and added into the skyline. For '10', a pivot point is selected as $i$. Finally, the skyline are identified as $j$, $a$, $c$, $i$, $e$ and $g$. During the skyline computation, all points are compared with two or less times, as Figure 4 illustrates with a tree structure.

**Algorithm 2** SkyTree($\mathcal{S}$, $\mathcal{R}$, $\mathcal{L}$)

**Input**: A dataset $\mathcal{S} = \{ p^1, \ldots, p^n \}$
**Output**: A list of skyline points $\mathcal{L}$

1: $max \leftarrow (1 << d) - 2$. // *Set the size of a lattice.*
2: $\mathcal{I} \leftarrow \{\}$ // *Initialize the list with identical points to* $p^V$.
3: $\mathcal{H}[1, \ldots, max] \leftarrow \{\}$. // *Initialize the list of array* $\mathcal{H}$.
4: $tail \leftarrow |\mathcal{S}|$. // *Mark the last position of* $\mathcal{S}$.
5: $p^V \leftarrow$ **SelectPivotPoint**($\mathcal{S}$, $\mathcal{R}$, $tail$).
6: **for** $\forall\ p \in \mathcal{S}[1, \ldots, tail]$ **do**
7: $\quad LID \leftarrow \mathcal{M}(p, p^V)$ // $\mathcal{M}$ *maps* $p$ *into an encoded value.*
8: $\quad$ **if** $LID < max$ **then**
9: $\quad\quad \mathcal{H}[LID]$.Add($p$). // *Generate the list of each LID.*
10: $\quad$ **else if** $p = p^V$ **then**
11: $\quad\quad \mathcal{I}$.Add($p$). // *Store points identical to* $p^V$.
12: $\quad$ **end if**
13: **end for**
14: $\mathcal{L}$.AddRange($\mathcal{I}$).
15: **for** $LID \leftarrow 1$ to $max$ **do**
16: $\quad$ **if** $\mathcal{H}[LID]$.Size() $> 0$ **then**
17: $\quad\quad$ **for** $\forall\ DID \in \mathcal{B}$: $DID \prec_{Par} LID$ **do**
18: $\quad\quad\quad$ **EliminateDominatedPoints**($\mathcal{H}[DID]$, $\mathcal{H}[LID]$).
19: $\quad\quad$ **end for**
20: $\quad\quad$ **if** $\mathcal{H}[LID]$.Size() $> 0$ **then**
21: $\quad\quad\quad$ // *Set value ranges in* $R_{LID}$ *corresponding to* $LID$.
22: $\quad\quad\quad R_{LID} \leftarrow [p_1^{min}, p_1^{max}] \times \ldots \times [p_d^{min}, p_d^{max}]$.
23: $\quad\quad\quad \mathcal{T} \leftarrow$ SkyTree($\mathcal{H}[LID], R_{LID}, \mathcal{L}$).
24: $\quad\quad\quad \mathcal{L}$.AddRange($\mathcal{T}$).
25: $\quad\quad$ **end if**
26: $\quad$ **end if**
27: **end for**

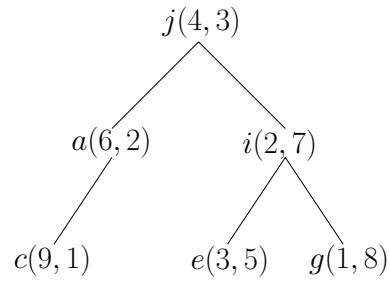
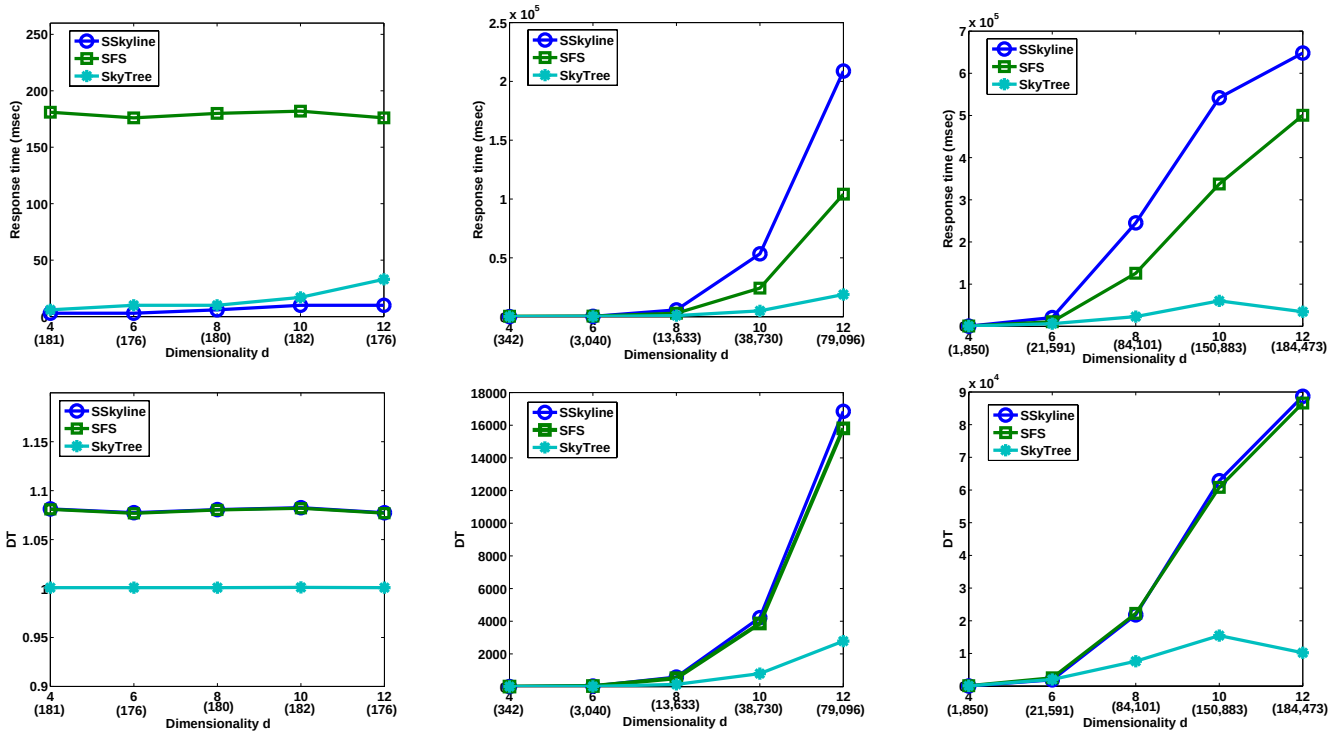



**Figure 4: A skyline tree structure in Figure 3**

This skyline tree structure can also be used to efficiently compute our approximation for $k$ representative. More specifically, when the required size of representative skyline $k$ is 3, we start the traversal from the root node and enqueue $j$ in Figure 4 into a priority queue, keeping elements sorted in the order of the volume $V$ of dominating region. We then iteratively pop the top element and enqueue its child nodes– For instance, when the current top node $j$ is dequeued, its child nodes $a$ and $i$ are enqueued. This iteration continues until $k$ points are popped, *e.g.*, $k$ representative skyline points in Figure 4 are $j$, $a$, and $i$.

# 5. EXPERIMENTS

This section presents empirical evaluation results for Algorithm SkyTree. We explain experimental settings (Section 5.1), and validate the efficiency in terms of response time for local skyline computation (Section 5.2) and communication cost to identify global skyline results (Section 5.3).




(a) Correlated (b) Independent (c) Anti-correlated

**Figure 5: Varying $d$**

## 5.1 Experimental Settings

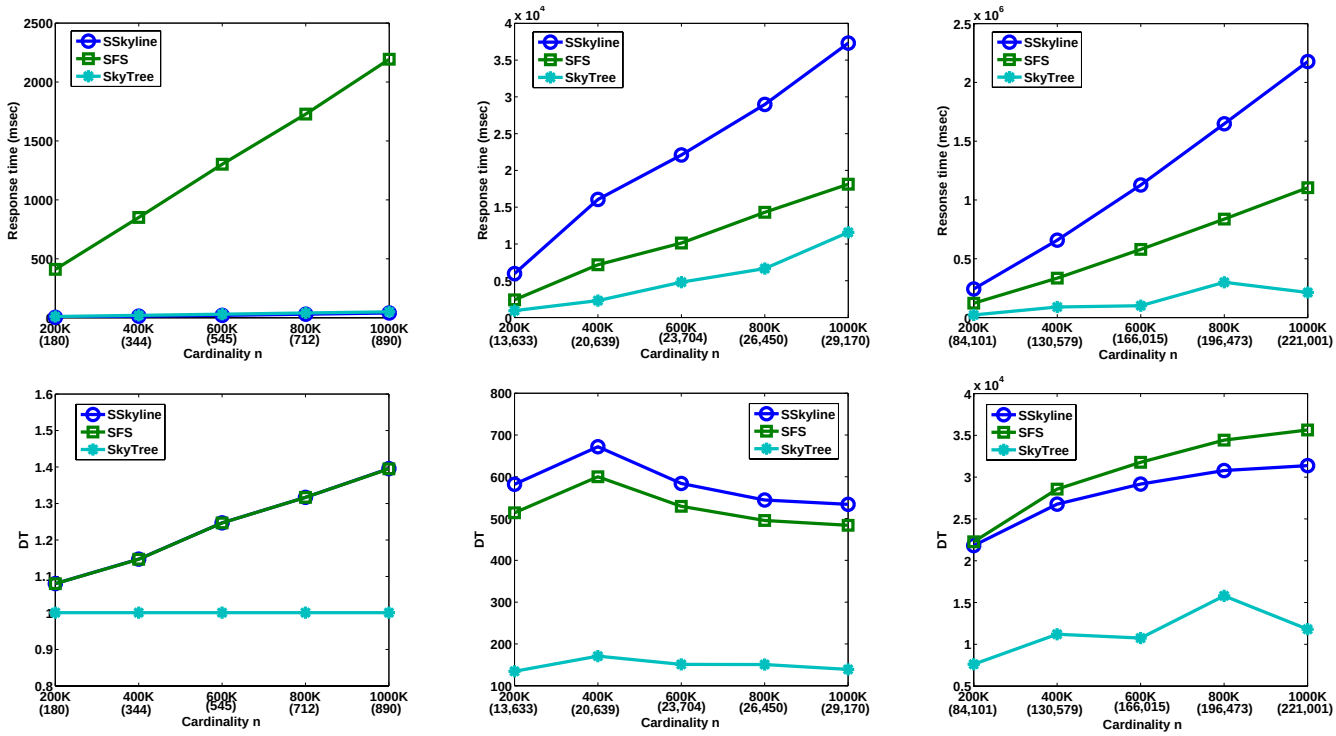
All the algorithms are implemented in C++ language, assuming that window size for skyline computation is unbound. All experiments were conducted on Windows XP with Intel Core Duo 2.66 GHz CPU and 2GB RAM.

To validate in extensive settings, we synthetically generated various datasets, varying three parameters– distribution, cardinality, and dimensionality. All attributes are real numbers in (0, 1). More specifically:

- Distribution: We generate three datasets, *i.e.*, **Correlated**, **Independent**, and **Anti-correlated**, following the data generation instructions presented in [5].
- Dimensionality $d$: We vary dimensionality $d$ from 4 to 12 (Default: $d = 8$)
- Cardinality $n$: We vary cardinality $n$ from $200K$ to $1000K$. (Default: $n = 200K$)

With these datasets, we compare our proposed algorithm with baselines, using the following two cost metrics:

- Computation cost for local results: We first validate the efficiency of our algorithm in identifying local skylines from a single sensor, by comparing against general non-index skyline algorithms below as baselines. As a cost metric, we measure response time and the number of dominance tets.
  - Algorithm SFS [7]: To compare with a general class of non-index algorithms, we implement SFS where objects are pre-sorted in a required topological order for free. While this assumption is not practical and unfavorable to us, our intention of comparing with this baseline is to validate the efficiency of our algorithm even in these unfavorable settings.
  - Algorithm SSkyline [19]: This algorithm is reported to outperform all non-index algorithms.
- Communication cost for global results: We then validate the efficiency of our algorithm in aggregating local results and identify global skylines, by comparing against distributed skyline computation algorithms optimizing for communication costs, as baselines.
  - Algorithm SFS [7]: We can use SFS to identify $k$ initial results using the volume of dominance region as a sorting function and communicate these results across sensors.
  - Algorithm SaLSa$^V$ [2]: Algorithm SaLSa improves SFS by using a sorted index based on minC function. However, we observe that, while minC function is desirable for pruning out non-skylines, it is less desirable for maintaining skylines. We thus implement SaLSa$^V$ ordering intermediate skylines in the order of the volume $V$ of dominating regions.




(a) Correlated (b) Independent (c) Anti-correlated

**Figure 6: Varying $n$**

## 5.2 Local Skyline Computation

This section extensively evaluates the efficiency of Algorithm SkyTree in computing local skyline computation over varying dimensionality and cardinality. To validate the performance, we leverage two measure such as *response time* and *the number of dominance tests* ($DT$) per a point [2].

### 5.2.1 *The Effect of Dimensionality*

We evaluate the effect of dimensionality in computing local skyline as illustrated Figure 5. As references, we also state the number of skyline points of each dataset (corresponding to values in the $x$-axis) below– For instance, for an anti-correlated dataset in Figure 5(c), when $d = 4$, we generated has 1,850 skyline points, which increases to 184,473 when $d$ increases to 12. Existing non-index algorithms are significantly affected by such growths in skyline points over increasing $d$, as they do not optimize quadratic comparison between skyline point pairs. In a clear contrast, Algorithm SkyTree is highly scalable over $d$ and thus significantly outperforms existing algorithms, especially when the dimensionality is high. For instance, Algorithm SkyTree returns skyline results in less than 60 seconds in all datasets, while it may take more than 600 seconds for SSkyline (*i.e.*, 10 times speedup). This drastic speedup of Algorithm SkyTree is achieved by considering both dominance and incomparability relationships between regions in the lattice, based on which we reduce unnecessary point-wise dominance tests both for non-skyline and skyline points. We observe consistently for $DT$, where SkyTree incurs three times less dominance tests compared to SSkyline.

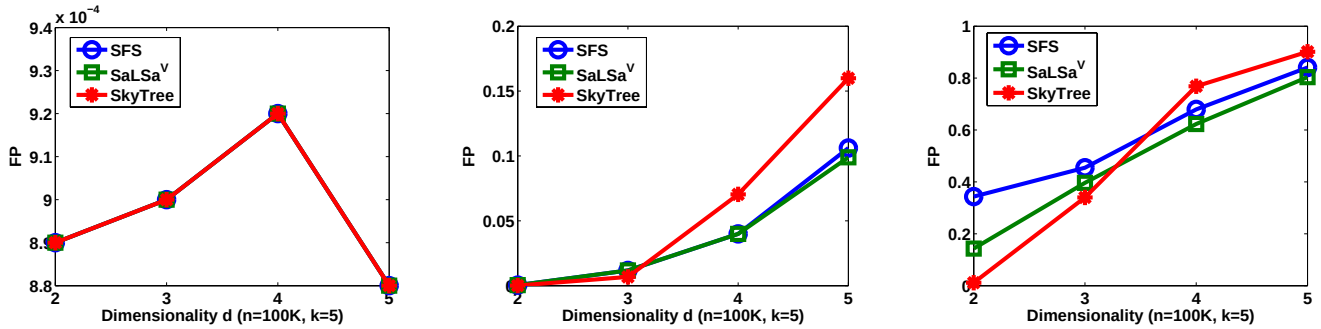
### 5.2.2 *The Effect of Cardinality*

We evaluate the effect of cardinality in computing local skyline as illustrated Figure 6. Similar to the dimensionality, as the cardinality increases, the number of both skyline and non-skyline points also increases. That is, Algorithm SkyTree, using both dominance and incomparability relationships to reduce dominance tests for both skylines and non-skylines, has larger room for improvements for high cardinality dataset. As a result, our proposed algorithm significantly outperforms all other algorithms in all experimental settings, especially when $n$ is high– For instance, when $n = 1000K$ on anti-correlated dataset, Algorithm SkyTree achieves up to 10 times speedup than Algorithm SSkyline, *i.e.*, SkyTree = 213 *sec* and SSkyline = 2177 *sec*.

