

Xmap: Fast Dimension Reduction Algorithm for Multivariate Streamline Data

Byung-Hoon Park[†], Nagiza F. Samatova[‡], George Ostrouchov and Al Geist
Computer Science and Mathematics Division,
Oak Ridge National Laboratory[‡],
P.O. Box 2008, Oak Ridge, TN 37831-6367

Abstract

Analyzing continuous flow-in data has gained significant attention lately. This new data model — often called data streams — includes scientific simulation output, satellite images, financial data, Web logs, network traffic data, etc. Processing data streams presents both practical and theoretical challenges; it often requires immediate results as streams flow in, and data size easily grows too fast to handle. This paper considers dimension reduction of a data stream. It particularly introduces a novel dimension reduction algorithm called *Xmap*, which exploits the connection between FastMap and the convex hull of data in Euclidean space. The paper shows that Xmap efficiently identifies a small representative subset of the past data and maintains well-approximated lower dimensional map at any given time.

1 Introduction

Dimension reduction techniques begin with n objects as points in a d -dimensional vector space and map the objects onto n points in a k -dimensional vector space, where $k < d$. A more general situation arises when the point coordinates are not known and only pairwise distances (or a distance function to compute them) are available. This mapping of objects based on their distances only into a k -dimensional vector space is called finite metric space embedding [6].

High dimensionality of data is one source of problems that challenges data practitioners. Visualization requires mapping the data to two or three dimensions. When the number of dimensions reaches thousands (or, even hundreds), visualizing two or three dimensions becomes combinatorially impossible. At the same time, many high dimensional data sets also include insignificant or irrelevant features, which often deteriorate the performance of many data mining processes like clustering [8] and classification [12].

In numerous scientific disciplines, data is naturally generated as a stream that is often very high dimensional. One good example can be found in the climate modeling community. A climate model deals with a series of outputs that are generated over an extremely long duration of a simulation process. The outputs, which include simulated values of numerous climate variables over a large number of grid points around the globe, are then used to understand complex climate processes.

Reducing the dimensionality of such a data stream is intrinsically difficult. The volume of data easily grows into almost unmanageable amount. For example, a simulation of a high-resolution ocean model generates data at an average of 2MB/sec [2] and typically runs for months. It essentially precludes any naive monolithic approach that regenerates the output from the ground up every time it observes new chunk of data from the source. Therefore, reducing dimensionality from such a high volume data stream calls for more efficient and scalable approaches.

In this paper, we propose a novel approach called *Xmap* that achieves dimension reduction from a stream of high dimensional multivariate data. Based on our recent observation that outputs of the FastMap heuristic

[†]Corresponding authors {parkbh,samatovan}@ornl.gov

[‡]Oak Ridge National Laboratory is managed by UT-Battelle for the LLC U.S. D.O.E. under Contract No. DE-AC05-00OR22725.

[5] are vertices of the convex hull around the original data [10], Xmap effectively identifies a representative subset of a data stream, which is substantially small in size. It then dynamically updates the representative subset as more data flow in from the stream source, and produces a projection map (or, lower dimensional representation) at any given time.

The paper is organized as follows. In Section 2, we present some background material. In particular, the convex hull connection to the FastMap heuristic is introduced. In section 3, Xmap is described in detail. Some observations from empirical studies are presented in section 4. Finally, section 5 concludes the paper with a discussion on future directions.

2 Background

This section presents some background materials of the paper. First, we define some notation that will be used throughout the paper.

2.1 Notation

We assume that the data stream is a series of blocks $\mathcal{S} = \{B_1, B_2, B_3, \dots\}$ (possibly endless) that are received at different time steps t_1, t_2, \dots , etc. — B_i is observed at time t_i . \mathcal{S}_n denotes a subset of \mathcal{S} that includes all the data blocks presented up-to t_n , i.e., $\mathcal{S}_n = \{B_1, B_2, \dots, B_n\}$.

O_a is a representation of a data point in some vector space (in our case, it is some Euclidean space) and \mathcal{H} is a hyperplane in the space under consideration. $d(O_a, O_b)$ and $\overline{O_a O_b}$ denotes the distance and the line between O_a and O_b , respectively. A pair of points (O_a, O_b) is called a *pivot* in some special case.

