

# A Novel Effective Distributed Dimensionality Reduction Algorithm

Panagis Magdalinos, Christos Doulkeridis, Michalis Vazirgiannis  
Department of Informatics, AUEB, Greece  
{pmagdal,cdoulk,mvazirg}@aueb.gr

**Keywords:** Distributed dimensionality reduction, clustering, distributed knowledge discovery

## Abstract

Dimensionality reduction algorithms are extremely useful in various disciplines, especially related to data processing in high dimensional spaces. However, most algorithms proposed in the literature assume total knowledge of data usually residing in a centralized location. While this still suffices for several applications, there is an increasing need for management of vast data collections in a distributed way, since the assembly of data centrally is often infeasible. Towards this end, in this paper, a novel distributed dimensionality reduction (DDR) algorithm is proposed. The algorithm is compared with other effective centralized dimensionality reduction techniques and approximates the quality of FastMap, considered as one of the most effective algorithms, while its central execution outperforms FastMap. We prove our claims through experiments on a high dimensional synthetic dataset.

## 1. Introduction

Dimensionality reduction algorithms are extremely useful in various disciplines, especially related to data processing in high dimensional spaces. The latter becomes a difficult task as dimensions increase, because of the two distinct problems: the “empty space phenomenon” and “the curse of dimensionality” [1],[2]. The first denotes the fact that in high dimensional spaces data is sparsely situated, appearing at equal distance from one another. The “curse of dimensionality” on the other hand refers to the fact that in the absence of simplifying assumptions, the sample needed to estimate a function of several variables to a given degree of accuracy grows exponentially with the number of variables. A thorough investigation considering both aforementioned facts from the perspective of the nearest neighbor retrieval can be found in [20],[19].

Besides high dimensionality, another problem encountered in the area of data processing is the large amount of data. Data is not always situated on a single machine, but is usually scattered in a network. The latter is more obvious nowadays with the emergence of several novel applications such as peer-to-peer, sensor networks, data streams, etc. The ability to collect, store, process and subsequently index huge amounts of data has necessitated the development of algorithms that can extract useful information from distributed data corpuses. The scientific field of distributed knowledge discovery (DKD) addresses this issue. Distributed knowledge discovery is divided into two distinct categories. *Homogeneous*, where resources queried are arbitrarily distributed among nodes although described by the same features and *heterogeneous*, where all participants share the same knowledge but described by different features. Another possible categorization is acquired when information is considered as a huge resources x features matrix. If the rows or the matrix are shared among peers then the distribution is called horizontal (which is analogous to homogeneous) while the division of columns denotes a case of vertical distribution (equal to heterogeneous).

This paper proposes a novel effective dimensionality reduction algorithm that enables the compression of data processed, while retaining information for subsequent clustering or classification purposes. The algorithm proposed however exhibits the ability of distributed execution tackling the issue of distributed dimensionality reduction (DDR) from the perspective of

a distributed, homogeneous knowledge discovery problem. Despite the distributed nature of the approach, the reduction and indexing performance produced approximates the one exhibited by a well-known centralized algorithm, namely FastMap ([17]).

The paper is organized as follows: in Section 2 we review the related work regarding dimensionality reduction techniques. In Section 3, we identify the requirements for a distributed dimensionality reduction algorithm, while in Section 4, we present the novel algorithm. In Section 5, the experimental results are presented, and finally in Section 6, we conclude the paper.

## 2. Related Work

Each dimensionality reduction algorithm must fulfill some requirements in order to be considered effective and efficient. Briefly stated, the prerequisites are: a) the discovery of the intrinsic dimensionality of the dataset, b) the preservation of correlation dimensions between data, while projecting to a lower dimensionality space, and c) the least possible loss of information.

One of the initial methods proposed is the multidimensional scaling technique (MDS) often referred today as classic MDS (<http://www.statsoft.com>, <http://www.diap.polimi.it>). MDS is an explorative technique of data analysis that provides a depiction of the processed dataset in a lower dimensionality space with the usage of correlation information. In general, MDS can be considered as a methodology for dimensionality reduction proposing the use of numerical analysis transformations on data until a certain criterion is maximized or minimized.