## 5.3 Global Skyline Computation

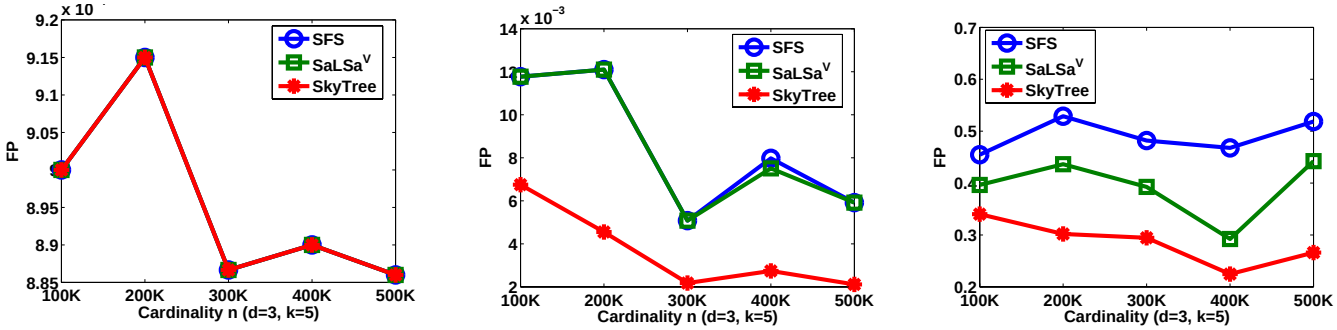
This section presents the efficiency of Algorithm SkyTree in computing global skyline computation over varying dimensionality and cardinality. As a cost metric, we use a *fetched points* ($FP$) [2]. Given $k$ initial representative skyline points $RS$ from one sensor, we measure the ratio of the objects that are not dominated by these initial results, from each of the remaining sensors. In other words, the lower the $FP$ is, the less the communication cost is. More formally,

$$FP(RS, \mathcal{S}_i) = 1 - \frac{|(p,q)|p \prec q, p \in RS, q \in \mathcal{S}_i\}|}{|\mathcal{S}_i|}.$$




(a) Correlated (b) Independent (c) Anti-correlated

**Figure 7: Varying $d$**




(a) Correlated (b) Independent (c) Anti-correlated

**Figure 8: Varying $n$**

### 5.3.1 *The Effect of Dimensionality*

Figure 7 describes the effect of dimensionality for global skyline computation. Observe that, for independent and anti-correlated datasets, Algorithm SkyTree outperforms other algorithms in low dimensional space. For instance, in two dimensional space, Algorithm SFS and SaLSa$^V$ incur 12 and 30 times more communication overhead over Algorithm SkyTree respectively. The cost gap closes for high-dimensional data, as the accuracy of our greedy approximation decreases, as the number of skylines exponentially increases over increasing $d$. However, considering our approximation incurs much less computational overhead, our proposed approach still has advantage over existing algorithms.

### 5.3.2 *The Effect of Cardinality*

Figure 8 describes the effect of cardinality for global skyline computation. Observe that $FP$ decreases as the cardinality increases and so does the number of skylines. However, for all cardinality settings in all datasets, Algorithm SkyTree consistently outperforms other algorithms. In particular, the $FP$ of SFS and SaLSa$^V$ is up to 2 and 3 times higher than Algorithm SkyTree in the independent distribution.

## 6. CONCLUSION

This paper studied to support scalable skyline computation over sensor data. Towards this goal, we designed Algorithm SkyTree requiring no pre-computed index, but instead building a dynamic lattice and exploiting block-level dominance tests. In particular, we studied how to improve the overall efficiency in terms of both response time and communication cost. Our evaluation results validated that Algorithm SkyTree significantly outperforms existing algorithms.

## 7. REFERENCES


[1] W. Balke, U. Güntzer, and J. X. Zheng. Efficient distributed skylining for web information systems. In *EDBT*, 2004.

[2] I. Bartolini, P. Ciaccia, and M. Patella. Efficient sort-based skyline evaluation. *ACM TODS*, 2008.

[3] J. Bentley, H. Kung, M. Schkolnick, and C. Thompson. On the average number of maxima in a set of vectors and applications. In *JACM*, 1978.

[4] J. L. Bentley, K. L. Clarkson, and D. B. Levine. Fast linear expected-time algorithms for computing maxima and convex hulls. In *SODA*, 1991.

[5] S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *ICDE*, 2001.

[6] S. Chaudhuri, N. Dalvi, and R. Kaushik. Robust cardinality and cost estimation for skyline operator. In *ICDE*, 2006.

[7] J. Chomicki, P. Godfery, J. Gryz, and D. Liang. Skyline with presorting. In *ICDE*, 2003.

[8] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with presorting: Theory and optimizations. In *Intelligent Information Systems*, 2005.

[9] P. Godfrey. Skyline cardinality for relational

processing. In *Foundations of Information and Knowledge Systems*, 2004.

[10] P. Godfrey, R. Shipley, and J. Gryz. Maximal vector computation in large data sets. In *VLDB*, 2005.

[11] K. Hose, C. Lemke, and K. Sattler. Processing relaxed skylines in pdms using distributed data summaries. In *CIKM*, 2006.

[12] D. Kossmann, F. Ramsak, and S. Rost. Shooting stars in the sky: An online algorithm for skyline queries. In *VLDB*, 2002.

[13] H. T. Kung, F. Luccio, and F. Preparata. On finding the maxima of a set of vectors. *JACM*, 1975.

[14] K. C. Lee, B. Zheng, H. Li, and W.-C. Lee. Approaching the skyline in Z order. In *VLDB*, 2007.

[15] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang. Selecting stars: The k most representative skyline operator. In *ICDE*, 2007.

[16] E. Lo, K. Y. Yip, K.-I. Lin, and D. W. Cheung. Progressive skylining over web-accessible database. *Data & Knowledge Enginnering*, 2006.

[17] M. Morse, J. M. Patel, and H. Jagadish. Efficient skyline computation over low-cardinailty domains. In *VLDB*, 2007.

[18] D. Papadias, Y. Tao, G. Fu, and B. Seeger. An optimal and progessive algorithm for skyline queries. In *SIGMOD*, 2003.

[19] S. Park, T. Kim, J. Park, J. Kim, and H. Im. Parallel skyline computation on multicore architectures. In *ICDE*, 2009.

[20] K. Tan, P. Eng, and B. C. Ooi. Efficient progressive skyline computation. In *VLDB*, 2001.

# Clustering of Power Quality Event Data Collected via Monitoring Systems Installed on the Electricity Network

Mennan Güder[*,†], Nihan Kesim Çiçekli[†], Özgül Salor*, Işık Çadırcı*

* Power Electronics Group,
TUBITAK-UZAY, Ankara, Turkey
*{mennan.guder, ozgul.salor*

Hacettepe University, Ankara, Turkey
*isik.cadirci}@uzay.tubitak.gov.tr*

[†] Department of Computer Engineering,
Middle East Technical University, Ankara, Turkey
*nihan@ceng.metu.edu.tr*

## ABSTRACT
In this paper, a k-means-based clustering method applied to power quality event data is described. The data are collected by the power quality (PQ) monitors, which are developed through the National PQ Project and installed on the electricity network. The PQ monitors detect the PQ events defined as voltage sags, swells, and interruptions by the IEC Standard 61000-4-30, and collect the raw data of the event. The proposed method aims to cope with the huge event data size and cluster the event types so that PQ events are ultimately classified. The method helps to manage the event data to come up with PQ assessments for the specific measurement points and to make comparisons of various measurement points in terms of PQ of the electricity network.

## Categories and Subject Descriptors
H.2.8 [**DATABASE MANAGEMENT**]: Database Applications – *data mining.*

## General Terms
Algorithms, Management, Measurement, Design.

## Keywords
Power Quality, Power Quality Event, k-Means Clustering, Data Mining.

## 1. INTRODUCTION
With the increased use of costly investments on delicate power electronics devices, the classification of power quality (PQ) disturbances (or events) gains importance. Power quality events occurring most frequently are voltage sags, swells, and interruptions which may even cause shut down of processes run by electronics devices. Therefore, it is important to detect and classify PQ events occurring on a specific site to take

countermeasures against the potential PQ events.

Various methods have been proposed for classification of power quality events using data mining techniques [1-3]. In [1], automatic segmentation is applied on the three year simultaneous measurement data of three phase harmonic current and voltages at seven measurement points. In [2], data processing by using a phase corrected wavelet transform and classifications of very short duration transient power quality disturbance patterns are presented [2]. In [3], results of clustering software ACPro is used to identify the classes in the harmonic data and link analysis is applied to merge the obtained clusters into super-groups.

Other methods based on signal processing techniques have also been proposed [4-10]. In [4], covariance behaviour of several features derived from the event data is used for PQ event detection and classification. Classification of PQ events such as harmonics, sags, and capacitor switching is achieved using time-frequency analysis of the voltage and current waveforms in [5]. Neural networks have been used by [6] for PQ disturbance classification, while fuzzy-expert systems are used by [7] for the same purpose. Wavelets are used in [8-10] for PQ event classification. In these types of systems [4-10], PQ events are characterized by several features and these features are classified for single phase voltages.

The proposed clustering method proposed in this paper focuses on the root-mean-square (rms) values of the voltage waveform for clustering taken every half-cycle of the 50 Hz power signal as recommended in the IEC Standard 61000-4-30 [11] for all of the three phases. The novelty of this approach is that the clustering is achieved on three-phase voltage rms data. This helps to post-classify the clusters according to the power system events. The clustered units are 3-second rms voltages of the event data of three phases forming a (3x300) matrix. PQ events are detected and the event data are collected by the PQ Monitors developed through the National Power Quality Project [12-14]. Clustering is achieved on the rms event data matrices.

"Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SensorKDD'09, June 28, 2009, Paris, France.

Copyright 2009 ACM 978-1-60558-668-7...$5.00."

The following sections present the data collection process, the proposed clustering method, results, and the conclusions.

## 2. PQ MONITORING SYSTEM

A brief description of the PQ monitoring system, which has been used to detect PQ events and collect corresponding data, is given in this section. Figure 1 illustrates the PQ monitoring system developed through the National PQ Project of Turkey [12-14]. PQ monitors are now connected to 300 measurement points of the electricity transmission system all over the country. PQ events (voltage sags, swells and interruptions) are detected by the PQ monitors and three-second raw data, which are three-phase voltages and currents. They are immediately sent to the National PQ Monitoring Center (NPQMC) and stored in the database as given in Fig. 1. The PQ monitors have been developed through the National PQ Project of Turkey and all PQ parameter computations and event detections are achieved according to the IEC Standard [11]. The raw data of the events are sampled by 512 samples per cycle (i.e. 25600 Samples per second). The event data collection starts 0.5 seconds before the start of an event and ends after 2.5 seconds. The raw data for each event (voltages and currents for three phases) stored in NPQMC is 3.5 MB and the data format is binary. The database at the NPQMC can also be reached and event searches can be inserted from remote PCs through the developed Client Software as given in Fig. 1.

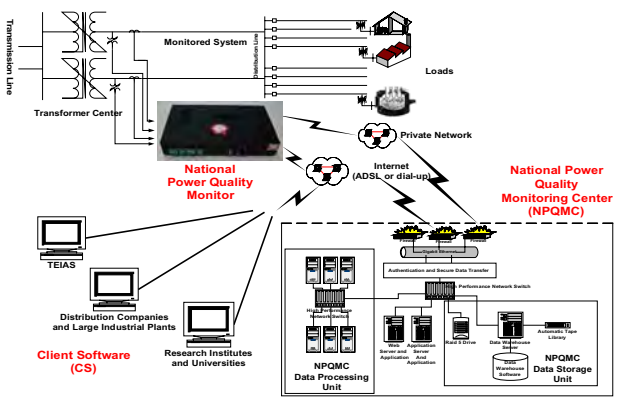



**Figure 1**. System Architecture of the PQ Monitoring System.

## 3. SYSTEM DESCRIPTION

The block diagram of the proposed PQ event clustering algorithm is given in Fig 2.

The event data collected for a specific measurement point are downloaded from the NPQMC, including three-phase voltages of sag, swell and interruption events sampled at a sampling rate of 25.6 kS/s. Each event data package includes three-second data. Before k-means clustering, data matrices are prepared. Rms values of the three-phase voltages computed for 512-sample windows shifted by 256 samples are obtained for each event data package as recommended in [11]. This process results in rms voltage matrices of size (3x300). Then a normalization process is applied so that all the rms voltages are represented as the percentages of their nominal value. These matrices are then clustered by k-means algorithm. The distance measure used for the k-means algorithm is the Euclidean distance between the matrices. The matrix presentations of the events 1 and 2 can be given as,

$$M1 = \begin{bmatrix} V1_A[0] & V1_A[1] & V1_A[2] & \cdots & V1_A[299] \\ V1_B[0] & V1_B[1] & V1_B[2] & \cdots & V1_B[299] \\ V1_C[0] & V1_C[1] & V1_C[2] & \cdots & V1_C[299] \end{bmatrix}$$

, (1)

$$M2 = \begin{bmatrix} V2_A[0] & V2_A[1] & V2_A[2] & \cdots & V2_A[299] \\ V2_B[0] & V2_B[1] & V2_B[2] & \cdots & V2_B[299] \\ V2_C[0] & V2_C[1] & V2_C[2] & \cdots & V2_C[299] \end{bmatrix}$$

. (2)

The distance function which calculates the distance between events 1 and 2 is defined as,

$$d_{1,2} = \sum_{i=A}^{C} \Big\{ \sum_{j=0}^{49} \left(V1_i[j] - V2_i[j]\right)^2 + K \sum_{j=50}^{149} \left(V1_i[j] - V2_i[j]\right)^2 + \sum_{j=151}^{299} \left(V1_i[j] - V2_i[j]\right)^2 \Big\}$$

(3)

where $K$ is the bias coefficient of the distance measure, which emphasizes the one-second part of the event data starting from the 0.5$^{th}$ second. This biasing, which is for K>1, is applied to obtain a clustering more dependent on first one-second part of the event. The square root of the distance value is decreased by one precision in order to prevent the small but continuous differences from being dominant during the cluster determination in the k-means algorithm.

The clustering method used in this paper is a multistep version of Lloyd's algorithm [15] with the efficiency and domain specific modifications. Since Lloyd's algorithm may result in clusters far from the optimal, the modifications are needed to ensure better clustering. The detailed block diagram of the proposed method is given in Fig. 3. Lloyd's algorithm is applied as the first step in the clustering method as given in Fig. 3. Afterwards the total mean square distance between the centers of each class and the matrices belonging to that class are computed ($E_i$). The classes with the mean error larger than the mean of $E_i$ ($E_{mean}$) are then refined. This refining process results in either the elimination of the center data of that cluster or the subdivision of the cluster into two clusters so that the mean error of each new class is less than $E_{mean}$. Once the refining is done to finalize the clusters, rule-based clustering is achieved to obtain event classes representing specific classes such as motor starting, line-to-line fault, line-to-neutral fault, capacitor switching, and etc.

**Figure 2**. Data Flow Diagram of the Proposed Method.

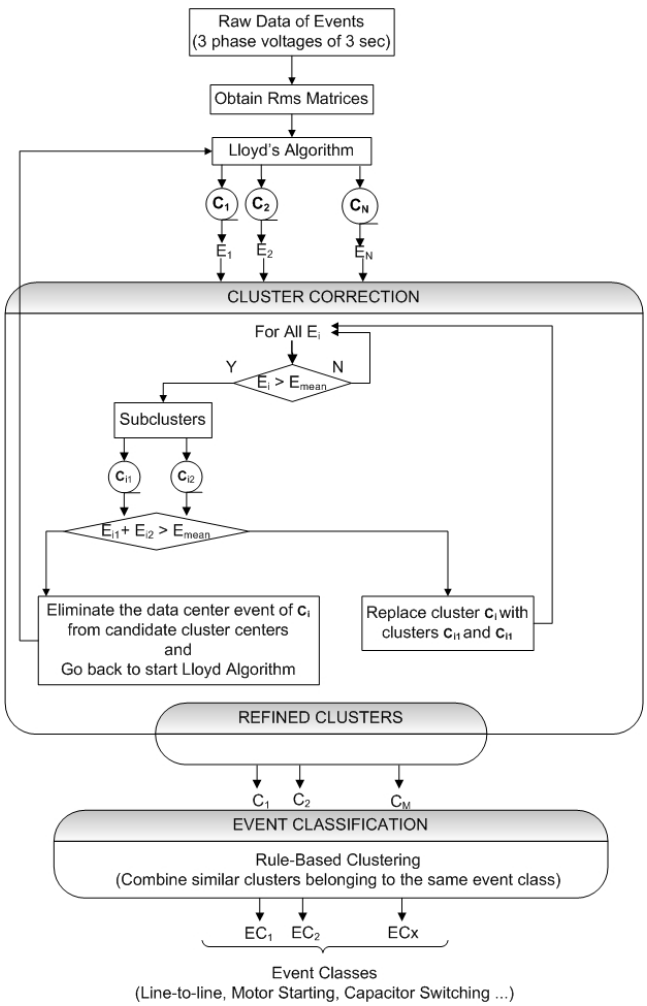
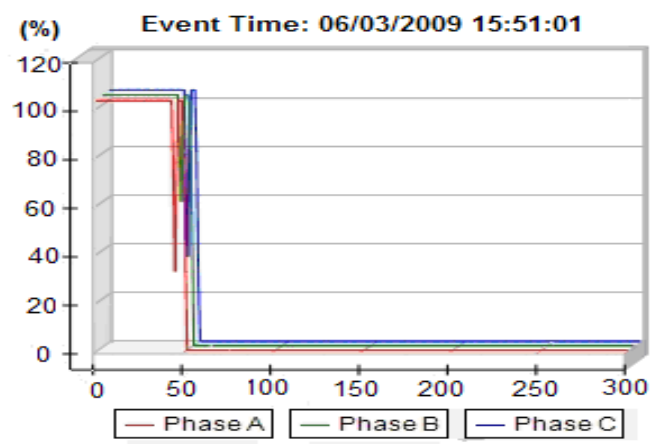


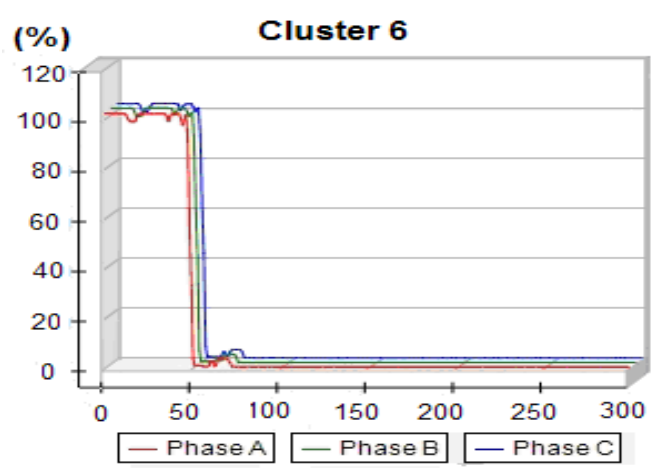

**Figure 3**. Block Diagram of the Proposed Method.

An example for the raw data of a voltage interruption, the rms data obtained to form the (3x300) matrix and the cluster which it belongs to as a result of the cluster refining are given in Fig. 4.