$\mathcal{C}(D)$ denotes the convex hull of data set D in d -dimensional Euclidean space, where d stands for the number of attributes of D . \mathcal{P}_i denotes a subset of points that are vertices of the convex hull $\mathcal{C}(\mathcal{E}_{i-1} \cup \mathcal{B}_i)$, where $\mathcal{E}_{i-1} = \cup_{k=1}^{i-1} \mathcal{P}_k$ and initially $\mathcal{E}_0 = \{\}$. We also use the term *extreme point* to denote a point $O_a \in \mathcal{P}_j$ at some time step t_j .

2.2 FastMap and Its Connection to the Convex Hull

FastMap is first introduced in [5] as a fast alternative to Multidimensional Scaling (MDS) [11] and a generalization of Principal Component Analysis (PCA) [7]. MDS is a finite metric space embedding method and PCA is a popular dimension reduction method. The context of [5] is similarity searching in multimedia databases. Given dimension k and Euclidean distances between n objects, FastMap maps the objects onto n points in k -dimensional Euclidean space. An implicit assumption by FastMap that the objects are points in a d -dimensional Euclidean space ($d \geq k$) is noted in [6]. Because of this assumption, FastMap is usually viewed as a dimension reduction method.

Given the Euclidean distance between any two points (objects) of a d -dimensional data set D , k iterations of FastMap produce a k -dimensional ($k \leq d$) representation of D . Each iteration selects from D a pair of points (or, pivots) that define an axis and computes coordinates of the D points along this axis. The pairwise distances for D can then be updated to reflect a projection of D onto the subspace (a hyperplane passing through the origin) orthogonal to this axis. The next iteration implicitly operates on the projected D in the subspace. However, these projections are accumulated and jointly performed only for the distances that are needed. In this manner, after k iterations, the D points end up with k coordinates giving their k -dimensional representation.

Pivot elements are chosen by the *choose-distant-objects* heuristic shown in Fig. 1. Initially, $i = 0$. After selecting a pivot pair (O_{a_i}, O_{b_i}) for the i -th iteration, the i -th coordinate of each point $O_x \in S$ is computed as

$$x_i = \frac{d_{i-1}^2(O_{a_i}, O_x) + d_{i-1}^2(O_{a_i}, O_{b_i}) - d_{i-1}^2(O_{b_i}, O_x)}{2d_{i-1}(O_{a_i}, O_{b_i})}. \quad (1)$$

where $d_i(O_x, O_y)$ is the Euclidean distance between points O_x and O_y after their i -th projection onto a pivot-defined hyperplane. This projection is based on the law of cosines and current distances from the two

pivot points. The distances are updated whenever needed in *Choose-distant-objects* or in (1). An update for a single iteration is presented in [5] and we extend this in [1] to a combined update

$$d_i^2(O_x, O_y) = d_0^2(O_x, O_y) - \sum_{j=1}^i (x_j - y_j)^2. \quad (2)$$

This is based on the Pythagorean theorem and the sequence of i projections onto hyperplanes perpendicular to pivot axes.

After choosing a pair of vertices, FastMap projects the set D into a subspace orthogonal to the vector defined by the pivot pair (O_a, O_b) and repeats the *Choose-Distant-Objects* heuristic in the subspace of dimension $d - 1$. Pivot pairs and projections are computed until suitably many orthogonal vectors are extracted to be used as the principal axes of the lower dimensional representation of D . It is not difficult to show that a pivot pair is a pair of convex hull vertices within its current working subspace. Are they all also vertices of convex hull $\mathcal{C}(D)$ in the original space? The answer is yes, subject to a uniqueness caveat requiring that no pair of points (except the current pivot points) get projected onto the same point. Assuming that the points D are in sufficiently *general position* [13] takes care of this. Because we have a finite set of points, we can perturb them by an arbitrarily small amount to achieve such a general position. This observation leads us to the following lemma [10].