The best dimensionality reduction approach is Principal Components Analysis [1], [2]. PCA achieves high stress minimization and high level of mutual information preservation. The algorithm applies singular value decomposition on the correlation matrix and retains only the  $k$  greatest singular values and vectors. In general, all singular value decomposition based methods exhibit high quality of results. Latent Semantic Indexing (LSI - [18]) is a special case, because the process utilized for the projection also manages to capture and bring forward semantic information contained in data. If the Stress criterion of MDS is replaced by the level of mutual information preservation the method in question is Independent Component Analysis [1],[2]. In the case of PCA, the use of the negative entropy function, as defined by Shannon, produces the Projection Pursuit method [2].

One of the fastest methods available in this area is FastMap [17]. FastMap maps data from dimension  $n$  to  $n-1$  by projection on a hyperplane perpendicular to the line defined by the two most distant points in the processed space. Recursive application of this procedure achieves the projection of  $N$  point from space  $R^n$  to subspace  $R^k$  in  $O(Nk)$  time while retaining distances among data. The Discrete Fourier Transform ([4] - DFT), is another method for fast projection and compression of data, which perceives each point as a series of randomly selected instances of a continuous signal and transforms it to a sum of basic signals. Afterwards, basic signals that do not add up to the final reconstruction are rejected; consequently, the corresponding coordinates are absconded thus resulting in the compression and reduction of data. PAA (Piecewise Aggregate Approximation) [4] is a close relative of DFT that projects each point independently from the rest. After fixing a window size  $f$ , all sets of  $f$  coordinates are replaced by their mean value. The main drawback of this fast approach ( $O(n)$ ) is its dependence on the size of the initial window. If the latter is big, then sharp changes in data will be lost, as all will be smoothed to their mean value.

All previously presented algorithms except from MDS, are classified as linear, because they try to project data in a globally linear space of lower dimensionality. On the contrary, non-linear methods try to preserve linearity in the locality of each point. By adding up the local linear fractals of

projection space one can achieve the formation of a non-linear projection space satisfying our requirements. Prominent methods employed for non-linear dimensionality reduction are the spring models [8], self organizing (Kohonen) maps [5], neural networks [1][2] and non-linear PCA [2]. The general idea of non-linear projection has recently steered much research in the field of dimensionality reduction. Isomap [8], C-Isomap [12] and Local Linear Embedding [11],[10] are relatively new methods for non-linear reduction. The most novel approach presented in bibliography is Landmark MDS or shortly LMDS [3]. The major goal of LMDS is the provision of a dimensionality reduction approach adequate for large datasets that cannot be loaded on main memory. The cost of this approach is  $O(2kbN + k^2N + b^3)$  ( $N$  being the cardinality of the projected set,  $b$  the number of landmark points and  $k$  the dimensionality of the projecting space) assuming that no heuristic is used. If a heuristic is employed for the selection of the initial points then a  $O(bN)$  factor is added to the aforementioned cost.

### **3. Requirements of a DDR Algorithm and Applicability of Centralized Algorithms**

The aim of this section is the identification of some initial requirements that a dimensionality reduction method must fulfill, in order to be used in distributed environments, along with an evaluation of the applicability of the previously described centralized algorithms in this context. Before dwelling in further analysis, some assumptions are stated. It is assumed that all resources can be described as points in a high dimensional space, i.e.  $R^n$ , while the latter is common to all participating nodes that form a network. Moreover no node can have global knowledge of the data/corpus being processed, but only a small fraction. Both assumptions analogize the problem to a horizontally distributed knowledge discovery problem.

Given a dimensionality reduction algorithm and a dataset of  $N$  resources, distributed arbitrarily among the nodes of a network, the following requirements must hold for the distributed execution of an algorithm: (1) Each point should be projected to the new subspace independently from the rest of the dataset.. (2) Distances between points should be preserved while projecting to a new subspace. The latter must hold true both locally and globally. Given two points  $A, B$ , their distance ( $d$ ) in the high dimension space, and their distance ( $d'$ ) in the projection space, the algorithm must guarantee that these values will be preserved even when the points belong to different network nodes. (3) The algorithm should be fast to compute, and linear to the total number of points projected.