(a)




(b)




(c)

**Figure 4.** Raw data of the voltage interruption event (a), rms values of the event (b), and cluster that the event belongs to (c).

## 4. PQ EVENT CLUSTERING RESULTS AND DISCUSSION

In this section, some examples of the event clusters which are determined as the basic fault classes as a result of the rule-based clustering are presented. The power quality terms "sag", "swell", and "interruption" are used for voltage variations in the classification. Sag is the drop in the voltage level from the

nominal value to below 90 % of it, while a swell event is the voltage increase above 110 % of the nominal value. Interruptions are more serious voltage drops, which are below 10 % of the nominal [11].

The cluster number 8 ($C_8$) is a typical cluster for line-to-ground fault in Phase-A as observed in Fig. 5. A sag event of approximately 0.5 seconds is observed in Phase-A, and swells are observed in Phase-B and Phase-C. This is the common behavior of the transmission system during a line-to-ground fault. The voltage level for the sag is approximately 45%, which shows that the fault does not occur at exactly the measurement point, but in its neighborhood, because the voltage level is not decreased to 0%.

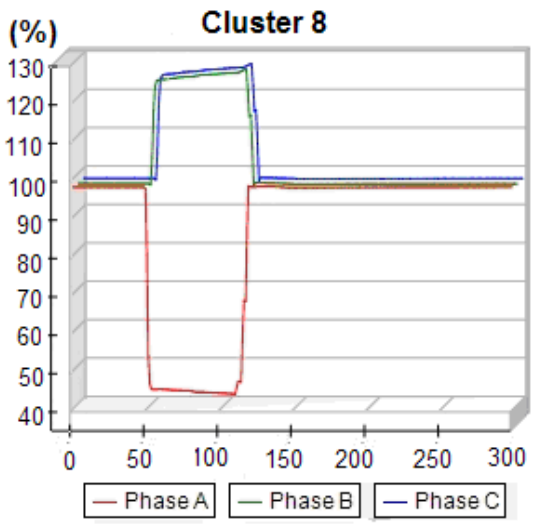



**Figure 5**. Cluster Number 8. A typical class for line-to-ground fault at Phase-A.

In Fig. 4.c , Cluster 6 ($C_6$), which is a typical case of an interruption in all phases, is illustrated. This can be the beginning of either a short term interruption or a sustained interruption according to the IEEE Standard [16] depending on the duration of the event.

Cluster 7 ($C_7$) is shown in Fig. 6. As observed, this is a sag event in all three phases. Since the duration of the event is very short (approximately one cycle of the 50 Hz power signal), it can be defined as an "instantaneous" sag event according to [16]. It is observed that after the recovery of the event another voltage drop is observed but it is immediately recovered before it reaches to a sag level.

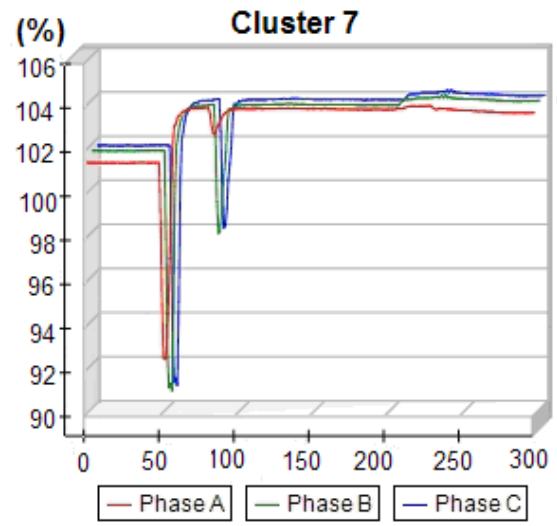



**Figure 6.** Cluster Number 7. A typical instantaneous sag event in all three phases.

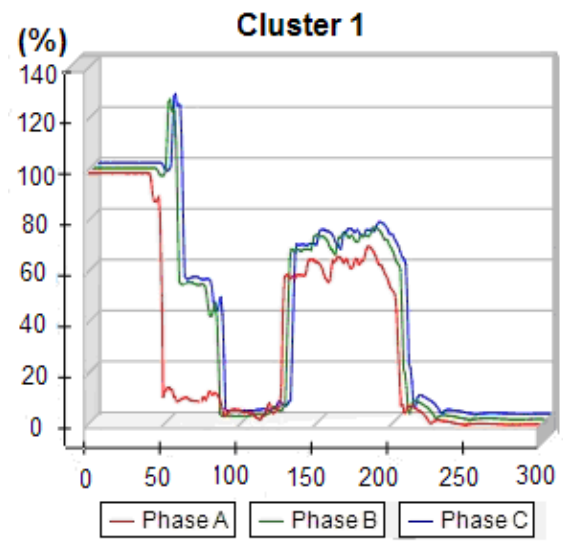



**Figure 7.** Interruption in Phase-A followed by instantaneous swell, sag and interruption in Phase-B and Phase-C.

In Fig. 7, Cluster 1 ($C_1$) is shown. This cluster is among the most interesting clusters, since it includes various consecutive events in all phases. The event starts with an interruption in Phase-A, followed by instantaneous swells, sags, and interruptions in Phase-B and Phase-C. Then a recovery attempt is observed in the power system, in which voltages of all phases raise up to approximately 70 % of the nominal voltage; however, the attempt comes out to be unsuccessful and finally all phases end up with an interruption level.

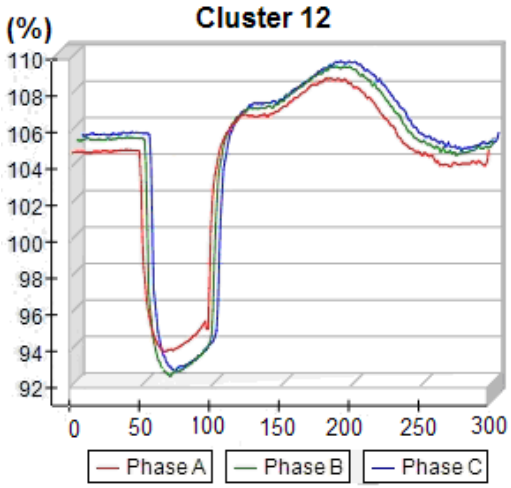



**Figure 8.** Sag event in all three phases.

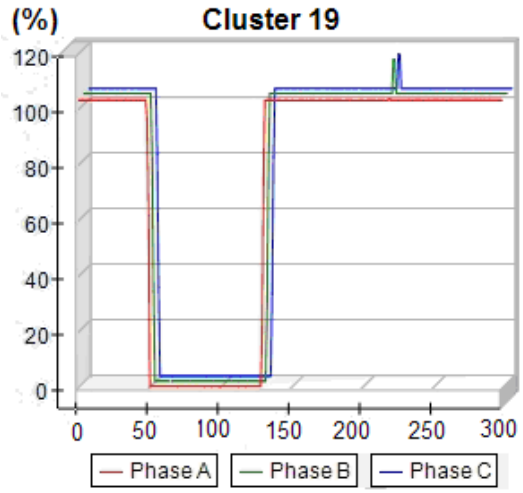



**Figure 9.** Voltage interruption in all three phases.

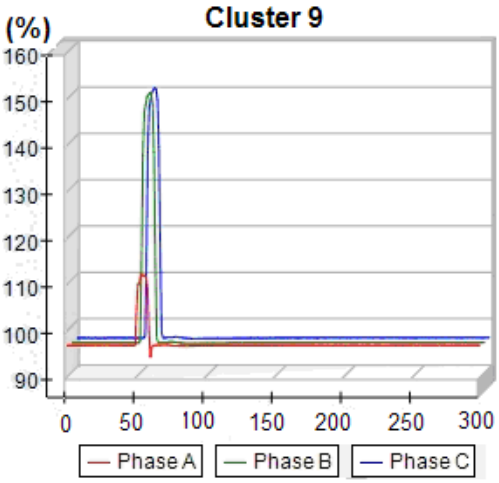
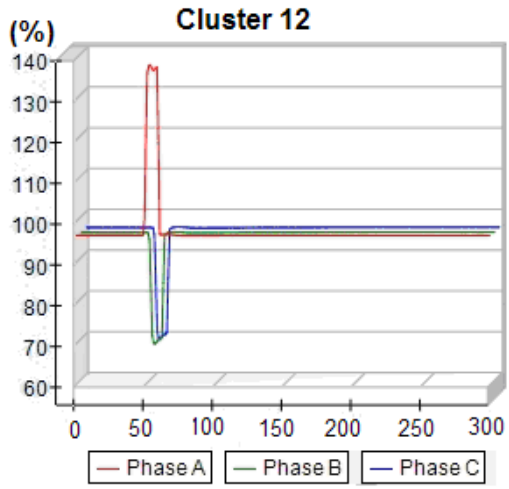
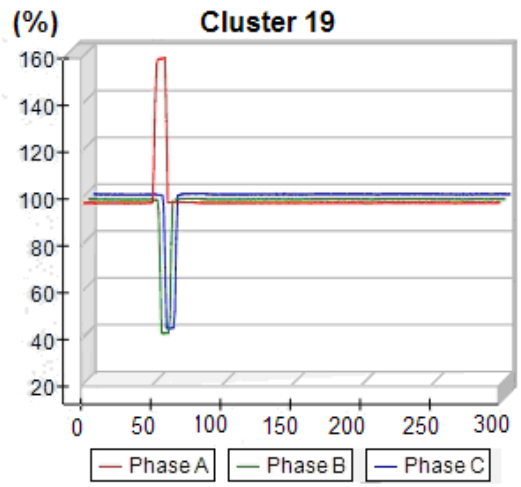



**Figure 10**. Voltage swells in all phases.




(a)




(b)

**Figure 11**. Voltage swells in Phase-A and sags in Phase-B and Phase-C.

Cluster 12 ($C_{12}$) in Fig. 8 consists of swells in all three phases, which correspond to start of a high power consuming device. In Fig. 10, Cluster 19 ($C_{19}$) represents an interruption in all three phases. In Fig. 11, Cluster 9 ($C_9$) is a typical example for instantaneous swells of all phases.

The clusters shown in Figure-5 to Figure-10 were all for a total cluster number of 20. Almost all of the clusters represent a specific power quality event class. When the number of clusters is increased to 30, on the other hand, some clusters belong to the same type of event classes while these similar clusters give clues about the distance from the occurrence of the event. In Fig. 11, such an example is illustrated for the total cluster number of 30. Cluster 12 ($C_{12}$) and Cluster 19 ($C_{19}$), in (a) and (b), represent sag in Phase-A and swells in the other phases. The event cluster in (b) represents events occurring at a point further than the point in which the events represented by the cluster in (a) occurs, because the sag and swell levels are lower in Cluster 12 ($C_{12}$), which means that the effect of the sags and

swells have less influence on the quality of power at the measurement point.

## 5. CLUSTER REFINING AND ERROR COMPUTATION

The clusters which have inner error value higher than the mean error are refined as described in the Fig.3. As observed from the comparison of Fig.12 and Fig.13, eight new clusters are formed, and both the overall mean error and the individual error values are decreased when the number of clusters is increased as expected. Lower inner clustering error makes the clustering analysis easier. However correlation between the events in the same cluster decreases as the refinement become deeper.

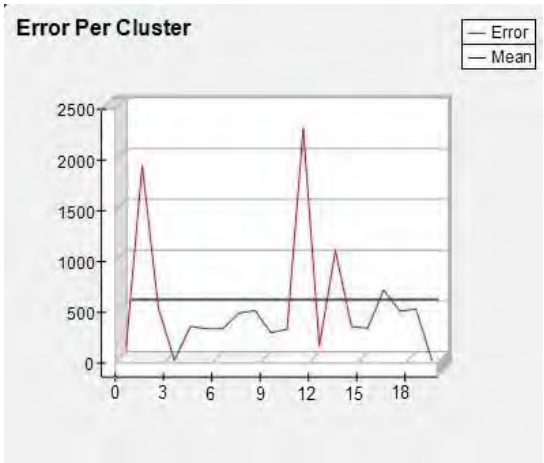



**Figure 12.** Error results of clustering applied on Alibeykoy Transformer substation event data without error based cluster refinement

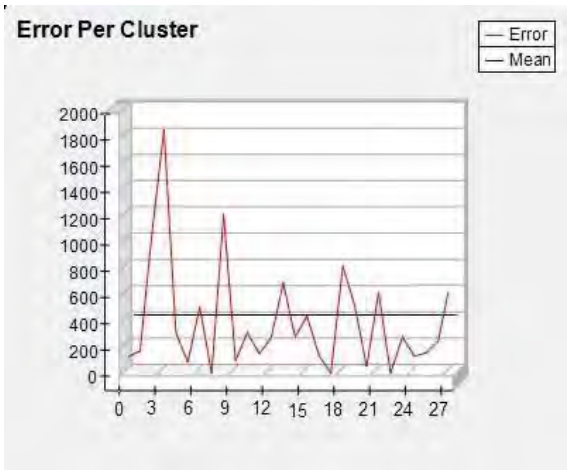



**Figure 13.** Error results of clustering applied on Alibeykoy Transformer substation event data with error based cluster refinement

## 6. CONCLUSIONS

In this paper, a k-means-based clustering method applied to power quality event data is described. The data are collected by the power quality (PQ) monitors, which are developed through the National PQ Project and installed on the electricity network. It has been shown by the proposed algorithm that PQ events of three phases can be clustered together to obtain meaningful clusters in terms of PQ. As the number clusters are increased, the clusters with similar events start to appear; however, those similar clusters represent the same type of events occurring at different locations of the power system, which can be preferred depending on the application. The clustering has been applied to data collected at five distributed transformer substations. The proposed algorithm will be applied to 300 measurement points being monitored currently by the National Power Quality Project of Turkey as future work. As a result, a generalized PQ event clustering will be obtained for the whole transmission system to obtain a characteristic event distribution in the country. Current waveforms in addition to voltage waveforms of the events will also be considered for a better classification of the PQ events.

## 7. ACKNOWLEDGMENTS

This research and technology development work is carried out as a subproject of the National Power Quality Project of Turkey. Authors would like to thank the Public Research Support Group (KAMAG) of the TÜBİTAK for full financial support of the project.

## 8. REFERENCES

[1] A. Asheibi, D. Stirling, D. Soetanto, "Analyzing Harmonic Monitoring Data Using Data Mining", 5th Australian Data Mining Conference, AusDM, 2006.

[2] P. K. Dash, I. L.W. Chun, M. V. Chilukuri, "Power Quality Data Mining Using Soft Computing and Wavelet Transform", Conference on Convergent Technologies for Asia-Pacific Region, TENCON 2003.

[3] A. Asheibi, D. Stirling, D. Robinson, "Identification of Load Power Quality Characteristics Using Data Mining", IEEE CCECE/CCGEI, 2006.

[4] O. N. Gerek, D. G. Ece, A. Barkana, "Covariance Analysis of Voltage Waveform Signature for Power-Quality Event Classification", IEEE Trans. on Power Delivery, vol. 21, no. 4, Oct. 2006.

[5] M. Wang, G. I. Rowe, A. V. Mamishev, "Classification of Power Quality Events Using Optimal Time-Frequency Representations – Part 2: Application", IEEE Trans. on Power Delivery, vol. 19, no. 3, July 2003.

[6] M. Uyar, S. Yildirim, M. T. Gencoglu, "An expert system based on S-transform and neural network for automatic classification of power quality disturbances", Expert Systems with Applications, Elsevier, 2008.

[7] Y. Liao, J. B. Lee, "A fuzzy expert system for classifying power quality disturbances", Electrical Power and Energy Systems, vol. 26, pp. 199-205, 2004.

[8] G.S. Hu, F.F. Zhu, Z. Ren, "Power quality disturbance identification using wavelet packet entropy and weighted support vector machines", Expert Systems with Applications, vol. 35, pp. 143-149, 2008.

[9] M. Uyar, S. Yildirim, M. T. Gencoglu, "An effective wavelet-based feature extraction method for classification of power quality disturbance signals", Electric Power Systems Research, vol. 78, pp. 1747-1755, 2008.

[10] B. Biswal, P. K. dash, J. B. V. Reddy, "Power Signal Classification Using Dynamic Wavelet Network", Applied Soft Computing, vol. 9, pp. 118-1252009.

[11] IEC Standard 61000-4-30, Electromagnetic Compatibility (EMC) – Part 4-30: Testing and Measurement Techniques – power Quality Measurement Methods.

[12] T. Demirci, A. Kalaycioglu, Ö. Salor, et.al., "National PQ Monitoring Network for Turkish Electricity Transmission System", IEEE Conference on Instrumentation and Measurement Technology, IMTC 2007.

[13] http://www.guckalitesi.gen.tr

[14] Ö. Salor, S. Buhan, Ö. Ünsar, et.al., "Mobile Monitoring System to Take Nationwide PQ Measurements on Electricity Transmission Systems", the Measurement Journal of Elsevier, vol. 42, pp. 501-515, 2009.