Lemma 1 *All FastMap pivot pairs of a data set D are a subset of the vertices of $\mathcal{C}(D)$, the convex hull of the data.*

3 Extreme Points Driven Reduced Map

This section describes the proposed eXtreme points driven reduced **Map** (Xmap). Given a new data block B_i from a data stream \mathcal{S} , Xmap computes and updates the projection map for \mathcal{S}_i (i.e., the entire data observed by t_i). Internally, Xmap adopts the heuristic of FastMap when it chooses a set of pivots from data; it selects a pair of points that are possibly farthest apart. Then, how do we ensure that Xmap successfully finds pivots of \mathcal{S}_i , when it is observing B_i only? Xmap finds its solution from the connection between pivot sets and the convex hull that is introduced in the previous section.

3.1 Expanding Boundary of a Data Stream

The FastMap heuristic sifts out k pairs of extreme points from a d -dimensional data D , where $k < d$. Since these extreme points are vertices of the convex hull $\mathcal{C}(D)$ (see Section 2.2), they can be considered as an approximate boundary of D . The intuition behind Xmap is to expand the boundary of a data stream \mathcal{S} as it receives a new block B_i . Since the FastMap heuristic always seeks for pairs of points that are farthest apart, the well-preserved boundary will most likely produce a well-approximated set of pivots.

Xmap dynamically expands the boundary of a data stream \mathcal{S} as a new block B_i is presented at time t_i . The boundary that Xmap maintains is a set of extreme points denoted as \mathcal{E}_i , where i stands for the time step. Xmap starts by producing the first pivot set \mathcal{P}_1 from B_1 and puts it into \mathcal{E}_1 , i.e., $\mathcal{E}_1 = \mathcal{P}_1$. Then the

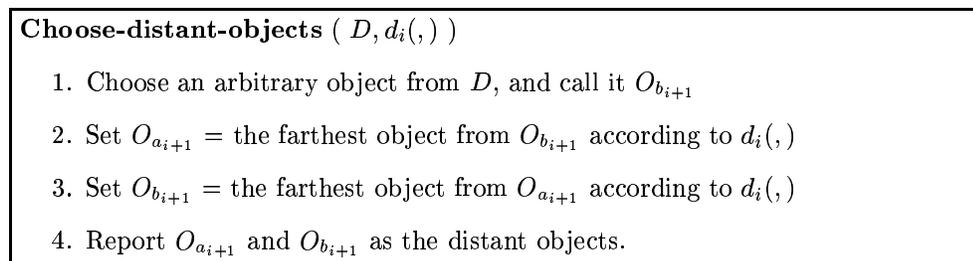


Figure 1: Choose-distant-objects heuristic for iteration i .

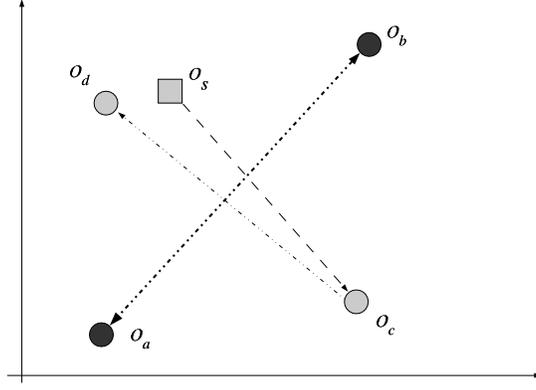


Figure 2: The situation when (O_c, O_d) is detected as the farthest pair with a choice of O_s as a seed, although (O_a, O_b) is farthest apart.

next pivot set \mathcal{P}_2 is extracted from $\mathcal{E}_1 \cup B_2$ and \mathcal{E}_2 is produced by adding \mathcal{P}_2 to \mathcal{E}_1 . Subsequently, \mathcal{E}_i is appended to B_{i+1} and, again, \mathcal{P}_{i+1} is extracted and \mathcal{E}_{i+1} is produced by adding \mathcal{P}_{i+1} to \mathcal{E}_i . From \mathcal{P}_i , the approximated reduced map (a set of k orthogonal projection vectors) of \mathcal{S}_i is obtained.