The vast majority of dimensionality reduction techniques attempt to map points in a low dimensional space by exploiting the correlations among them. This is not tolerable in our case, because no node can acquire full knowledge of the data shared by the network. As an example, one can imagine the use of LSI, PCA and in general all SVD based methods. In the case of LSI or PCA, the abruption of certain singular values and singular vectors retains only the dimensions that provide the most valuable information regarding the correlation of the data, while discarded information is regarded as noise. There is no way to ensure however the validity of the comparison of two models generated by two different corpuses. The reason is rather simple and straightforward. Correlation dimensions initially perceived as noise and thus discarded, could carry valuable information, if SVD would have been carried out on the union of the two corpuses. Furthermore, SVD based methods, especially LSI, appear to have low scaling ability because of their complexity ( $N^3$ ,  $N$  being the size of the resources correlation matrix) and the fact that when vast amounts of data are processed it is not easy to distinguish noise from information.

One could argue however, that SVD is applicable in horizontally distributed data. Although this is the case, the cost of applying an SVD update algorithm is equal to the cost of re-calculating the decomposition [13], while the folding in technique (addition of data based on the assumption that

the decomposition is not influenced by new information) deteriorates quickly [14]. The Discrete Fourier Transform, although it satisfies the first and third requirement, discards dimensions in the depiction of the transformed signal, based on their significance. In our case, this would prevent even local comparison of data, because the discarded dimensions would differ among resources.

Only two of the presented methods can be applied in our case, LMDS and PAA. In the case of LMDS, a node can be arbitrarily chosen and assigned the task of reducing the initial points, which are provided by the rest. Afterwards, both projected and original data can be broadcasted across the network and each node may proceed independently. What the “adapted” LMDS achieves with high complexity and network traffic, PAA can achieve it with relatively no cost. The major drawback in this case is the size of the rolling window. If the latter is big (reduced dimensionality  $\ll$  original dimensionality) and the points are sparse then all variation will be lost.

#### 4. The Proposed Algorithm

An algorithm with lower complexity than LMDS and lower network traffic would be an adequate solution to our problem. The DDR algorithm presented in this section is an attempt to reach these standards, while fulfilling the requirements stated in the previous section. The approach follows the general principles of the LMDS adaptation, while differentiating in the way each step is achieved and exhibiting lower complexity and network traffic. The setup of the problem is the same. Given  $N$  resources represented as points of  $R^n$ , distributed arbitrarily in a network of  $p$  nodes, we want to find a projection of the data in  $R^k$ , while retaining distances and the ability to perform clustering afterwards. Each node is assumed to possess  $\lceil p/N \rceil$  resources. The algorithm is divided into four distinct steps.

**Step 1:** An aggregator node is selected. The selection can be made randomly or based on same kind of “built in” heuristic (i.e. a transformation of the IP address of nodes) as described in [16]. The aforementioned node is assigned all tasks that need to be executed centralized.

**Step 2:** Afterwards,  $k$  points must be sampled from the whole dataset and forwarded to the selected node. Each node selects and forwards  $\lceil k/p \rceil$  points resulting in  $O(nk)$  network traffic load. The selection can be made with one of the following ways:

- Each node randomly selects from the resources owned  $\lceil k/p \rceil$  points.
- Each node selects the  $\lceil k/p \rceil$  most far off points of its collection trying to create a kernel of points with long connections among them. We refer to this heuristic as MaxDist. The cost of the selection is  $O(\lceil k/p \rceil)$ , when random selection is employed, and  $O(\lceil N/p \rceil \lceil k/p \rceil)$ , when MaxDist is used.

**Step 3:** Selected points are projected by the aggregator in the  $R^k$  space with the use of the FastMap algorithm and all data (original coordinates of resources and projections) are flooded to the rest of the peers. The initialization of the FastMap algorithm needs  $O(k^2n)$  time and its execution  $O(k^2)$ , while the broadcasting of the result produces  $O(nk + k^2)$  network traffic.