[15] S. P. Lloyd, "Least Square Quantization in PCM", IEEE Trans. Information Theory, vol. 28, 129-137, 1982.

[16] IEEE Std 1159-1995, Recommended Practice for Monitoring Electric Power Quality.

# CHALLENGE PAPERS

# SENSORKDD'09 CUP

## Problem Description

For this inaugural SensorKDD Challenge, held in conjunction with the 3rd International Workshop on Knowledge Discovery from Sensor Data (SensorKD'09), we pose a general and open-ended problem related to the workshop theme *Climate Change, Energy Assurance, and Infrastructural Impacts*. Specifically, we focus on the first component and provide a large climate dataset consisting of daily global air temperature and precipitation measurements for a 50-year period (1950-1999).

The task is to detect significant changes in this data. Note that "change" in this context may be an abrupt change (i.e., an anomaly or outlier), a gradual change (i.e., a shift in distribution), an extreme event (i.e. heat wave, severe rainfall, drought) or something different altogether. We intentionally omit a precise definition here to invite creative problem definitions and solutions.

## Data

The data consists of daily global air temperature and precipitation measurements, aggregated from heterogeneous sensors, for the period from 1950-1999 (see [1] for details). This data is also publicly available for download, for example from [2] or [3] (registration may be required).

### References

[1] R. Kistler, et al. "The NCEP/NCAR 50-year reanalysis." Bulletin of the American Meteorological Society, 82(2), 247-267 (2001).

[2] http://www.cdc.noaa.gov/data/reanalysis/reanalysis.shtml

[3] http://dss.ucar.edu/datasets/ds090.0/

**The workshop organizers acknowledge the assistance of Dr. Auroop Ganguly, Prof. Nitesh Chawla, and Karsten Steinhaeuser in organizing the SensorKDD'09 Challenge.**

# Change detection in rainfall and temperature patterns over India

Shivam Tripathi and Rao S. Govindaraju
School of Civil Engineering, Purdue University
West Lafayette, Indiana, USA
govind@purdue.edu

## ABSTRACT

The changes in rainfall and temperature patterns over India were detected using Mann-Kendall trend test, Bayesian change point analysis, and a hidden Markov model. A regionalization method was developed to identify homogeneous regions that experience similar weather states. The regionalization helped in finding contiguous regions with strong change signals. The data were investigated at different temporal and spatial resolution to explore the nature of changes. The study found that all India summer monsoon is stable, but the winter or the north-east monsoon is gradually intensifying. It also detected an abrupt drop in the winter and spring temperature over north-central India and a gradual increase in the summer temperature over the peninsular India. Robustness of the detected changes were evaluated using recent reanalysis datasets.

## Categories and Subject Descriptors

G.3 [**Probability and statistics**]: Markov processes, probabilistic algorithms; I.5.3 [**Pattern recognition**]: Clustering—*algorithms, similarity measures*; J.2 [**Physical sciences and engineering**]: Earth and atmospheric sciences

## General Terms

Algorithms

## Keywords

Indian summer monsoon, trend, Bayesian change point analysis, hidden Markov models, Markov random fields

## 1. INTRODUCTION

Lenton et al. [15] found that Indian summer monsoon (ISM) is one of those nine climatic phenomenon whose disruption due to climate change will have a drastic impact on the climate over the entire globe. To predict the behavior of ISM in the scenario of possible climate change requires a thorough understanding of the changes it has undergone in the past. Recently, many studies have attempted to discover changes in different facets of Indian monsoon and ascribe causes to those changes [10, 13]. However, no definite conclusions have yet been reached because the climate change signals in ISM are confounded by the large inherent variability in the monsoon at intra-seasonal, annual, and decadal time steps. This study undertook the challenge of finding changes in the rainfall and temperature over India using the data provided in the SensorKDD-2009 Challenge. Data exploration and visualization methods were used to screen the regions having homogeneous characteristics and strong change signals. The screened regions were investigated for changes in the seasonal, monthly, and daily characteristics of rainfall and temperature using novel change detection methods.

The remainder of the paper is structured as follows: (i) the statistical tools used to discover changes are first outlined, (ii) the data used in the study are described, (iii) the methodology used to identify changes and the results obtained are discussed, and finally (iv) conclusions are presented.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD'09,* June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

## 2. STATISTICAL METHODS USED TO DETECT CHANGES

### 2.1 Mann-Kendall trend test

The Mann-Kendall trend test (MKT; [11, 17]) is among the most widely used test for detecting linear trend in hydrological time series. The test is based on the *order statistics* and is therefore less sensitive to the outliers. The test statistic in the MKT follows a standard normal distribution, therefore the significance of trends at a desired significance level can be evaluated by comparing its value with standard normal variate. Like other trend tests, the MKT assumes observations to be independent and identically distributed.

### 2.2 Bayesian change-point analysis

Change point detection algorithms have been independently developed in many branches of science and engineering including ecology [3], hydrology [18], and signal processing [14] to detect abrupt changes in sequential data. The underlying assumptions in most of these algorithms can be explained by using *product partition model* [2] in which sequence of observations $x_1, x_2, \ldots, x_N$ are assumed to be

divisible into non overlapping $K$ *product partitions* as

$$p(x_1, \ldots, x_N) = p_1(x_1, \ldots, x_{\rho_1}) p_2(x_{\rho_1+1}, \ldots, x_{\rho_2}) \cdots p_K(x_{\rho_{K-1}+1}, \ldots, x_N)$$

The delineations between the partitions are called *change points* [1]. The inference problem is to identify the number of change points $K$, location of change points $x_{\rho_k}, k = 1, \ldots, K-1$, and the parameters of distributions $p_k, k = 1, \ldots, K$.

The inference problem can be solved by using either Frequentist approaches [16] that includes recently developed methods like support vector machine [7] or Bayesian approaches [1]. This study uses the Bayesian method developed by Lavielle [14], which employs a *penalized contrast function* to detect abrupt changes in the sequential data.

The algorithm assigns uniform prior distribution to the change point sequences and estimates its posterior distribution using Markov chain Monte Carlo based sampling method [9]. The contrast function used in the study assumes data to be independent and Normally distributed, and enables detection of abrupt-changes in the mean, variance, or both properties of the data. The penalty term used to identify the number of change points $K$ is adaptively estimated by the algorithm. The Matlab implementation of the algorithm was obtained from the website `http://www.math.u-psud.fr/lavielle/programs`.

## 2.3 Autoregressive Hidden Markov model

The preceding algorithms to detect linear trend or abrupt changes assume data to be independent. The independence assumption can be justified for the monthly, seasonal, or annual statistics in the rainfall and temperature data. However, the data at daily time step have strong temporal dependence because a weather state associated with them typically persists for many days. Hence, detecting changes in the daily data requires careful modeling of the temporal structure which in this study is achieved by using *autoregressive hidden Markov model* (ARHMM) [8]. Tripathi and Govindaraju [22] presents the application of ARHMM for modeling Indian summer monsoon rainfall in detail. For completeness, the ARHMM model is briefly described below.

Let the rainfall or temperature at time $t$ be denoted by $x_t$, $t = 1, \ldots, N$ $\{x_t \in \Re$ and $\boldsymbol{x} = [x_1, \ldots, x_N]^{\mathrm{T}}\}$. In an ARHMM, the data $x_t$ are assumed to depend on - (i) the state variable $\boldsymbol{z}_t$ $\{\boldsymbol{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^{\mathrm{T}}\}$ that denotes state of the weather, is hidden (not observed) and follows the first order Markov property, and (ii) the data values at previous time steps $\breve{\boldsymbol{x}}_t = [x_{t-1}, \ldots, x_{t-L}]^{\mathrm{T}}$ where $L$ is the number of lags. To attain flexibility in modeling *state duration*, an expanded sate HMM with *type A* topology was used. The model parameters which are essentially the parameters of the emission distribution $p(x_t|\boldsymbol{z}_t, \breve{\boldsymbol{x}}_t)$, transition distribution $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$, and distribution of the initial state $p(\boldsymbol{z}_1)$ were estimated by using the *variational Bayes* method [4].

## 2.4 Markov random fields

It is often difficult to detect changes in the data over a single station or a grid point due to inherently large variability and/or sampling issues, even if significant changes are present in an extended homogeneous area [10]. It is therefore imperative to identify regions that are homogeneous in terms of data characteristics. In the literature, homogeneous groups are often delineated using physiographic make-up of the region. However, this criterion may not always be sufficient to uniquely identify a homogeneous region, and in such cases it is desirable to let the observed data guide the delineation process.

In this study, the aim of regionalization is to find groupings (or clusters) of grid points such that the durations and timings of the weather states in grid points within a group are similar. To achieve this aim, a metric is required that can quantify the similarity between the grid points based on the characteristics of the weather states. Spatial contiguity of the clusters is also desired in regionalization, which is traditionally realized by using spatial co-ordinates (latitude/longitude) of the grid points as additional attributes in defining a similarity metric [20]. However, many studies on clustering have demonstrated that this approach of achieving spatial contiguity often leads to poor regionalization, particularly if the desired cluster does not form a *simple region* with similar spatial coordinates [23]. This study address the issues of similarity metric and spatial contiguity by developing a regionalization algorithm in the Bayesian framework by using the concepts of HMM and Markov random fields.

Let the weather state at grid '$l$' be given by $\boldsymbol{Z}_l$, $l = 1, \ldots, M$ ($\boldsymbol{Z}_l = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ and $\mathcal{Z} = \{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_M\}$) where $M$ is the total number of grid points with observed data. Let $C$ be the number of clusters, and $\boldsymbol{f} = [f_1, \ldots, f_M]^{\mathrm{T}}$ ($f_l \in \{1, \ldots, C\}$) be the cluster labels. The posterior distribution of the cluster labels is obtained by using Bayes' rule as

$$p(\boldsymbol{f}|\mathcal{Z}) = \frac{p(\mathcal{Z}|\boldsymbol{f})p(\boldsymbol{f})}{p(\mathcal{Z})} \tag{1}$$

where $p(\boldsymbol{f})$ is the prior probability of the cluster labels and $p(\mathcal{Z}|\boldsymbol{f})$ is the likelihood function estimated as

$$p(\mathcal{Z}|\boldsymbol{f}) = \prod_{l=1}^{M} p(\boldsymbol{Z}_l|f_l) = \prod_{l=1}^{M}\prod_{c=1}^{C} \mathcal{N}(\boldsymbol{Z}_l|\boldsymbol{\mu}_{\boldsymbol{Z}_c}, \boldsymbol{\Sigma}_{\boldsymbol{Z}_c})^{\mathcal{I}_{lc}}. \tag{2}$$

In Eq. 2, $\mathcal{I}_{lc}$ is an indicator variable taking a value of one if $f_l = c$, and zero otherwise; and $\boldsymbol{\mu}_{\boldsymbol{Z}_c}$ and $\boldsymbol{\Sigma}_{\boldsymbol{Z}_c}$ are the mean and covariance of the weather state in the cluster $c$. The probability $\mathcal{N}(\boldsymbol{Z}_l|\boldsymbol{\mu}_{\boldsymbol{Z}_c}, \boldsymbol{\Sigma}_{\boldsymbol{Z}_c})$ will be large if the durations and timings of the weather states in grid '$l$' are similar to those of the region $c$, and relatively smaller otherwise, thus providing the desired similarity metric.

To encode the preference for spatial contiguity in the clusters, the prior probability $p(\boldsymbol{f})$ in Eq. 1 is chosen to be a Markov random field (MRF). An MRF, also known as a *Markov network*, is commonly used to model the joint distribution between spatially dependent variables. Using Hammersley-Clifford theorem [6], $p(\boldsymbol{f})$ is expressed as

$$p(\boldsymbol{f}) = \frac{1}{\Xi} \exp[-\sum_{l,m\in\mathbb{N}} E(f_l, f_m)] \tag{3}$$

where $\Xi$ is a normalizing constant, $E$ is an energy function in the space of clusters, and $\mathbb{N}$ is a *neighborhood set* for the grid points illustrated in Fig. 1. The prior distribution encodes the belief that the grid points in the neighborhood set are more likely to have same cluster labels - the degree of belief being controlled by the energy function.

The posterior probability $p(\boldsymbol{f}|\mathcal{Z})$ in Eq. 1 with an MRF prior cannot be estimated analytically. It can be approximated using variational Bayes or Markov chain Monte Carlo

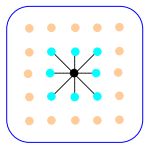


**Figure 1: The neighborhood set of the Markov random field used in the study. The neighborhood set for a grid point consists of 8 neighboring grid points.**

methods, however, the required computational cost can be excessive. In this study, for simplicity, the posterior distribution is approximated by a delta function at its mode $p(\boldsymbol{f}|\mathcal{Z}) \sim \delta(\boldsymbol{f}^\star)$ by maximizing the logarithm of the posterior distribution

$$\boldsymbol{f}^\star = \arg\max_{f} \sum_{l=1}^{M}\sum_{c=1}^{C} \ln p(\boldsymbol{Z}_l|f_l, \boldsymbol{\mu}_{\boldsymbol{Z}_c}, \boldsymbol{\Sigma}_{\boldsymbol{Z}_c}) - \sum_{l,m\in\mathbb{N}} E(f_l, f_m). \tag{4}$$

The maximization is performed iteratively by executing the following two steps. In the first step, for an assumed value of parameters ($\boldsymbol{\mu}_{\boldsymbol{Z}_c}$ and $\boldsymbol{\Sigma}_{\boldsymbol{Z}_c}$), the objective function is maximized with respect $\boldsymbol{f}$ using an algorithm based on *graph cuts* that guarantees the solution to lie within a constant factor from the global maximum (see [5] for details on the graph-cut algorithm). In the second step, for a given $\boldsymbol{f}$, the objective function is maximized w.r.t. $\boldsymbol{\mu}_{\boldsymbol{Z}_c}$ and $\boldsymbol{\Sigma}_{\boldsymbol{Z}_c}$. To make the calculations simple, the covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{Z}_c}$ is assumed to be shared by all the clusters, i.e. $\boldsymbol{\Sigma}_{\boldsymbol{Z}_c} = \epsilon\boldsymbol{I}$, where $\boldsymbol{I}$ is an identity matrix. In the limit $\epsilon \to 0$ the remaining parameter $\boldsymbol{\mu}_{\boldsymbol{Z}_c}$ can be obtained by maximizing the distortion function

$$\boldsymbol{J}(\boldsymbol{\mu}_{\boldsymbol{Z}_c}) = -\frac{1}{2}\sum_{l=1}^{M}\sum_{c=1}^{C} r_{lc}\|\boldsymbol{Z}_l - \boldsymbol{\mu}_{\boldsymbol{Z}_c}\|^2 \tag{5}$$

leading to the following estimation equations

$$r_{lc} = \begin{cases} 1 & \text{if } c = \arg\min_j \|\boldsymbol{Z}_l - \boldsymbol{\mu}_{\boldsymbol{Z}_j}\|^2 \,, \\ 0 & \text{otherwise;} \end{cases} \tag{6}$$

and

$$\boldsymbol{\mu}_{\boldsymbol{Z}_c} = \frac{1}{M_c}\sum_{l=1}^{M} r_{lc}\boldsymbol{Z}_l \tag{7}$$

where $M_c = \sum_l r_{lc}$ represents total number of grid points in the cluster $c$. Notice that due to the simplifying assumption made for the covariance matrix, the second step in the maximization of Eq. 4 resembles K-means clustering algorithm [20]. The energy function in Eq. 3 is selected as

$$E(f_l, f_m) = \begin{cases} 0 & \text{if } f_l = f_m, \forall l, m \in \mathbb{N}, \\ \lambda & \text{if } f_l \neq f_m, \forall l, m \in \mathbb{N}. \end{cases} \tag{8}$$

The value of $\lambda = 0$ corresponds to the uniform distribution in the space of clusters, i.e. no preference for spatial contiguity or $p(\boldsymbol{f})$ is noninformative. Higher values of $\lambda$ forces neighbors to be in the same clusters. After observing results from many simulations, the values of $\lambda \approx 0.01$ to 0.1 were found to be suitable for the present study. It was observed that the choice of prior distribution, $p(\boldsymbol{f})$, essentially affects the cluster assignment of primarily those grids points that are either on the fringes of a region or are outliers.

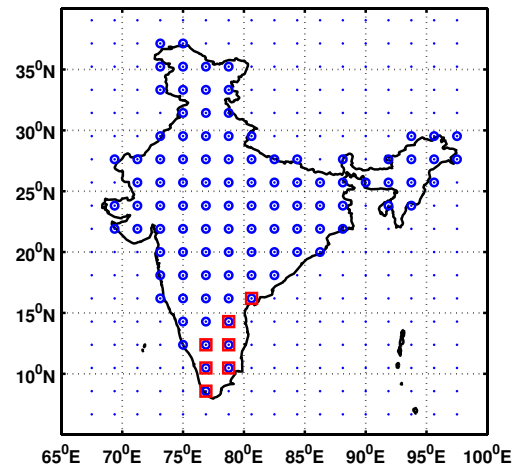



**Figure 2: The circles denote the grid points in the NCEP/NCAR data over the Indian summer monsoon region. The grid points in the north-east or winter monsoon region are marked by squares.**

## 3. DATA USED IN THE STUDY

Figure 2 shows 101 grid points over India in the NCEP / NCAR dataset [12]. The units of rainfall were reconverted form Kg/m$^2$/s to mm/day. There were no missing values in the data over the study region. In addition to 50 years (1950 to 1999) of reanalysis data provided for the challenge, nine years (2000 to 2008) of recent NCEP/NCAR reanalysis data were used to test robustness of the detected changes.