To ensure that the extreme point set \mathcal{E}_i evolves to expand the boundary properly, Xmap uses \mathcal{E}_{i-1} as seeds to extract \mathcal{P}_i given B_i . When choosing a pivot at each iteration, a poor selection can be made depending on a seed point (See step 1 of Figure 1). This instability of the FastMap heuristic can result in failing to properly expand the boundary. For example, let us assume that a point $O_a \in \mathcal{P}_{i-1}$ and $O_b \in B_i$ are farthest apart in some dimension under consideration. For (O_a, O_b) to be chosen as a pivot, either O_a or O_b should be farthest apart from the initial seed point O_s . However, since O_s is chosen at random, neither O_a nor O_b may be the farthest point from O_s . This situation is illustrated in Figure 2. Xmap remedies this problem by driving \mathcal{E}_{i-1} to be chosen as seeds in addition to a random seed from B_i .

```

Xmap( $\mathcal{S}$ )
begin
   $\mathcal{E}_0 = \{\}$ 
  for each data block  $B_i \in \{B_1, B_2, \dots\}$  from  $\mathcal{S}$ 
    1. apply FastMap heuristic to  $B_i \cup \mathcal{E}_{i-1}$  with  $\mathcal{E}_{i-1}$  as additional seeds
    2. produce  $\mathcal{P}_i$  as the reduced projection map at  $t_i$ 
    3.  $\mathcal{E}_i = \mathcal{E}_{i-1} \cup \mathcal{P}_i$ .
  end
end

```

Figure 3: Xmap

3.2 Including Missing Exterior Points Using Hyperplane

Although \mathcal{P}_i is a subset of vertices of the convex hull of $B_i \cup \mathcal{E}_{i-1}$, it may potentially exclude many extreme points (or, other vertices of the convex hull) due to its limited coverage. This section describes an approach that helps to identify such missing extreme points.

Given the $(i+1)$ -th pivot points $O_{(i+1)}^a$ and $O_{(i+1)}^b$, let us consider how the $(i+1)$ -th projection vector is obtained from the i -th projection vector. For the sake of simplicity, we assume that both $O_{(i+1)}^a$ and $O_{(i+1)}^b$ are already mapped to the first $i-1$ projection vectors in a proper way. As depicted in Figure 4, the $(i+1)$ -th projection vector is chosen by moving $O_{(i+1)}^a$ to $O_{(i+1)}^b$ along a direction parallel to the i -th projection

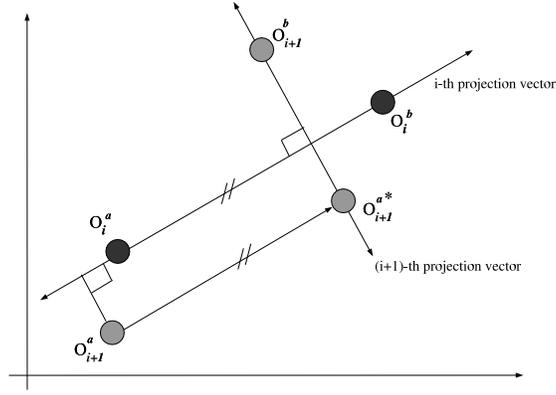


Figure 4: The $(i + 1)$ -th projection vector is obtained by translating O_{i+1}^a to O_{i+1}^b along a direction that is parallel to the i -th projection vector.

vector. Note that $\overline{O_{(i+1)}^a O_{(i+1)}^b}$ may not intersect $\overline{O_{(i)}^a O_{(i)}^b}$ when $O_{(i+1)}^a$ and $O_{(i+1)}^b$ are translated to lie in the $(i + 1)$ -th projection vector. However, it is straightforward to translate $O_{(i+1)}^a$ and $O_{(i+1)}^b$ along the i -th or $(i + 1)$ -th projection vectors so that they intersect $\overline{O_{(i)}^a O_{(i)}^b}$. Let us denote $O_{(i+1)}^{a*}$ and $O_{(i+1)}^{b*}$ are such translated points of $O_{(i+1)}^a$ and $O_{(i+1)}^b$ (In this regard, $O_{(i)}^a$ and $O_{(i)}^b$ are in fact $O_{(i)}^{a*}$ and $O_{(i)}^{b*}$, respectively). Now let us consider the following lemma.