**Step 4:** In the final step of the procedure, each node is obliged to project the resources owned to the new subspace with the use of the provided points (hereafter referred as *landmark points*). During the projection, the algorithm attempts to preserve distances, meaning that the resource projected must have equal distance from the landmark points both in the original and in the projection space. If  $x$  is the projected point,  $L$  the set of  $k$  landmark points and  $l_i$  the landmark points then this requirement is stated as  $\|x^{(k)} - l_i^{(k)}\| = \|x^{(n)} - l_i^{(n)}\|$  for  $i=1..k$ . The algorithm searches the common trace of all  $k$  hyperspheres, which is in fact the projection of point  $x$  in the reduced space. The result can easily be obtained by solving the above system of nonlinear equations with the Newton method.

If the approximation is precise, then the algorithm converges, otherwise the algorithm deviates and produces a result after the completion of a certain amount of iterations. This step produces on each node a load analogous to  $O(\lceil N/p \rceil \lceil k/p \rceil k^3/3)$ .

For any set of points the algorithm will produce a solution if the triangular inequality is sustained in the original space. For any point  $S$  of the initial space and the landmark points  $A, B$  equation  $\|AB^{\rightarrow}\| \leq \|SA^{\rightarrow}\| + \|SB^{\rightarrow}\|$  (1) holds true. The system defined for the projection ( $\|SA^{\rightarrow}\| = \|S'A'^{\rightarrow}\|$ ,  $\|SB^{\rightarrow}\| = \|S'B'^{\rightarrow}\|$ ) does not have a solution, if there exists no common trace between the created hyperspheres. This is translated as  $\|A'B'^{\rightarrow}\| \geq \|S'A'^{\rightarrow}\| + \|S'B'^{\rightarrow}\|$  or equally  $\|A'B'^{\rightarrow}\| \geq \|SA^{\rightarrow}\| + \|SB^{\rightarrow}\|$  (2) since  $\|SA^{\rightarrow}\| = \|S'A'^{\rightarrow}\|$  and  $\|SB^{\rightarrow}\| = \|S'B'^{\rightarrow}\|$  by default. After projecting  $A, B$  with FastMap the original and projected distances between these points are associated through inequality  $\|A'B'^{\rightarrow}\| \leq \|AB^{\rightarrow}\|$  (3). Consequently based on (3), (1) we conclude that equation (2) is never true, meaning that the system in question always has a solution (there always exists a projection) provided that the triangular inequality is sustained in the original space. Moreover, the time needed to compute this solution depends only on the approximation vector provided initially to the Newton method and the accuracy factor  $\epsilon$ .

To sum up, the proposed algorithm differs from other widely employed dimensionality reduction approaches for three distinct reasons. Initially, the projection of the vast majority of points is done independently from the rest, meaning that only the landmark points affect the projection. Moreover, landmark points remain unaffected by subsequent projections while the projection itself is independent of the sampled data. Finally the minimization criterion employed by the algorithm is  $\sum_{i \in I} \{|\text{distance}_{\text{orig}} - \text{distance}_{\text{new}}|\}$ , applied to each point independently, contrary to the widely employed Stress function that is applied to the whole set of data.

Compared to the distributed LMDS adaptation - also proposed in this paper- our algorithm exhibits lower network load and computational complexity. Indeed, distributed application of LMDS produces  $O(2bn + bk)$  network traffic and requires  $O(k^2 \lceil N/p \rceil + bk \lceil N/p \rceil)$  time for all nodes, while for the aggregator the load is  $O(k^2 \lceil N/p \rceil + bk \lceil N/p \rceil + b^3)$ . Note that  $b$  is larger than  $k$  in all cases and signifies the number of points selected for the execution of LMDS. On the other hand our algorithm produces  $O(2nk + k^2)$  traffic load and requires  $O(\lceil N/p \rceil \lceil k/p \rceil k^3/3)$  time. This value is augmented at the aggregator node for an amount of  $O(k^2)$  due to the execution of FastMap.

Apart from the lower complexity, the proposed algorithm comes with one more advantage against the distributed application of LMDS. The sampling procedure can be carried out once in the lifetime of a network and the result can be forwarded to all nodes entering the network at any time. Projection is independent of the sample, because each resource is projected to a point abstaining analogously far or close in the reduced space. On the contrary, since LMDS employs classic MDS that requires the solution of a generalized eigenvector problem, updates have to take place periodically, since content changes affect the projection.