## 4. RESULTS AND DISCUSSION

The changes detected in the rainfall data are first presented followed by changes detected in the temperature data.

### 4.1 Changes in rainfall

India receives most of its rainfall from two monsoon seasons - the *south-west* or the *summer monsoon* during June to September in which most parts in the country receive rainfall, and the *north-east* or *winter monsoon* during November and December in which rainfall is confined to the eastern peninsular India (Fig. 2).

#### *4.1.1 Changes in the summer monsoon rainfall*

We start by exploring changes in the spatially averaged rainfall over the monsoon region. Figure 3 illustrates all India summer monsoon rainfall over the analysis period (1950-1999) along with a linear trend line. The MKT statistic for the time series is -1.10 which is not significant even at 5% level. The Bayesian change point analysis also could not find any significant abrupt change in the time series, implying that the summer monsoon is stable.

Next, changes were explored in the spatially distributed rainfall over the monsoon period. Figure 4 shows the MKT statistics for the median rainfall during June to September (JJAS). The results indicate that there is a decreasing trend over the Himalayas in the north, Western Ghats in the south, and hilly regions in the north-east. Apparently all the significant changes appear in the mountainous regions. However, we did not investigate changes over these

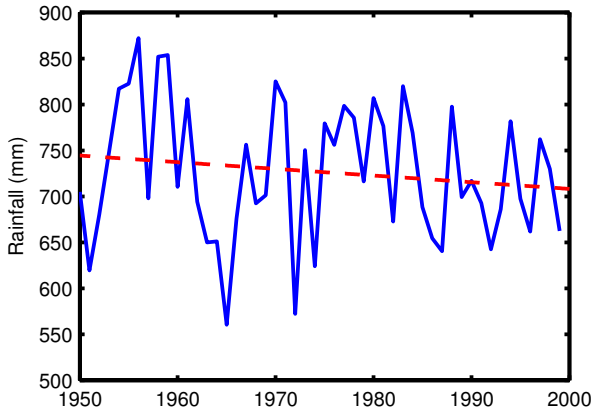



**Figure 3: Spatially average summer monsoon rainfall over India along with linear trend line. The time series exhibits neither significant linear trend nor significant abrupt changes.**

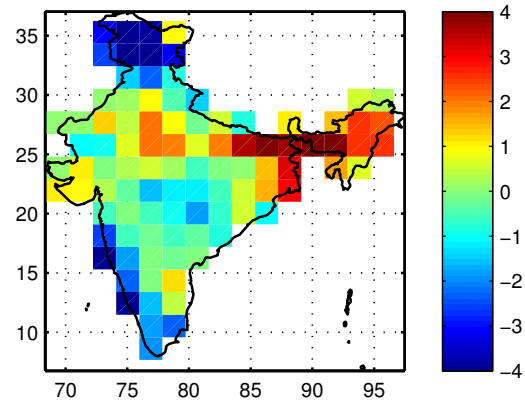



**Figure 4: Mann-Kendall statistics for testing linear trend in the median rainfall during June to September over India. The trends are significant at 1% level for the grid points with absolute value of test statistics greater than 2.3. The positive values of test statistics indicate increasing trend and vice versa.**

regions in detail because firstly, NCEP/NCAR analysis has some known issues in resolving rainfall over the high altitude regions [12], secondly, the rain gage density over these hilly regions is very low [19] to lend credibility to the reanalysis results, and finally, the changes appear over the isolated grid points instead of large contiguous regions.

The summer monsoon is most vigorous over the Indian sub-continent during July and August (JA), so we investigated rainfall during these two months in detail. Figure 5 shows the MKT statistics for the rainfall in JA, where apart from the hilly regions, the *north-west* plains (west of the river Ganges) appears to have an increasing trend in the median rainfall. Both gradual and abrupt changes were explored in various other rainfall statistics of these two months, and the most interesting result was obtained for the $90^{\text{th}}$ percentile (abbreviated as 90%tile in the paper) rainfall. The MKT statistic for the 90%tile July rainfall (Fig. 6) shows that the magnitude of extreme rainfall has increased over the north-west plains.

The results thus far indicate a weak signal of increasing JA rainfall over the north-west plains - the increase is more conspicuous for the extreme rainfall events. To gain more insight into the nature of these changes, the daily rainfall data were investigated, but before that it was deemed essential to delineate homogeneous regions having similar timings and durations of weather states. The regions were delineated using the regionalization algorithm discussed in section 2.4.

Figure 7 presents the homogeneous regions identified in the JA rainfall by the algorithm. The grid points in the north-west plains that have positive trends belong to region 3. The median rainfall time series in region 3 along with the probability of an abrupt change is shown in Fig. 8(a). The Bayesian change point algorithm suggested that around 1975 there was regime change in the monsoon characteristic of the region. The mean of the median rainfall after 1975 (2.14mm) is roughly twice the corresponding value before 1975 (1.18mm). The standard deviation has also increased from 0.78mm before 1975 to 1.30mm after 1975.

To ascertain the nature of increase, an ARHMM with four weather states was fitted to the daily average rainfall data in region 3. The results from the analysis indicate that the

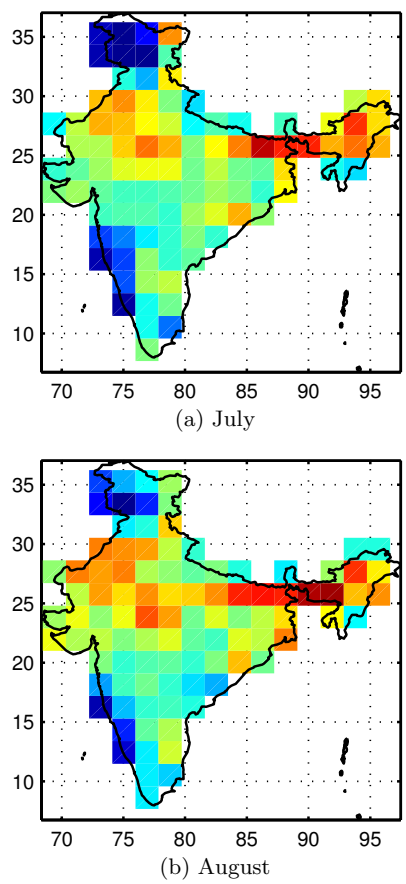



**Figure 5: Mann-Kendall statistics for testing linear trend in the median rainfall during (a) July and (b) August. The color coding is same as in Fig. 4.**

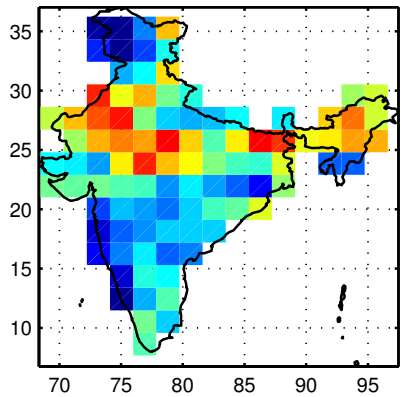



**Figure 6: Mann-Kendall statistics for testing linear trend in the 90%tile of July rainfall. The color coding is same as in Fig. 4.**

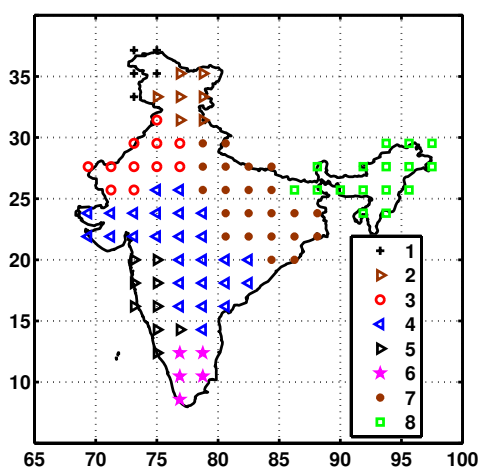
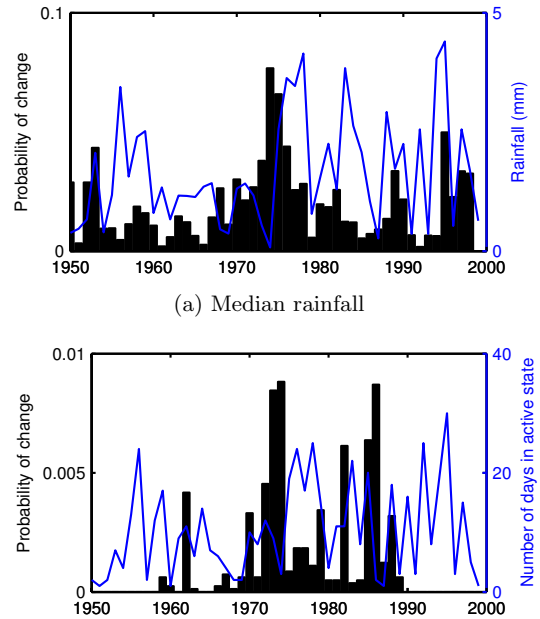



**Figure 7: Eight regions identified by the regionalization algorithm using July and August rainfall. The value of $\lambda$ in Eq. 8 was set to 0.08**




(a) Median rainfall

(b) Number of days in active state

**Figure 8: Probability of abrupt changes in the median rainfall and number of days in active state during July and August over region 3 (Fig. 7). The algorithm detected change point around 1975.**

number of days in the *active state* (i.e. the weather state associated with heavy rainfall events) has increased in post 1975 climate. Before 1975 there used to be on average 7.50 days of active spell whereas, after 1975, this value has gone up to 13 days.

### *4.1.2 Changes in the winter monsoon rainfall*

Figure 9 shows the spatially averaged rainfall time series over the north-east monsoon region along with linear trend line. The MKT statistic for the time series is 3.46 which implies a significant positive trend at 1% level. Figure 10 presents spatial distribution of the MKT statistics in the median rainfall during November and December, and it also suggests a positive trend in the rainfall over the winter monsoon region, the trend being more significant in November than in December, and more prominent in the 90%tile rainfall (Fig. 11) than in the median rainfall (Fig. 10).

To investigate the characteristics of the changes, an ARHMM with four weather weather state was fitted to the average daily rainfall over the winter monsoon region. The results suggest that the number of days in the active state has increased (Fig. 12(a)), but unlike the abrupt change observed in region 3 of the summer monsoon, the increase is gradual (MKT statistic = 3.55). In addition to the number of days in active state, the magnitude of the extreme rainfall has also increased (MKT statistic = 3.51) as shown in Fig. 12(b).

## 4.2 Changes in temperature

Figure 13 shows linear trend in the temperature during four typical seasons in the northern hemisphere, namely - DJF, MAM, JJA, and SON. The most conspicuous features

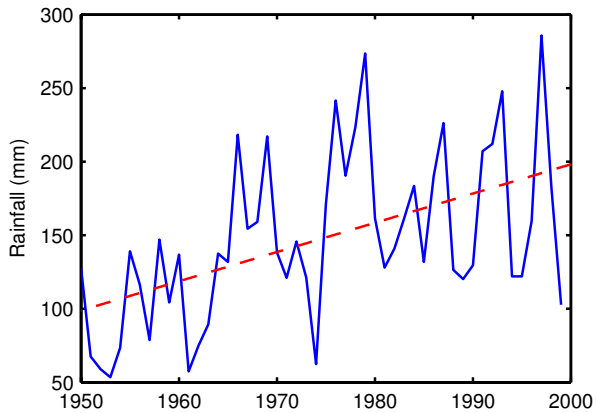



**Figure 9: Spatially average winter monsoon rainfall along with linear trend line. The time series exhibits linear trend significant at 1% level.**

in the figure are - negative trend over the north and central India during DJF, negative trend in the Gangetic plains during MAM and positive trend over the peninsular India during JJA. To investigate the characteristics of these changes, the *temperature homogeneous regions* delineated by India meteorology department and shown in Fig. 14 were used.

Figure 15 shows the 90%tile temperature time series in the north-central (NC) region during the winter months (DJF). The Bayesian change point analysis detected an abrupt change in the time series in late 1960s. The magnitude of the 90%tile temperature in the winter months has dropped by about 1.4°C (from 21.57°C to 20.18°C) after 1960s. No significant changes were detected in other characteristics of the temperature over the NC region during the winter months.

The analysis of MAM months over the NC region indicate a sudden change in the temperature during 1960s. The change was detected in the 90%tile, median, and 5%tile temperature time series. For brevity, only the 5%tile temperature time series and the probability of change in it are shown in Fig. 16. Similar changes (sudden drop) around the same period (1960s) in both DJF and MAM months suggest that there can be a common cause for these changes. It is interesting to note that in 1960s the NC region underwent *green revolution* in which large tracts of land were brought under irrigation.

In the peninsular India, the median and the 5%tile temperature time series show significant linear trend (1% significance level; MKT statistics being 5.85 and 5.90, respectively) while 90%tile time series shows no significant changes (Fig. 17). This result suggests that the magnitude of extreme temperature has not increased during the summer monsoon, but the the average temperature has gone up by about 1°C. A possible reason for this increase could be the proximity of the region to the Indian Ocean where the sea surface temperature has increased by about the same magnitude in the monsoon months over the last 50 years [10].

## 4.3 Robustness of the detected changes

To test robustness of the detected changes, nine years (2000 to 2008) of additional NCEP/NCAR reanalysis data were used. It should be emphasized that nine years of data

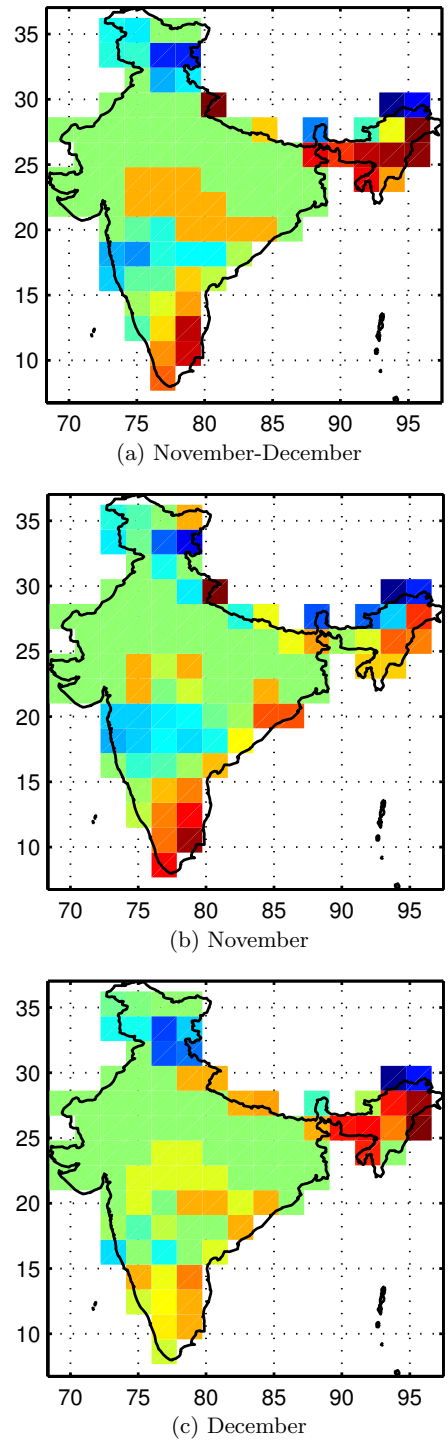



**Figure 10: Mann-Kendall statistics for testing linear trend in the median rainfall during (a) November-December, (b) November, and (c)December. The color coding is same as in Fig. 4.**

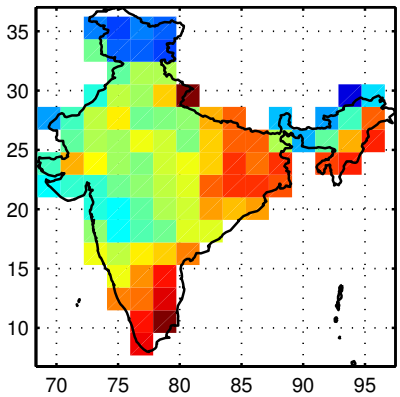
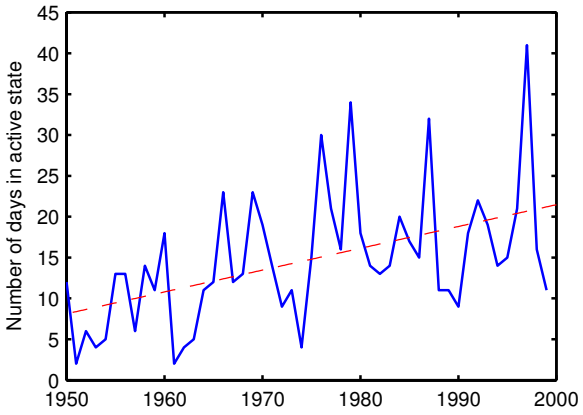
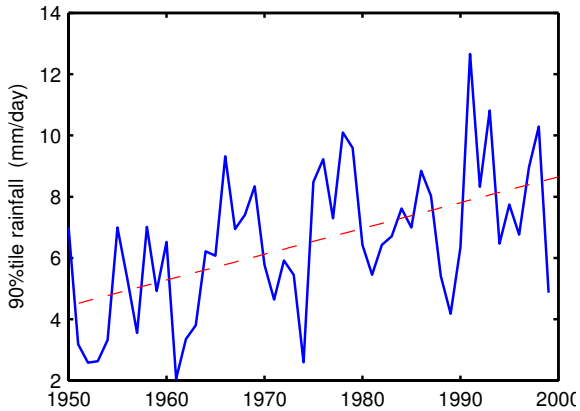