Lemma 2 Consider a set of k pairs $\mathcal{P} = \{(O_{(1)}^{a*}, O_{(1)}^{b*}), (O_{(2)}^{a*}, O_{(2)}^{b*}), \dots, (O_{(k)}^{a*}, O_{(k)}^{b*})\}$, where $O_{(i)}^{a*}$ and $O_{(i)}^{b*}$ are points in the i -th projection vector, and are translated from the i -th pivot points $O_{(i)}^a$ and $O_{(i)}^b$. Let us further assume that each $\overline{O_{(i)}^{a*} O_{(i)}^{b*}}$ intersects every other $\overline{O_{(j)}^{a*} O_{(j)}^{b*}}$, where $1 \leq i \neq j \leq k$. Consider a set of k points $\mathcal{F}_j = (O_{(1)}^c, O_{(2)}^c, \dots, O_{(k)}^c)$, where each $O_{(i)}^c$ is either $O_{(i)}^{a*}$ or $O_{(i)}^{b*}$ of the i -th pair in \mathcal{P} . Then $\{\mathcal{F}_j\}$ forms a convex polytope that has at most 2^k facets.

Proof: If all points in \mathcal{P} are distinct, the convex hull of this set is a k -dimensional crosspolytope [13, page 8], which has 2^k facets. Each facet is a simplex on a set of k vertices from \mathcal{F} . If any points are not distinct, then some facets are a face of lower dimension. Since i -th projection of any point $O_x \in \mathcal{P}$ lies in $\overline{O_{(i)}^{a*} O_{(i)}^{b*}}$, each \mathcal{F}_j is essentially a system of k constraints (linearly independent if points are distinct) defining a supporting hyperplane [13]. Therefore $\{\mathcal{F}_j\}$ forms a convex polytope with at most 2^k facets. ■

Lemma 2 illustrates that the polytope based on \mathcal{P} is a convex set. In other words, \mathcal{P} itself forms a convex boundary. Therefore points that lie outside the boundary of \mathcal{P} are good candidates for extreme points that are excluded in selecting \mathcal{P} . According to Lemma 2, such points can be efficiently sifted by testing each point against 2^k hyperplanes. Given such a hyperplane \mathcal{H} , if a point O_a lies opposite halfspace that includes \mathcal{P} , then O_a is identified as an exterior point to \mathcal{P} . In this case O_a becomes a candidate for an extreme point. A pictorial description in 2-d space is shown in Figure 5.

4 Empirical Study

We seek to compare the performance of Xmap with that of a monolithic FastMap, i.e., the one that recomputes the projection map from the ground up each time a new block is presented. Since our objective is to preserve distances in a reduced dimension, we use the following well-known stress function in the comparison.

$$\text{stress} = \sqrt{\frac{\sum_{O_a, O_b} (d'(O_a, O_b) - d_0(O_a, O_b))^2}{\sum_{O_a, O_b} d_0(O_a, O_b)^2}}$$

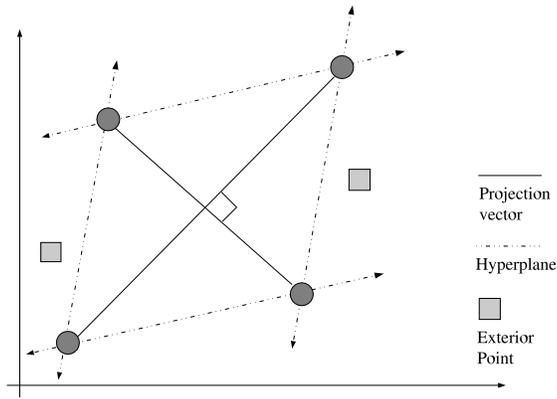


Figure 5: Exterior points detected by hyperplanes.

where $d_0(O_a, O_b)$ is the original distance between points O_a and O_b and $d'(O_a, O_b)$ is the distance between their images in the projected k -dimensional space. We refer the reader to [4] for a review of stress functions and their applications.