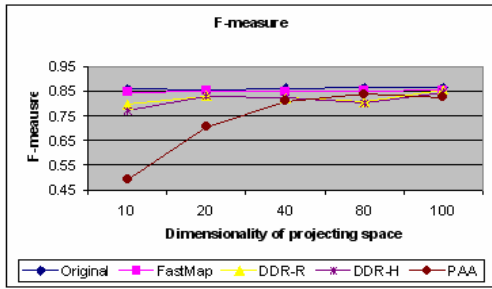
## 5. Experimental Results

In an attempt to evaluate the proposed algorithm, a series of experiments on a synthetic dataset was carried out. The goal was to prove the validity of the approach while exhibiting results of quality close to well-known centralized approaches. In all experiments, we arbitrarily created a set of high dimensional vectors, which constructed a set of ten well separated clusters, so as to ensure that the applicability of clustering is unaffected by the high dimensionality of the processed space. The clustering algorithm employed was K-Means.

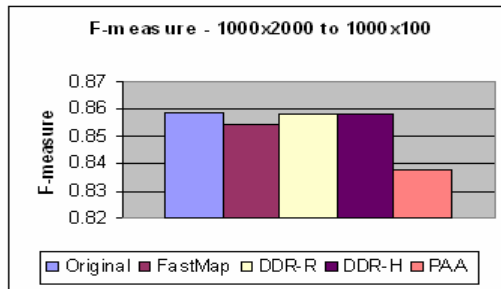
The data generator takes as input the number of vectors ( $s$ ), and the number of clusters ( $c$ ) to be created. All vectors coordinates are initialized by values belonging to  $[0,1]$ . At the second step the generator produces a set of  $c$  different integers ( $p_i, i=1\dots 10$ ). Finally, each set of  $\lceil s/c \rceil$  vectors changes the  $p_i$  coordinate of the elements contained to 5. This value ensures that each set of  $\lceil s/c \rceil$  vectors is well separated from the rest, meaning that no overlapping occurs between clusters.

The points were subsequently projected to a predefined lower dimensionality space through the usage of four different algorithms. The first algorithm, which has been used as a point of reference, was FastMap. Afterwards, two different setups of our new algorithm were employed. The first (named DDR-R) used a random sample of initial data, while the other (named DDR-H) employed the MaxDist heuristic. PAA was also tested in order to evaluate its stability and quality in large-scale reduction processes. Finally, K-Means was employed in order to evaluate the clustering quality after the reduction. The Newton method employed by our algorithm utilizes as an approximation vector the perpendicular projection of the point (referred to as  $x$ ) to the new subspace with every coordinate augmented by a factor  $(a^2-1)\|x\|$  ( $a=0.7$ )

Results presented in this paper come from the projection of 1000 vectors of dimensionality 2000 to dimensions 10, 20, 40, 80, 100. Due to space limitations, three more sets of experiments are omitted, but can be found in the extended version of this paper [21].



**Figure 1: Deviation of clustering quality**



**Figure 2: Outperforming FastMap in clustering quality maintenance**

As far as clustering is concerned, figure 1 gives valuable insight and allows us to draw some initial conclusions. With a sampling of only 2%-4% of data, high quality projection and clustering is achieved. F-measure is in fact less than 5% lower than the one achieved with centralized projection of the data (FastMap). Moreover, when projecting from initial dimensionality 2000 to 100 dimensions, both DDR-R and DDR-H outperform FastMap, as exhibited in figure 2.

Another interesting result is that the method is not influenced by the way the initial set of points are selected, allowing in fact the usage of random sampling and thus lowering the complexity of the process. The projection quality is measured by computing the stress value. All experiments exhibited the same behavior, producing a very low stress value, almost equal to the one exhibited by FastMap. Moreover, the mathematically proven fact that the stress value decreases as projection dimension increases was also demonstrated. Finally, the projection was unaffected by the way initial points were sampled. Figure 3 demonstrates these facts, while Figure 4 demonstrates the time requirements of all four approaches.