**Figure 11: Mann-Kendall statistics for testing linear trend in the 90%tile of November rainfall. The color coding is same as in Fig. 4.**

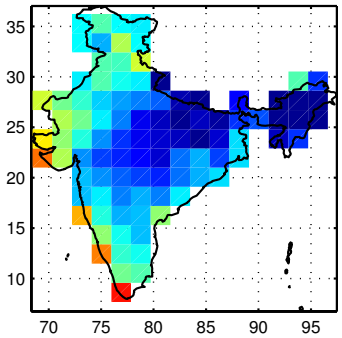



(a) Number of days in active state

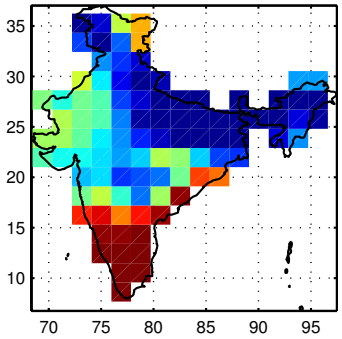



(b) 90%tile rainfall

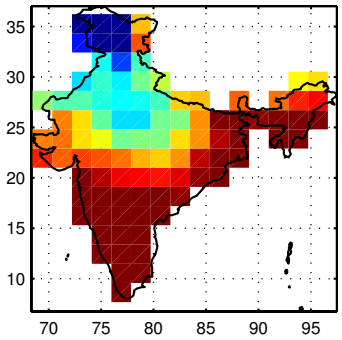
**Figure 12: Linear trend in (a) the number of days in active state and (b) 90%tile rainfall during November-December over the winter monsoon region. The trends are significant at 1% level.**



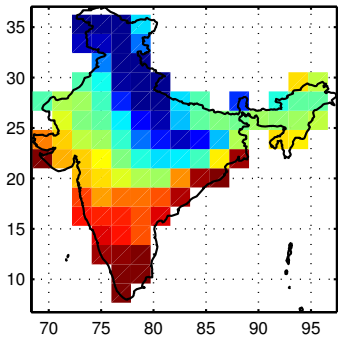

(a) DJF




(b) MAM




(c) JJA




(d) SON

**Figure 13: Mann-Kendall statistics for testing linear trend in the median temperature during four seasons. The color coding is same as in Fig. 4.**

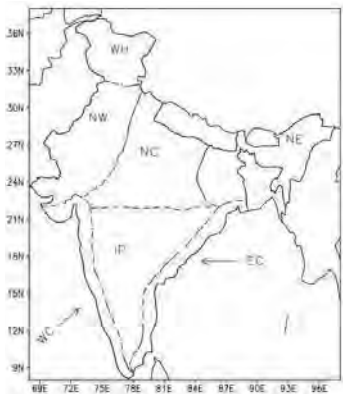



**Figure 14: Homogeneous temperature regions identified by India meteorological department, obtained from the website http://www.tropmet.res.in. The regions are named as: WH - Western Himalayas, NW- North-west, NC- North-central, NE- North-east, WC- West coast, EC- East coast, and IP- Interior Peninsula.**

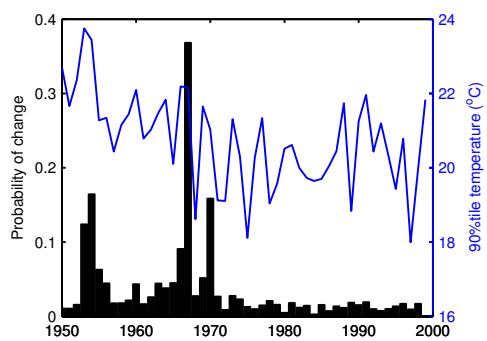



**Figure 15: Probability of abrupt changes in the 90%tile temperature over NC region during winter months (DJF). The algorithm detected change point in 1960s.**

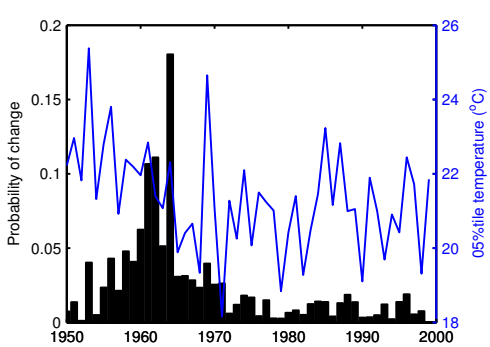



**Figure 16: Probability of abrupt changes in the 5%tile temperature over NC region during MAM months. The algorithm detected change point in 1960s.**

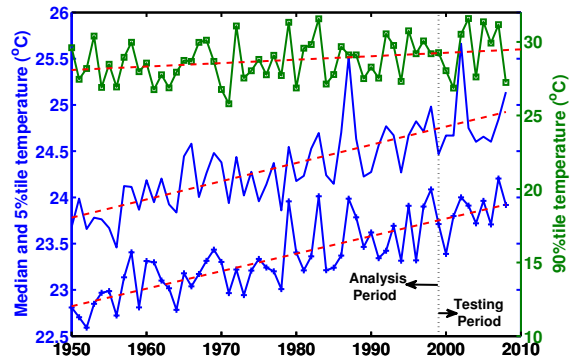



**Figure 17: Linear trend in the median, 5%tile, and 90%tile temperature over a region constituting Interior peninsula, and East and West coasts (Fig. 14). The solid line denotes median temperature, solid lines with square and plus symbols denote 95%tile and 5%tile temperature respectively, and dashed line represents linear trend. The trend is significant for median and 5%tile temperature at 1% level. The trends detected in the analysis period (1950 to 1999) were tested for their robustness in the test period (2000 to 2008).**

are not sufficient to test the robustness because presence of a few extreme years in the test data can significantly alter the test results. Nevertheless, the the steps described in the previous section were re-executed with the additional data and the results obtained are summarized below:

(1) Including test data in the analysis did not alter the probability of the abrupt change detected in the JA rainfall over north-west India during 1970s. The JA rainfall statistics over the region in post 1975 climate also remained same (mean of the median rainfall changed from 2.14 to 2.10mm, and number of active days changed from 13.0 to 13.4 days).

(2) The magnitude of gradually increasing trend detected in the winter monsoon (Fig. 9) was reduced by adding the test data (the MKT statistics dropped from 3.46 to 2.32) probably because of two extremely weak winter monsoon years (2002 and 2003).

(3) The abrupt changes detected in the temperature over north-central India during DJF and MAM months remained same in the test analysis, while the increasing trend in the JJA temperature over the peninsular India became more prominent (Fig. 17). The MKT statistics for the median and the 5%tile temperature time series increased from 5.85 and 5.90 to 6.83 and 6.98, respectively.

## 5. CONCLUDING REMARKS

In this study, Mann-Kendall trend test, Bayesian change point analysis, and a hidden Markov model were used to find changes in the rainfall and temperature patterns over India. A regionalization algorithm was developed to identify homogeneous regions experiencing similar weather states. The rainfall and temperature data were investigated at different spatial and temporal resolutions to identify the nature of changes. The main conclusions of the study are:

(1) All India summer monsoon rainfall is stable over the 50 years of analysis period. There is a weak signal that the number of active days (days with heavy rainfall events) has increased over the north-west plains after 1970s. This sudden change in the number of active days transpired around the same time when the climate over Pacific Ocean underwent a regime change [21] which is speculated to have an impact on the Indian summer monsoon.

(2) The winter or the north-east monsoon is gradually increasing due to significant increase in the frequency and magnitude of heavy rainfall events.

(3) The winter (DJF) and the spring (MAM) temperature over north-central India has encountered a sudden drop in mid 1960s which can be attributed to the green revolution in the area.

(4) The temperature over the Peninsular India has significantly increased during the monsoon months (JJA). This gradual increase may be linked to the increase in sea surface temperature in the Indian Ocean.

## 6. REFERENCES


[1] R. P. Adams and D. J. MacKay. *Bayesian online changepoint detection*. University of Cambridge, Cambridge, UK, 2007.

[2] D. Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279, 1992.

[3] B. Beckage, L. Joseph, P. Belisle, D. B. Wolfson, and W. J. Platt. Bayesian change-point analyses in ecology. *New Phytologist*, 174(2):456–467, 2007.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, USA, first edition, Aug 2006.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[6] P. Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems. A volume in honor of John M. Hammersley*, pages 19–32. Oxford University Press, 1990.

[7] F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, 2005.

[8] Y. Ephraim, D. Malah, and B. H. Juang. On the application of hidden Markov models for enhancing noisy speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 533–536, 1988.

[9] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, second edition, 2004.

[10] B. N. Goswami, V. Venugopal, D. Sengupta, M. S. Madhusoodanan, and P. K. Xavier. Increasing trend of extreme rain events over India in a warming environment. *Science*, 314(5804):1442–1445, 2006.

[11] M. Kendall. *Rank Correlation Methods*. Charles Griffin & Company Limited, 1948.

[12] R. Kistler, E. Kalnay, W. Collins, S. Saha, G. White, J. Woollen, M. Chelliah, W. Ebisuzaki, M. Kanamitsu, V. Kousky, H. van den Dool, R. Jenne, and M. Fiorino. The NCEP-NCAR 50-year reanalysis: monthly means CD-ROM and documentation. *Bulletin of the American Meteorological Society*, 82(2):247–267, 2001.

[13] K. K. Kumar, B. Rajagopalan, and M. A. Cane. On the weakening relationship between the Indian Monsoon and ENSO. *Science*, 284(5423):2156–2159, 1999.

[14] M. Lavielle. Using penalized contrasts for the change-point problem. *Signal Processing*, 85(8):1501–1510, 2005.

[15] T. M. Lenton, H. Held, E. Kriegler, J. W. Hall, W. Lucht, S. Rahmstorf, and H. J. Schellnhuber. Tipping elements in the Earth's climate system. *Proceedings of the National Academy of Sciences*, 105(6):1786–1793, 2008.

[16] G. Lorden. Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6):1897–1908, 1971.

[17] H. B. Mann. Nonparametric tests against trend. *Econometrica*, 13(3):245–259, 1945.

[18] L. Perreault, J. Bernier, B. Bobee, and E. Parent. Bayesian change-point analysis in hydrometeorological time series. part 1. the normal model revisited. *Journal of Hydrology*, 235(3):221–241, 2000.

[19] M. Rajeevan, J. Bhate, J. D. Kale, and B. Lal. High resolution daily gridded rainfall data for the Indian region: analysis of break and active monsoon spells. *Current Science*, 91(3):296–306, 2006.

[20] A. R. Rao and V. V. Srinivas. *Regionalization of watersheds - an approach based on cluster analysis*. Springer, New York, NY, USA, 2008.

[21] E. Suhas and B. N. Goswami. Regime shift in Indian summer monsoon climatological intraseasonal oscillations. *Geophysical Research Letters*, 35, L20703, 2008.

[22] S. Tripathi and R. S. Govindaraju. On the identification of intra-seasonal changes in the Indian summer monsoon. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data (Sensor-KDD 2009)*, 9 pages, in review, 2009.

[23] R. Zabih and V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Proceeding of Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 437–444, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

# Anomaly Detection and Spatio-Temporal Analysis of Global Climate System

Mahashweta Das
Dept. of Comp. Sc. & Engg.
The Ohio State University
Columbus, OH 43210
dasm@cse.ohio-state.edu

Srinivasan Parthasarathy
Dept. of Comp. Sc. & Engg.
The Ohio State University
Columbus, OH 43210
srini@cse.ohio-state.edu

## ABSTRACT

Knowledge discovery from temporal, spatial and spatio -temporal data is pivotal for understanding and predicting the behavior of Earth's ecosystem model. An important influence leaving its impact on the ecosystem is the global climate system. In this paper, the Earth Science data that we have analyzed consists of daily global air temperature and precipitation measurements, aggregated from heterogeneous sensors for fifty years (1950-1999). The enormous amount of data that is available for analysis requires employment of data mining techniques for discovering interesting patterns, detecting significant changes and extracting meaningful insights from the data. Our work considers the problem of detecting anomalous (abnormal or unexpected) behavior in the global climate system, discovering teleconnection patterns and providing consequential insights to the analysts.

## Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; H.2.8 [**Database Management**]: Database applications-Data Mining

## Keywords

global climate system, data mining, distance-based outlier detection, spatio-temporal analysis, climate phenomena, teleconnections

## 1. INTRODUCTION

Climate change is an emerging potential global stress on all of the Earth's ecosystems. For the past few billion years, the Earth's climate and the biosphere have been in constant flux, with events ranging from ice ages to long periods of warmth. Changes in climate either occur rapidly such as ice-sheet disintegration in the poles or gradually over a long period of time. Again, variation in climate occurs locally such as variation in humidity due to effects of irrigation as well as globally. A widely researched and debated climate phenomenon is global warming, which is the increase in Earth's average global temperature. Natural factors such as volcanic eruptions, changes in the Earth's rotation and orbit, amount of energy released from the Sun as well as human activities post Industrial Revolution, population growth, economic and technological development, anthropogenic greenhouse gases have affected the Earth's climate. In 2004, the Ecosystems Interagency Working Group (EIWG) of the U.S. Climate Change Science Program (CCSP) organized a workshop to identify and address the consequences of global climate changes for ecological systems. The global nature of environmental problems requires assistance from different disciplines such as statistics, mathematics and computer science for capturing and modeling significant changes in the structure and function of Earth's ecosystem.

Climate related observations from satellite sensors, airborne cameras and weather radars accrue terabytes of temporal, spatial and spatio-temporal data. The need to efficiently handle and effectively analyze such humongous amount of data encourages the application of data mining techniques in this domain of research. The challenges associated with analyzing these datasets are: (i) computational complexity due to the availability of astronomical amount of data, (ii) algorithmic complexity due to spatial dependency since parallelization of this work is difficult, and (iii) complexity due to non-linear correlations between processes. Data mining techniques and algorithms are equipped to maneuver such datasets and develop anticipatory insights about extreme weather condition changes (heat-wave, severe rainfall and drought), assess climate shifts, conceive teleconnection patterns and develop seasonal-to-interannual climate forecasts, while fostering the application of such analyses to the benefit of society. For example, Support Vector Machine and Artificial Neural Network can help classification and prediction of climate behavior; again identifying regions with similar climate trends can be achieved by K-Means and Principal Component Analysis. Another track of analyzing such datasets is detecting outliers and other anomalies. In the context of climate research, outliers may arise either due to erroneous measurements or due to some exceptional event taking place.

Our work deals with detecting spatial, temporal and spatio-temporal anomalies in the global climate system. A spatial outlier is a geographically referenced area whose climate measurements are significantly different from those of other objects in its spatial neighborhood whereas temporal outliers are those time periods which experience abrupt changes

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*SensorKDD* '09, June 28, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-668-7 ...$5.00.

in the historical similarity trends. A spatio-temporal outlier is one whose climate behavior is significantly different from those of other spatially and temporally referenced objects in its neighborhood. We have used data mining techniques and algorithms for this purpose. The data pre-processing step reduces the dimensionality of the dataset and the outlier detection algorithm works on the reduced feature space to identify outliers.

This paper addresses the following questions about the global climate dataset provided to us, as part of the challenge:

- What are the durations between 1950 and 1999 that have experienced anomalous behavior?
- What are the geographical locations on Earth which have savored irregularity in climate?
- What are some of the spatio-temporal outliers in the global climate dataset?
- Are the anomalies due to irregular temperature measurements and/or precipitation measurements?
- Do the detected time periods match the timings of some of the significant climate phenomena recorded in history? If yes, what are the events?
- Do the detected locations experience such aberrations due to some of these climate phenomena? In other words, is there a relationship between the outliers recognized in temporal analysis with those ascertained in spatial analysis?
- Does anomaly detection automate the discovery of teleconnection patterns?

The rest of the paper is organized as follows: we begin in Section 2 with the problem definition; Section 3 talks about the issues related to preprocessing the data including a brief description of the raw data format; Section 4 addresses some of the related work and the outlier detection method that we think are suited for the analysis of the dataset in hand; Section 5 and 6 are concerned with the experimental results and the analysis of the results; Section 7 discusses the results of our spatio-temporal analysis and finally, we conclude the paper in Section 9.

## 2. PROBLEM STATEMENT

Our focus in this work is to identify significant changes in the global climate dataset, consisting of both temperature and precipitation measurements. The original call does not mention any precise definition of the 'change'. It may be an unexpected change such as an anomaly, a gradual change or may be an extreme event like drought and flood. The changes can either be tracked across days (*1950-1999*) against which measurements have been collected or across different locations (*latitude, longitude*) on Earth or across both space and time. In this paper, we have detected spatial, temporal and spatio-temporal anomalies and endorsed how the results from the individual analysis are coordinated with each other. The results also manage to provide an improved understanding of potential links between climate events that occur relatively large distance apart. Our problem statement is as follows:

*Given the daily temperature and precipitation levels of all locations on the Earth, find those days and those locations in which temperature and precipitation is* ***different*** *compared to other days and locations. This is our notion of 'significant change' in the context of global climate data. Once the time periods and zones of aberration have been detected, match them to the available records of climate phenomena as well as to each other. Analyze the results for identifying teleconnections linking worldwide climate anomalies too.*

The following section provides a brief description of the dataset and how it has been preprocessed for our need.

## 3. DATA PREPARATION

For this challenge, we have used the daily global temperature and precipitation data, aggregated from heterogeneous sensors for 50 years (1950-1999) [9]. Each sensor corresponds to a physical location (latitude, longitude) on the Earth's surface. The data is downloadable in csv format from the Oak Ridge National Laboratory website [2].