For a preliminary experiment, we use *waveform* and *pendigits* data sets from UCI repository of machine learning databases [3]. These sets have 5,000 and 3,498 data points and have original dimensions of 21 and 16, respectively. The data sets were split into 100 pieces and presented to Xmap one piece at a time, which essentially simulates 100 time steps. At each time step t_i , the pivot set \mathcal{P}_i is used to project all the data points presented up-to t_i . Then, stress value is computed and recorded. For the stress value of the monolithic FastMap, all the data blocks presented up-to t_i are used both for producing a pivot set and computing the stress value.

The experiment was performed for 3 different reduced dimensions, k . For each k , the experiment is repeated 50 times and averages are presented in Figure 6. For most cases, Xmap produced well-approximated reduced maps that also have consistent stress values over the entire period. However, when $k=3$, its performance is somewhat unstable (especially, for waveform data set). In that case, the maintained extreme point set may be too small to capture a proper boundary structure. However, since the monolithic approach also performed poorly in the same case, it needs to be further examined before drawing a conclusion.

We also monitor how the size of the extreme point set is increased over time. The objective is two-fold. First, for the Xmap to be useful in practice, the extreme point set should be small at any given time. In particular, if its size increases in a strictly linear fashion over time, it will become soon too large to handle. Second, if the boundary structure of the past data is successfully maintained and shape of data does not change abruptly with the new data blocks, the extreme point set should be maintained with small increases in size. Figure 7 illustrates this. Clearly, it is shown that the size of extreme point set is maintained small in every case.

5 Conclusion and Future Research

This paper discussed a fast dimension reduction of a high dimensional multivariate data stream, and proposed Xmap as one such effort. In contrast to a naive monolithic approach that is not scalable, it showed that Xmap effectively produces a projection map by maintaining a substantially small subset from a data stream. A preliminary empirical results illustrate that Xmap is also competitive to a monolithic approach in producing distance-preserved reduced maps.

Xmap is still in an early stage. It has to be further improved in several aspects. The size of extreme set can increase in a linear fashion in some special cases. It is particularly true when the distribution of data stream (or, how it is spread in some Euclidean space) changes abruptly over time. In such a case, how to maintain the extreme set in its optimally minimum size needs to be resolved. One approach under

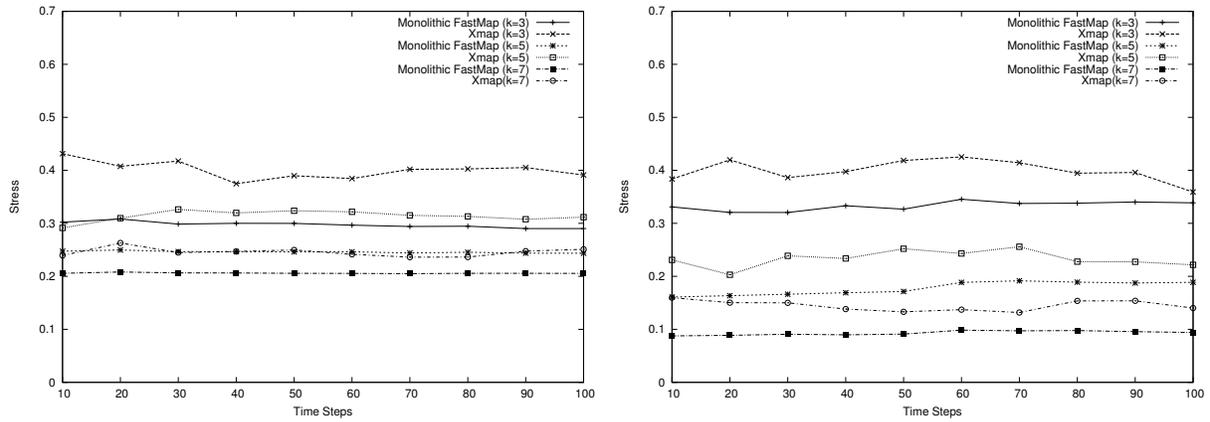


Figure 6: Stress values produced by Xmap and Monolithic FastMap from waveform (Top) and pendigit (Bottom) data sets.