In all above experiments, our algorithm was executed in a distributed way, as described in the previous sections. However, one can also employ this algorithm in a centralized way. In this case, the best way to choose the initial points is the execution of a clustering algorithm, namely K-Means with the usage of HAC as initialization process. After the latter's completion, the number of

clusters generated is supplied to our algorithm as the projection dimension and the centroids calculated as the landmark points. When our algorithm was evaluated with the aforementioned setup, it outperformed FastMap, reaching even more than 10% better clustering quality, together with an extremely low stress value. Figures 5, 6 demonstrate this fact in the projection of 300 points from an initial dimension of 1000 to 10.

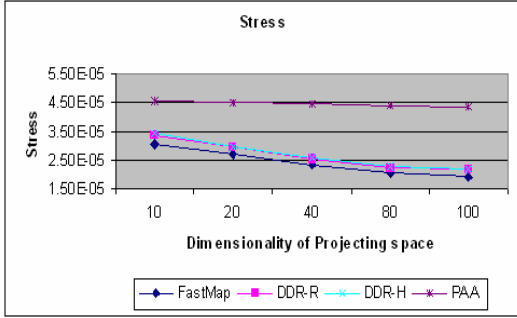


Figure 3: Projection quality

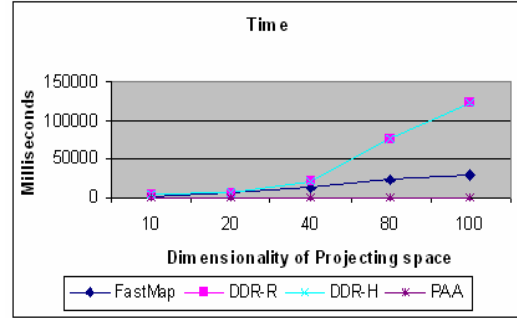


Figure 4: Time requirements

To sum up, the experimental evaluation presented in this section, leads to a primary conclusion stating that the proposed algorithm offers the possibility of distributed dimensionality reduction for large datasets providing projection quality equal to a centralized approach, namely FastMap. Furthermore, clustering the reduced data projected by our algorithm, retains high quality, marginally equal to the one achieved, when performing clustering in the original space (note that the initial clusters were well separated). Results obtained from clustering on the projections of the centrally executed FastMap, and our distributed executed algorithm exhibit the same quality. On the other hand, the use of the MaxDist heuristic does not ameliorate results. Finally, when our algorithm was used as a centralized dimensionality reduction approach and was evaluated against FastMap, it produced better quality results both in terms of F-Measure and Stress values.

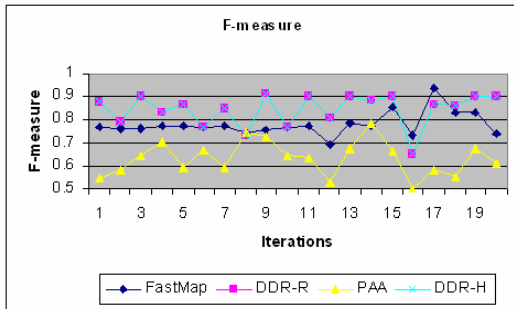


Figure 5: Clustering quality in centralized execution

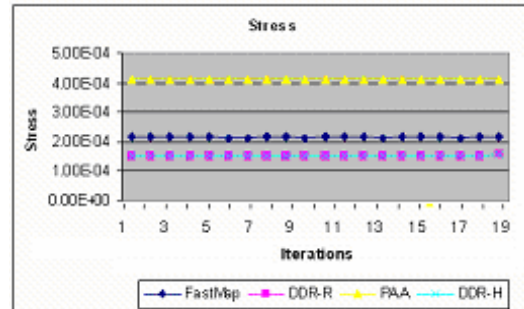


Figure 6: Stress in centralized execution

## 6. Conclusions and Future Work

This paper tackled the issue of distributed dimensionality reduction from the perspective of a distributed, homogeneous knowledge discovery problem. The bibliographic research indicated the absence of any appropriate solution to this problem. Furthermore, only one of the centralized approaches could be adjusted to fit our requirements. To the best of our knowledge, our approach and the distributed LMDS adaptation, both presented in this paper, are the first to provide a solution to this problem. However, our algorithm is the first approach that directly targets the problem of distributed dimensionality reduction. The quality of our results is almost equal to FastMap, measured in terms of Stress and F-measure values, while our algorithm's central execution

outperforms FastMap. Future work will primarily concentrate on evaluating our algorithm with real datasets against LMDS and PCA. The last comparison will demonstrate the viability of our approach against the best dimensionality reduction algorithm in the bibliography.