### 3.1 Raw data format

The temperature and precipitation values have been provided in separate files with each entry as:

$$latitude, longitude, temp_day1, \ldots, temp_dayN$$

Temperature records are in Degrees Celsius (°C) and precipitation records are in millimeters per day (mm/day). The raw data has 18048 rows (observations) and 18264 columns (days). The 18048 rows stand for locations on the Earth: 94 latitudes ranging from 90°N (88.5420°N) to -90°N (-88.5420°N) and 192 longitudes per latitude between 0°E and 360°E (358.1250°E); the columns stand for days from 1950 to 1999. It may be observed that each row is an independent observation of time series having 18262 data points. Figure 1 shows the plot of the temperature and precipitation for the first location (latitude 88.5420°, longitude 0.0°) in the datasets for all days (1950-1999).

### 3.2 Data Pre-processing

The question that we want to answer in the very first place is which years and which locations on Earth have experienced most anomaly. The downloaded data has thus been preprocessed accordingly, before using any data mining technique on them.

The challenge dataset is characterized by huge dimensionality and hence beseeches the employment of feature selection or feature compression techniques for data reduction. We have identified two relevant features, temperature and precipitation, each of which is a linear combination of the data points available in the respective datasets.

For detecting the outliers over time, we have first reduced the number of columns of each from 18264 to 18262 by removing the first two column entries (which have latitude and longitude information) and then transposed each of the original data to obtain a 18262 × 18048 matrix. Each of the final matrix looks as follows:

| | *location1* | *location2* | *location3* |
|---|---|---|---|
| *day1* | | | |
| *day2* | | | |

where each location stands for <*latitude, longitude*>. The dimensionality of each of this data matrix has now been re-

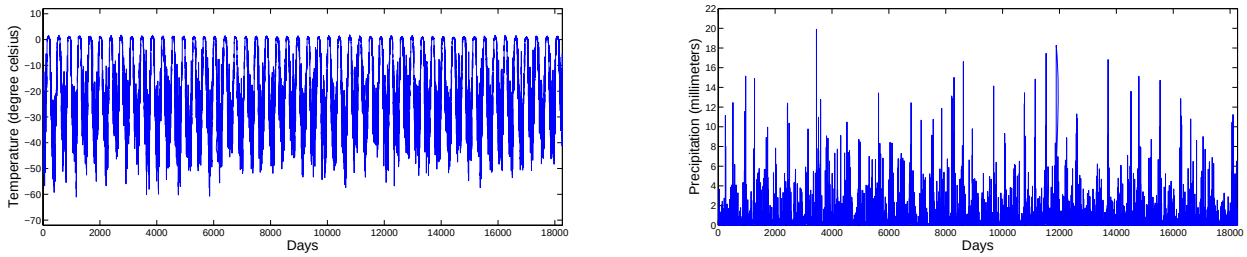



**Figure 1: Variation of temperature (left) and precipitation (right) for all days for the first location(latitude 88.5420°N, longitude 0.0°E). The graphs show periodic behavior for both the variables.**

duced from 18048 to 1 by computing the L2-Norm of temperature and precipitation measures for each day. The final pre-processed matrix is 18264 × 2 where the two column entries stand for the L2-Norm values of temperature and precipitation across all locations for each day respectively.

For detecting the outliers over space, we have again reduced the number of columns of each from 18264 to 18262 by removing the first two column entries. The matrix is 18048 × 18262 in this case and looks as follows:

| | *day1* | *day2* | *day3* |
|---|---|---|---|
| *location1* | | | |
| *location2* | | | |

where each location stands for <*latitude, longitude*>. The dimensionality of each of this data matrix has now been reduced from 18262 to 1 by computing the L2-Norm of temperature and precipitation values for each location. The final pre-processed matrix is 18048 × 2 where the two column entries stand for the L2-Norm values of temperature and precipitation values over all days for each location respectively.

For detecting outliers that exist when both spatial and temporal neighborhood are taken into consideration, we have selected the sub portion of interest from each of the original data matrix such that consecutive locations constituting the local spatial neighborhood are along rows and the temporal neighborhood is along column. Each of this matrix has then been converted to a n×m spatial grid which is as follows:

| | *longitude1* | *longitude2* | *longitude3* |
|---|---|---|---|
| *latitude1* | | | |
| *latitude2* | | | |
| *latitude3* | | | |

where the $(i,j)^{th}$ cell bears the temperature or precipitation measurement at a specific latitude and longitude on Earth.

All the steps of data pre-processing have been conducted in a Linux workstation with Matlab 2009a installation.

# 4. RELATED WORK

There has been enormous interest in mining spatial, temporal and spatio-temporal patterns of late since the world is changing at a constantly increasing pace. In 2001, Tan *et al.*[18] has discussed some of the challenges associated with preprocessing and analyzing Earth Science data as well as some of the patterns of interest namely, associations, trends, clusters and predictive models. Lin *et al.* [12] has used a high dimensional clustering method and a method that mines episode association rules in event sequences for discovering patterns such as teleconnections between the abnormally low temperature events of the North Atlantic and floods in Northern Bolivia. To analyze the effect of the oceans and atmosphere on land climate, Earth Scientists have developed climate indices that summarize the behavior of selected regions of Earth's oceans and atmosphere over time. In the past, researchers have used observations and eigenvalue analysis techniques, such as principal components analysis (PCA) and singular value decomposition (SVD) for discovering climate indices. Steinbach *et al.*[17] has proposed an alternative shared nearest neighbor (SNN) clustering-based methodology for determining climate indices such that centroids of the clusters summarize the behavior of the ocean or atmosphere in those regions. Kumar[11] has addressed the computational challenges in understanding patterns in global climate system from the context of high-performance data mining.

In 2008, Auroop R Ganguly and Karsten Steinhaeuser compared the problem of climate data mining with spatiotemporal data mining (SSTDM) in [5], formulating new challenges such as long-range and nonlinear dependence, nonlinear dynamic behavior, presence of thresholds, importance of extreme events caused by global climate changes, uncertainty quantification, etc. They presented a case study too to demonstrate how simple data mining applications are capable of bringing to light novel insights in climate science. Our work is a direct corroboration of this fact. The following subsection explains the data mining techniques that we have used for this work.

## 4.1 Distance-based Outlier Detection

Since the original dataset is not labeled, we have decided to use an unsupervised anomaly detection method. Unsupervised techniques can determine outliers based on nonparametric measures such as distance and density. Since outliers are events with very low probabilities, it is extremely difficult to compute the exact probability. Therefore, most of these techniques are heuristics and are based on the general concept of nearest neighbors. Knorr *et al.* [10] and later Ramaswamy *et al.* [14] proposed several definitions of distance based outlier detection techniques. As an example, consider the $k$ nearest neighbor based anomaly detection technique. For every data point, the algorithm first finds the average distance to its $k$ nearest neighbors. The 'outlierness' of a point is then proportional to this average distance. Higher the distance, more anomalous is the point. There are several other definitions of distance-based outlier detection

techniques such as the distance to its $k$-th nearest neighbor, distance to its nearest neighbor and more. In this work, we have used Orca, a near-linear time distance-based outlier detection algorithm developed by Bay and Schwabacher [3].

Orca is written in C++ and compiled with MinGW32 (Windows). All our simulations have been conducted on Windows platform. The default values for the distance computation have been chosen as the average distance to five nearest neighbors.

## 4.2 Neighborhood-based Outlier Detection

The traditional data mining algorithms are not often applicable for mining spatial datasets since spatial objects are affected by spatial processes in the neighborhood. The climate behavior at a particular location is indeed influenced by the climate conditions in its vicinity. Adam *et al.*[1] has addressed the identification of spatio-temporal outliers and trends in terms of the spatial and semantic relationships among objects by considering the underlying spatial processes as well as the features of these spatial objects. This notion of spatial dependency along with the consideration of a local temporal neighborhood helps us to identify spatio-temporal outliers.

In this approach, we have computed a similarity score for each unit in a n×m spatial grid by considering its r-point neighborhood. If the score of a unit is significantly different from the other scores in the grid, it is exhibiting spatially anomalous behavior. We have computed the scores of each of the unit in the spatial grid for a couple of consecutive timestamps. If a spatially anomalous unit is found to deviate from the historical trends, it is deemed to be a spatio-temporal outlier. Such spatio-temporal outliers encoding the sensor context information are often referred to as 'Conditional Anomalies'[16]. The cost of computing the score for all units in the spatial grid is linear, if a r-point neighborhood is considered, where r is much less than n×m

All steps of this neighborhood-based approach has been implemented in a Linux workstation with Matlab 2009a installation.

# 5. EXPERIMENTAL RESULTS

The results of our experiment on the global climate data using the method described in the preceding section have been illustrated here. The analysis of the results have been carried out in the next section.

## 5.1 Outliers over time

Table 1 depicts the top 20 Global Outlier Scores and the days associated with each of the scores. In other words, the table lists the top 20 days within 1950 and 1999 over all locations on Earth which have experienced most anomalous behavior, either due to irregular temperature or precipitation or both. Higher the Global Outlier Score, more anomalous the day is. In the table, the Temperature(Temp) and Precipitation(Prec) columns indicate each of their contributions towards the Global Outlier Score. Orca took 0.4 seconds to build the model.

Figure 2 shows the plot of the global outlier score and Figure 3 shows the plot of the temperature and precipitation outlier scores over all days (1950-1999).

## 5.2 Outliers over space

**Table 1: Outlier detection results over time**

| Rank | Day | GlobalScore | Temp | Prec |
|---|---|---|---|---|
| 1 | 17115 (Nov,1996) | 23562 | 11 | 89 |
| 2 | 17118 (Nov,1996) | 20732 | 36 | 64 |
| 3 | 17122 (Nov,1996) | 13470 | 8 | 92 |
| 4 | 17114 (Nov,1996) | 11843 | 7 | 93 |
| 5 | 17113 (Nov,1996) | 11311 | 19 | 81 |
| 6 | 17235 (April,1997) | 10407 | 24 | 76 |
| 7 | 941 (July,1952) | 7618 | 54 | 46 |
| 8 | 17237 (March,1997) | 5967 | 50 | 50 |
| 9 | 10802 (July,1980) | 5844 | 77 | 23 |
| 10 | 17123 (Nov,1996) | 5402 | 7 | 93 |
| 11 | 17488 (Nov,1997) | 4882 | 45 | 55 |
| 12 | 12112 (March,1984) | 4637 | 51 | 49 |
| 13 | 10801 (July,1980) | 4504 | 76 | 24 |
| 14 | 17233 (April,1997) | 4160 | 42 | 58 |
| 15 | 3631 (Dec,1959) | 3673 | 29 | 71 |
| 16 | 17236 (April,1997) | 3553 | 19 | 81 |
| 17 | 11668 (Dec,1982) | 3186 | 79 | 21 |
| 18 | 12111 (March,1984) | 3160 | 41 | 59 |
| 19 | 17125 (Nov,1996) | 3135 | 44 | 56 |
| 20 | 2882 (Nov,1957) | 3101 | 14 | 86 |

Table 2 depicts the top 20 Global Outlier Scores and the location Earth associated with each of the scores. In other words, the table lists the top 20 locations on the Earth between 90°N and 90°S,0°E and 360°E over all days (1950-1999) which have experienced most anomalous behavior, either due to irregular temperature or precipitation or both. Higher the Global Outlier Score, more anomalous the location is. In the table, the Temperature(Temp) and Precipitation(Prec) columns indicate each of their contribution towards the Global Outlier Score. Orca took 0.4 seconds to build the model.

**Table 2: Outlier detection results over space**

| Rank | Location | GlobalScore | Temp | Prec |
|---|---|---|---|---|
| 1 | 8600 (4°N,103°W) | 3122990 | 29 | 71 |
| 2 | 2477 (6°N,142°W) | 1936570 | 44 | 56 |
| 3 | 2668 (63°N,140°W) | 1361460 | 15 | 85 |
| 4 | 8601 (4°N,113°W) | 975072 | 55 | 45 |
| 5 | 8408 (6°N,103°W) | 975072 | 15 | 85 |
| 6 | 2476 (65°N,140°W) | 782831 | 15 | 85 |
| 7 | 8409 (6°N,105°W) | 636560 | 60 | 40 |
| 8 | 8792 (2°N,103°W) | 589208 | 64 | 36 |
| 9 | 16707 (77°S,3°E) | 462306 | 18 | 82 |
| 10 | 8217 (8°N,105°W) | 370309 | 13 | 87 |
| 11 | 2478 (65°N,144°W) | 364466 | 46 | 54 |
| 12 | 16708 (77°S,5°E) | 302682 | 21 | 79 |
| 13 | 8213 (8°N,97°W) | 296638 | 57 | 43 |
| 14 | 16355 (73°S,63°E) | 292548 | 33 | 67 |
| 15 | 11295 (21°S,116°W) | 276308 | 27 | 73 |
| 16 | 2860 (61°N,140°w) | 269883 | 31 | 69 |
| 17 | 7830 (12°N,99°W) | 265246 | 40 | 60 |
| 18 | 8214 (8°N,99°W) | 256445 | 39 | 61 |
| 19 | 2669 (63°N,142°W) | 244847 | 12 | 88 |
| 20 | 7829 (12°N,97°W) | 241949 | 54 | 46 |

Figure 2 shows the plot of the global outlier score and Figure 4 shows the plot of the temperature and precipitation

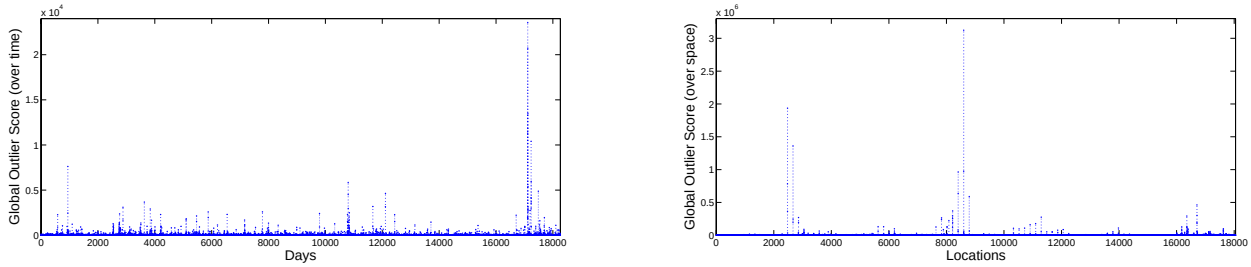



**Figure 2: Variation of global outlier score for all days (left) and for all locations on Earth(right).**

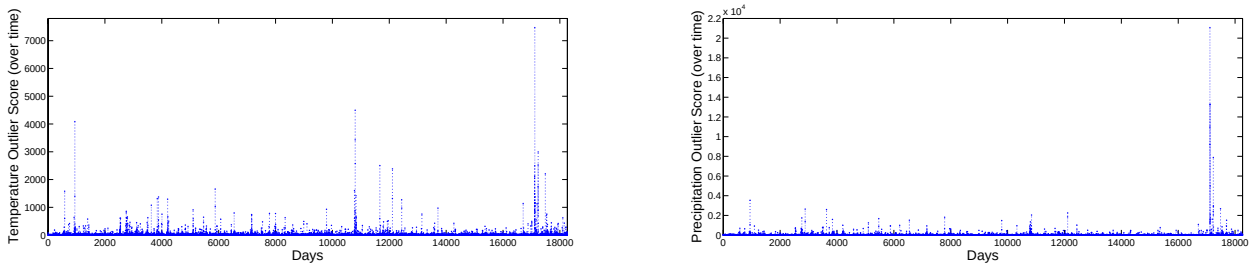



**Figure 3: Variation of temperature outlier score (left) and precipitation outlier score (right) for all days. The graphs show quite a few anomalous behavior for both the variables.**

outlier scores and over all locations (90°N to -90°N, 0°E to 360°E).

## 6. ANALYSIS OF RESULTS

The outliers detected by performing experiments on the global climate data have been analyzed in this section for a relevant interpretation.

### 6.1 Temporal Analysis

Based on the top 200 outlier scores over time, we infer that significant anomalous behavior has been experienced in the following time periods (ranked in order of the global outlier score):

1. 1996-1998: Anomalous behavior started creeping in during November, 1996 and it continued till July, 1998. The top outliers in the entire dataset (1950-1959) are from this period. During the first half of this anomalous behavior, there was abnormal increase in both temperature and precipitation measurements across different locations; finally, towards the end in 1998, heavy precipitation was recorded as well.

2. 1950-1952: This period was marked by fluctuating temperature measurements; however both temperature and precipitation contributed to it being marked as an outlier.

3. 1980-1984: During the July of 1980, there was a sudden increase in the temperature which was followed by both irregular temperature and precipitation till March 1984.

1955-1957, 1965-1967 and 1971-1973 are also detected as intervals of significant anomalous climate behavior.

Our results are reasonably aligned with the anomalies cached in the historical records of global climate data. We have identified outliers due to two major climate phenomena : the Pacific ocean signatures **El Niño** and **La Niña**, which are nothing but ocean-atmosphere phenomena marked by temperature fluctuations in water surfaces of the tropical Eastern Pacific Ocean. La Niña is characterized by unusually cold ocean temperatures in the Equatorial Pacific, whereas El Niño (commonly referred to as El Niño-Southern Oscillation or ENSO) is characterized by unusually warm ocean temperatures over the Pacific Ocean Equator. According to the consensus list of El Niño and La Niña [7], the years 1997 and 1998 were characterized by a cyclical sequence of El Niño and La Niña of very strong type. Again, [15] and [8] confirms that 1957, 1965, 1966, 1967, 1972, 1982 and 1983 are El Niño years whereas 1950, 1955, 1973 are spans of strong La Niña.