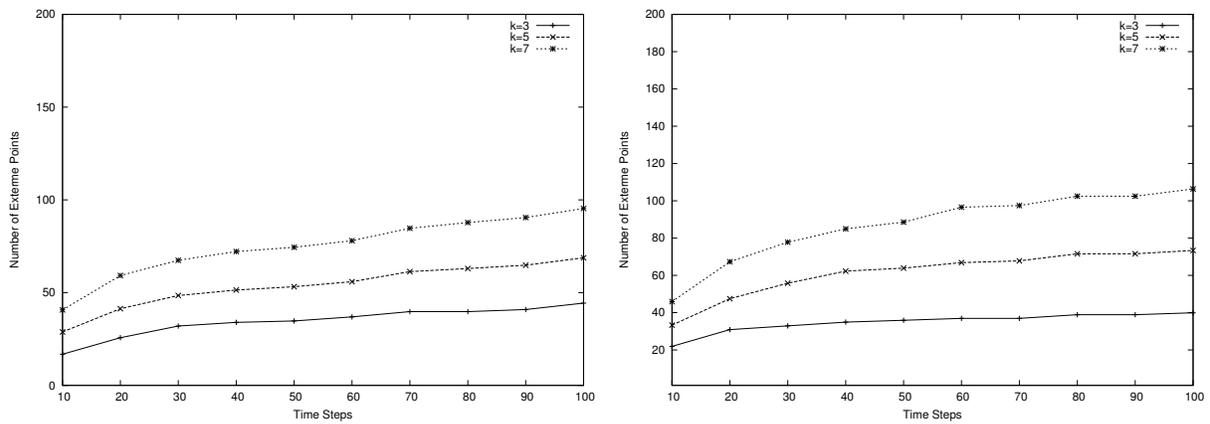


Figure 7: The number of extreme points that Xmap maintains over time from waveform (Top) and pendigit (Bottom) data sets.

development is to remove extreme points that fall inside all of the supporting hyperplanes at any given time.

Since the FastMap heuristic is intrinsically sensitive to outliers, Xmap may produce an undesired projection map. Currently, we are investigating a possibility to make the FastMap heuristic more robust and behave like principal component analysis [9]. An early investigation showed a positive result. However, further exploration will be left for future research.

Acknowledgments

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory (ORNL). This work was partially funded by the SciDAC program in the DOE Office of Advanced Scientific Computing Research. This research used resources of the Center for Computational Sciences at Oak Ridge National Laboratory. The submitted manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

- [1] Faisal N. Abu-Khzam, Nagiza F. Samatova, George Ostroouchov, Michael A. Langston, and Al Geist. Distributed dimension reduction algorithms for widely dispersed data. In *Parallel and Distributed Computing and Systems*. ACTA Press, 2002.
- [2] B. Allcock, B. Drach, D. Williams, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, and A. Shoshani. High-performance remote access to climate simulation data: A challenge problem for data grid technologies. In *2001 ACM/IEEE Conference on Supercomputing*, Denver, Colorado, 2001. ACM Press, New York, NY. USA.
- [3] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
- [4] T.T. Cox and M. A. Cox. *Multidimensional Scaling*, volume Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1994.
- [5] C. Faloutsos and K. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD Conference*, pages 163–174, San Jose, CA, May 1995.
- [6] Gísli R. Hjaltason and Hanan Samet. Contractive embedding methods for similarity searching in metric spaces. Technical Report CS-TR-4102, Computer Science Department, University of Maryland, College Park, MD 20742-3275, 2000.
- [7] Harlod Hotelling. Analysis of a complex of statistical variables into principal components. *J. Educ. Psych.*, 24:417–441, 498–520, 1933.
- [8] M. Jeon, H. Park, and J.B. Rosen. Dimension reduction based on centroids and least squares for efficient processing of text data. In *SIAM Conference on Data Mining*, Chicago, 2001.
- [9] I.T. Jolliffe. *Principal Component Analysis*, volume Springer series in statistics. Springer-Verlag, 1986.
- [10] G. Ostroouchov and N. F. Samatova. On fastmap and the convex hull of multivariate data: Toward fast and robust dimension reduction. In *communication*, 2003.
- [11] W. S. Torgerson. Multidimensional scaling i: Theory and method. *Psychometrika*, 17:401–419, 1952.
- [12] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning tools and techniques with Java Implementation*. Morgan kaufman, 1999.
- [13] Günter M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, 1995.