## 7. References

- [1]. "A Survey of Dimension Reduction Techniques", *I.K. Fodor*, US Department Of Energy, 2002
- [2]. "A Review of Dimension Reduction Techniques", *M.Carreira*, Technical Report, University of Stanford, 1997
- [3]. "Sparse Multidimensional Scaling Using landmark points", *Vin de Silva, Joshua B. Tenenbaum*, 2004
- [4]. "Dimensionality Reduction of Fast Similarity Search in Large Time Series Databases", *E.Keogh, K. Chakrabarti, M.Pazzani, S.Mehrotra*, Knowledge and Information Systems – Springer-Verlag, 2001
- [5]. "Self-Organization of Very Large Document Collection: State of the Art", *Teuvo Kohonen*, ICANN 1998
- [6]. "A Linear Iteration Time Layout Algorithm for Visualising High-Dimensional Data", *Matthew Chalmers*, 7<sup>th</sup> IEEE Visualization Conference, 1996.
- [7]. "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Sam T.Roweis, Lawrence K.Saul*, Science Magazine ([www.science.org](http://www.science.org)) 2000.
- [8]. "Unsupervised Learning Of Curved Manifolds", *Vin de Silva, Joshua B. Tenenbaum*, In Proceedings of the MSRI workshop on nonlinear estimation and classification. Springer Verlag, 2002.
- [9]. "Fast Multidimensional Scaling through Sampling, Springs and Interpolation", *Alistair Morrison, Greg Ross, Mathew Chalmers*, Information Visualization, Vol.2, Issue 1, 2003.
- [10]. "Think Globally, Fit Locally, Unsupervised Learning of Nonlinear Manifolds", *T.Roweis, Lawrence K.Saul*, Technical Report, 2002
- [11]. "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Vin de Silva, Joshua B. Tenenbaum, John C.Langford*, Science Magazine ([www.science.org](http://www.science.org)) 2000
- [12]. "Global versus local methods in nonlinear dimensionality reduction", *Vin de Silva, Joshua B. Tenenbaum.*, NIPS 2003
- [13]. "Information Management Tools for Updating SVD-encoded indexing schemes", *O'Brien*, Master Thesis, University of Tennessee, 1994
- [14]. "A Divide-and-Conquer Approach to the Singular Value Decomposition", *Jane Tougas*, Seminar on Machine Learning and Networked Information Spaces, University of Dajhousie, 2004
- [15]. "A Semidiscrete Matrix Decomposition for Latent Semantic Indexing in Information Retrieval", *Tamara G.Kolda, Dianne O'Leary*, ACM Transactions on Information Systems, Vol.16, No.4 October 1998.
- [16]. "DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks", *C.Doulkeridis, K.Noervaag, M.Vazirgiannis*. Technical Report, DB-NET, AUEB, 2006 (submitted for publication).
- [17]. "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets", *Christos Faloutsos, David Lin*, ACM SIGMOD 1995.
- [18]. "Indexing by Latent Semantic Analysis", *S.Deerwester, S.Dumais, T.Landauer, G.Fumas, R.Harshman*, Journal of the Society for Information Science, 1990
- [19]. "When is 'Nearest Neighbor' Meaningful?", *K.Beyer, J.Goldstein, R.Ramakrishan, U.Shaft*, ICDT 1999
- [20]. "What is nearest neighbor in high dimensional spaces?", *A.Hinneburg, C. Aggarwal, D.Keim*, VLDB 2000.
- [21]. "A Novel Effective Distributed Dimensionality Reduction Algorithm", *P.Magdalinos, C.Doulkeridis, M.Vazirgiannis*. Technical report, 2006, available at: [http://www.db-net.aueb.gr/index.php/publications/technical\\_reports](http://www.db-net.aueb.gr/index.php/publications/technical_reports)