### 6.2 Spatial Analysis

Based on the top 200 outlier scores over space, the following regions have been found to experience notable anomalous behavior either due to irregular temperature or precipitation or both (ranked in order of the global outlier score):

1. 12°N to 12°S, 80°W to 140°W: Area enclosed in this zone mainly includes equatorial South Pacific Ocean, parts of Ecuador, Colombia and Peru.

2. 60°N to 65°N, 110°W and 160°W: Area enclosed here includes parts of Alaska and Canada.

3. 73°S to 77°S : This zone encloses parts of Antarctica.

Parts of Chile, Bolivia, north-western coast of United States, Kenya, Bangladesh and Indonesia are also detected as locations experiencing anomalous climate.

Figure 5 shows the plot of the locations (based on top 400 outlier scores) on the world map which have been classified as an outlier due to irregular temperature (top) and precipitation (bottom).

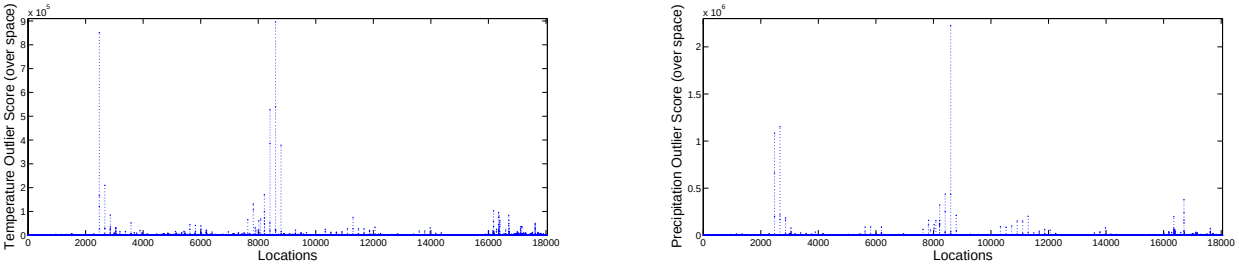



**Figure 4: Variation of temperature outlier score (left) and precipitation outlier score (right) for all locations on the Earth's surface. The graphs show quite a few anomalous behavior for both the variables.**

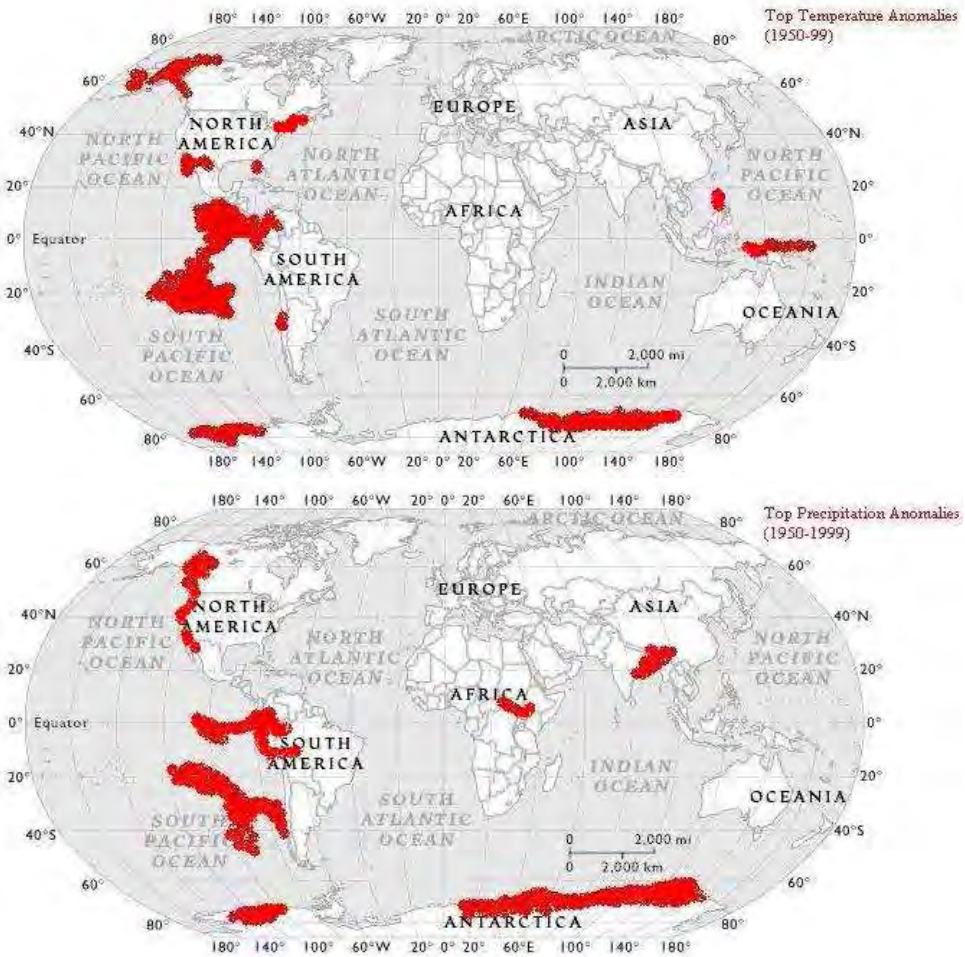



**Figure 5: Distribution of anomalous temperature (top) and precipitation (bottom) areas on Earth's surface**

These set of results go hand-in-hand with the areas affected by the impact of El Niño and La Niña and accounts for some of the calamities recorded in history such as drought and floods, due to these Southern Oscillations. In the figure 5, the most prominent zone is the area bounded by 12°N to 10°S, 80°W to 140°W. This is a part of the South Pacific Ocean, which is the cradle of El Niño-Southern Oscillation (ENSO). The effects of El Niño in South America are much more direct and strong than in North America. According to [6], the western coast of South America including Peru, Ecuador, Chile, northern Argentina and Southern Brazil experience unusual rainfall and flooding conditions. In North America and Alaska, winters are warmer than usual, whereas California, northwest Mexico and southwest US receive excessive rainfall. The winter season of 1997-1998 has been one of the wettest on record across California. Direct effects of El Niño result in drier conditions in parts of Southeast Asia and droughts in Northern Australia, which

receive little or absolutely no precipitation. Kenya, Tanzania and the White Nile Basin as well as parts of Bangladesh experience wetter than normal conditions. Figure 5 reflects most of the above-mentioned irregular behaviors that take place due to the fickle moods of the El Niño-Southern Oscillation.

### 6.3 Teleconnections

Teleconnection patterns refer to the relationship between climatic anomalies that occur in widely separated regions on Earth. As is evident from the previous two sections, the El Niño and La Niña variability in the Pacific region affects the climate behavior of different locations on Earth and hence some interesting teleconnections are associated with these ENSO events. For example, drought in Northern Australia and floods in East Africa are due to the Southern Oscillations. Figure 5 essentially reveals some of the teleconnection patterns on Earth. One of the most interesting teleconnection pattern in the figure is how the anomalies of sea surface temperature in the equatorial and tropical zones has resulted in anomalies in the polar Antarctic Peninsula. Thus, anomaly or outlier detection from the global climate system automates the discovery of interesting teleconnection patterns.

## 7. SPATIO-TEMPORAL ANOMALIES

In this section, we have presented our results and analyses of contextual or spatio-temporal anomalies. Figure 6 shows the plot of the winter (January) temperature in California, USA for four consecutive years - 1950, 1951, 1952, 1953. The plot fails to distinguish normal climate behavior from anomalous behavior, if any. If, the distance-based outlier detection method is used to compute the global outlier score for each location, it is also not much able to capture spatio-temporal abberations, as is evident from Figure 7(top). Figure 7(bottom) shows the plot of the similarity scores of winter temperature (January) in California, USA for the four consecutive years and it clearly highlights the contextual outliers. Four of the spatio-temporal outliers marked in Figure 7(bottom) have occured in a year witnessing an ENSO event.

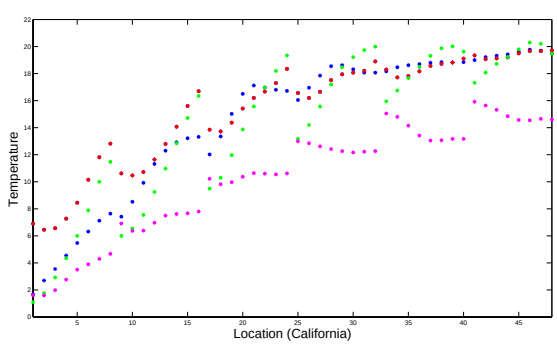



**Figure 6: Variation of winter (January) temperature in California, USA for 1950(magenta), 1951(green), 1952(red) and 1953(blue).**

Thus, the spatial and temporal neighborhood-based approach helps us to monitor the anomalies that are hard to detect in the global picture. The case study in the following subsection highlights how our approach of neighborhood-based outlier detection is suited for handling spatio-temporal outliers over the traditional distance-based outlier detection approach.

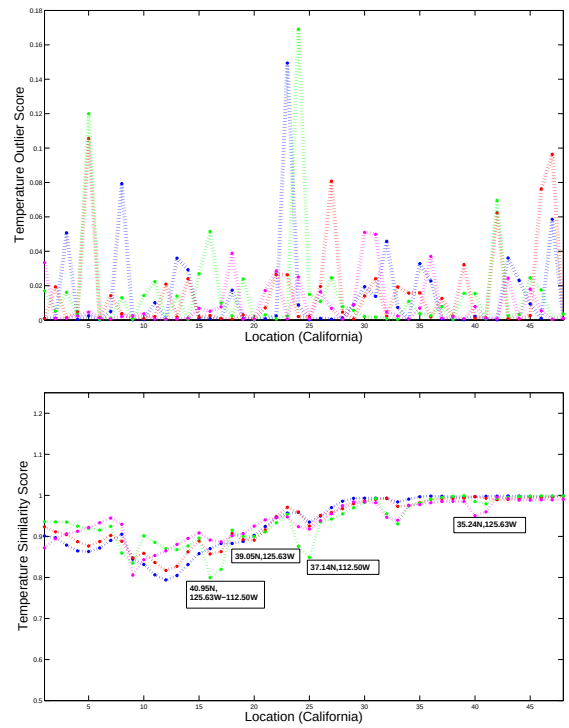



**Figure 7: Variation of distance-based outlier score (top) and neighborhood-based similarity score (bottom) of winter(January) temperature in California, USA for 1950(magenta), 1951(green), 1952(red) and 1953(blue).**

### 7.1 Ohio: A Case Study

We have analyzed the temperature of Ohio in recent consecutive years using the approach proposed in Section 4.2 to accentuate its merit. The data for the analysis has been downloaded from [4]. Figure 8 shows the plot of the temperature in the month of January in Ohio, USA for three consecutive years - 2004, 2005 and 2006. This plot, like Figure 6 cannot classify normal and anomalous climate behavior. The only piece of useful information that one can infer from Figure 8 is that the average temperature for the month of January has significantly increased from 2004 (green) to 2006 (blue). Figure 9(top) shows the plot of the distance-based outlier detection scores and Figure 9(bottom) shows the plot of the neighborhood-based similarity scores of average January temperature in Ohio, USA for the three consecutive years. According to the distance-based outlier detection approach which presents a global view, the locations 38.25°N, 80.25°W and 42.75°N, 80.75°W are experiencing anomalous behavior. However, on closer analysis using the neighborhood-based approach which presents a local view, we find that they are spatial anomalies and not spatio-temporal ones since the trend in climate at those loca-

tions are consistent over all three years. The spatio-temporal anomaly in Ohio is at location 41.25°N, 81.75°W in the year 2004. Newspaper reports [13] indicates that this location, close to Lake Erie did experience 'bone-chilling' temperature in January, 2004.

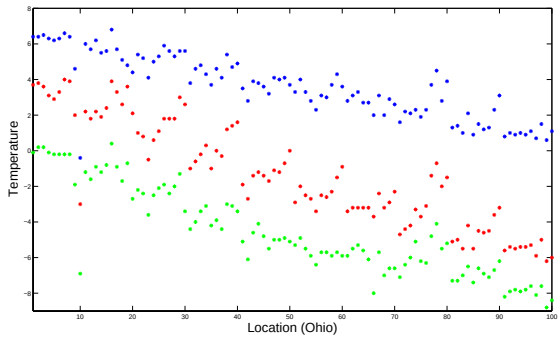



**Figure 8: Variation of average January temperature in Ohio, USA for 2004(green), 2005(red) and 2006(blue).**

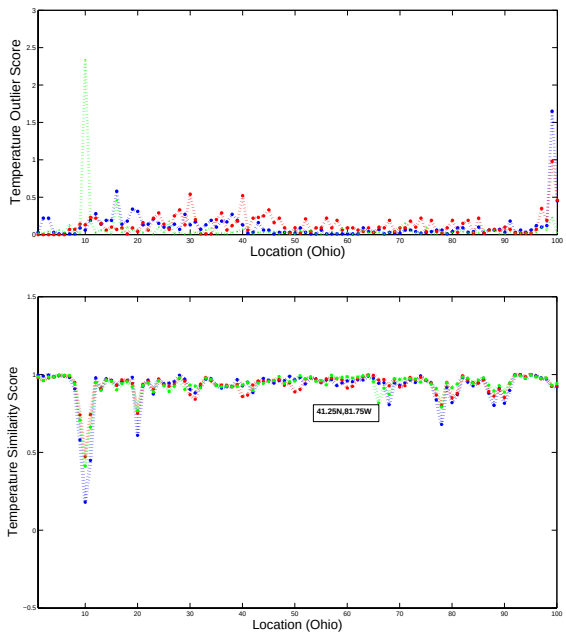



**Figure 9: Variation of distance-based outlier score (top) and neighborhood-based similarity score (bottom) of average January temperature in Ohio, USA for 2004(green), 2005(red) and 2006(blue).**

## 8. ACKNOWLEDGMENT

The authors would like to thank Dr.Kanishka Bhaduri, MCT Inc., Nasa Ames Research Centre, CA for his valuable guidance.

## 9. CONCLUSIONS

In this paper, we have used data mining algorithms for analyzing the spatial, temporal and spatio-temporal nature of Earth Science data. We have reduced data dimensionality as a preprocessing step and then identified outliers in the global climate system. The detected outliers not only stand for abrupt changes in the global climate over the years, but also explain some of the extreme events like drought and severe rainfall at specific locations on Earth. The results from our exclusive temporal and spatial analysis of the global climate data are fairly well-connected and reasonably aligned to the records of climate history. The spatial analysis of the outliers facilitates the discovery of interesting teleconnection patterns as well. In the future, we intend to extend our approach for determining spatio-temporal anomalies for handling growing region at reduced communication cost.

## 10. REFERENCES

[1] Nabil R. Adam, Vandana Pursnani Janeja, and Vijayalakshmi Atluri. Neighborhood based detection of anomalies in high dimensional spatio-temporal sensor datasets. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, 2004.

[2] The ORNL 50-Year Re analysis Data Download Website. `http://www.ornl.gov/sci/knowledgediscovery/SensorKDD-2009/challenge.htm`.

[3] Stephen D. Bay and Mark Schwabacher. Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

[4] University of Delaware Climate Data Archives. `http://climate.geog.udel.edu/~climate/html_pages/download.html`.

[5] A.R. Ganguly and K. Steinhaeuser. Data mining for climate change and impacts. In *Data Mining Workshops, 2008. ICDMW '08. IEEE International Conference on*, 2008.

[6] http://en.wikipedia.org/wiki/ENSO. El niño-southern oscillation: Wikipedia entry.

[7] http://ggweather.com/enso/oni.htm. Enso years: A consensus list.

[8] http://www.wrcc.dri.edu/enso. El niño, la nina and the western u.s., alaska and hawaii.

[9] Robert Kistler, Eugenia Kalnay, William Collins, Suranjana Saha, Glenn White, John Woollen, Muthuvel Chelliah, Wesley Ebisuzaki, Masao Kanamitsu, Vernon Kousky, Huug van den Dool, Roy Jenne, , and Michael Fiorino. The ncepŰncar 50-year reanalysis: Monthly means cd-rom and documentation. *Bulletin of the American Meteorological Society*, 82, 2001.

[10] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *The VLDB Journal*, 8, 2000.

[11] Vipin Kumar. High performance data mining - application for discovery of patterns in the global climate system. *Book Series Lecture Notes in Computer Science*, 4873, 2007.

[12] Fan Lin, XingXing Jin, Cheng Hu, XiaoPing Gao, Kunqing Xie, and XiaoFeng Lei. Discovery of

teleconnections using data mining technologies in global climate datasets. *Data Science Journal*, 6, 2007.

[13] AP Newspaper Report. `http://www.breitbart.com/article.php?id=D95NOGL80&show_article=1`.

[14] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29, 2000.

[15] Gao Shiying, Wang Jingshu, and Ding Yihui. The triggering effect of near-equatorial cyclones on el niño. *Advances in Atmospheric Sciences*, 5, 1988.

[16] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Conditional anomaly detection. *IEEE Trans. on Knowl. and Data Eng.*, 19, 2007.

[17] Michael Steinbach, Pang-Ning Tan, Vipin Kumar, Steven Klooster, and Christopher Potter. Discovery of climate indices using clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003.

[18] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Steven Klooster, Christopher Potter, and Alicia Torregrosa. Finding spatio-termporal patterns in earth science data: Goals, issues and results. *Temporal Data Mining Workshop, KDD*, 2001.