Enhancing Clustering Quality through Landmark-based Dimensionality Reduction

PANAGIS MAGDALINOS
Athens University of Economics and Business, Greece
and
CHRISTOS DOULKERIDIS
Norwegian University of Science and Technology, Norway
and
MICHALIS VAZIRGIANNIS
Athens University of Economics and Business, Greece

Scaling up data mining algorithms for data of both high dimensionality and cardinality has been lately recognized as one of the most challenging problems in data mining research. The reason is that typical data mining tasks, such as clustering, cannot produce high quality results when applied on high-dimensional and/or large –in terms of cardinality– datasets. Data pre-processing and in particular dimensionality reduction constitute promising tools to deal with this problem. However, most of the existing dimensionality reduction algorithms share also the same disadvantages with data mining algorithms, when applied on large datasets of high dimensionality. In this paper, we propose a fast and efficient dimensionality reduction algorithm (FEDRA), which is particularly scalable and therefore suitable for challenging datasets. FEDRA follows the landmark-based paradigm for embedding data objects in a low-dimensional projection space. By means of a theoretical analysis, we prove that FEDRA is efficient, while we demonstrate the achieved quality of results through experiments on datasets of higher cardinality and dimensionality than those employed in the evaluation of competitive algorithms. The obtained results prove that FEDRA manages to retain or ameliorate clustering quality while projecting in less than 10% of the initial dimensionality. Moreover, our algorithm produces embeddings that enable the faster convergence of clustering algorithms. Therefore, FEDRA emerges as a powerful and generic tool for data pre-processing, which can be integrated in other data mining algorithms, thus enhancing their performance.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications—*Data Mining*; H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Clustering*; H.3.4 [**Information Systems**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Landmarks, Dimensionality Reduction, Clustering Quality

1. INTRODUCTION

An increasing number of contemporary applications produce massive volumes of very high-dimensional data. In scientific databases, for example, it is common to encounter large sets of observations, represented by hundreds or even thousands of coordinates. Unfortunately the rate of data generation and accumulation significantly outperforms our ability to explore and analyze it. Nevertheless, in order to extract knowledge from these datasets, we need to access the underlying, hidden information. However, the size and complexity of these collections makes their processing and analysis impractical or even ineffective [Beyer et al. 1999]. Therefore, scaling up data mining algorithms for data of

both high dimensionality and cardinality has been recently recognized as one of the top-10 problems in data mining research [Yang and Wu 2006].

A potential solution to this problem is provided by data pre-processing techniques and particularly dimensionality reduction. Dimensionality reduction addresses these challenges by projecting data from the original high-dimensional space to a new, lower dimensional space while retaining useful data properties such as pairwise distances or other statistical properties (i.e., variance). However, the vast amount of generated data dictates methods that are both fast and exhibit low memory requirements. Unfortunately, the vast majority of existing algorithms are either computationally efficient at the expense of high memory requirements or they require limited memory at the expense of significant computational cost.

The main focus and application area of our work is the enhancement of the quality of clustering algorithms for high-dimensional datasets of particularly high cardinality, by means of efficient and scalable dimensionality reduction. To this end, we propose *FEDRA*, a fast and efficient dimensionality reduction algorithm, that aims to address these challenges directly. FEDRA belongs to the family of *landmark-based* dimensionality reduction algorithms. The basic intuition is that k n-dimensional objects are selected as *landmarks* and they are embedded to a k-dimensional projection space, by retaining their exact pairwise distances. Then, the remaining objects are embedded in the k-dimensional space, by requiring that their distances to the landmarks are exactly retained.

FEDRA is computationally efficient without high memory requirements (compared to existing algorithms) and as demonstrated experimentally, achieves high quality results when applied to typical clustering tasks. In particular, FEDRA manages to successfully reproduce the original cluster structure in a space of dimensionality lower than 10% of the initial dimensions while the obtained embedding significantly accelerates the convergence rate of k-Means. We emphasize that our experimental evaluation employs significantly larger datasets than those used in the study of competitive state-of-the-art algorithms. The results verify the applicability of FEDRA on large-scale clustering tasks. The individual contributions of this work are summarized as follows:

(1) We provide a theoretic and experimental study of the family of landmark-based dimensionality reduction algorithms. Each algorithm is assessed with respect to its computational resources requirements as well as its applicability and viability in hard dimensionality reduction problems.

(2) We present FEDRA, a provably fast and efficient dimensionality reduction algorithm that follows the principles of landmark-based dimensionality reduction.

(3) We provide a thorough theoretical analysis of FEDRA that includes the calculation of computational complexity, the proof of embedding existence, an assessment of projection quality and a geometric interpretation.

(4) We propose two extensions of the basic algorithm, including an effective landmark selection heuristic as well as a heuristic for choosing the best embedding for a data object out of a set of possible embeddings.

(5) Inspired by our previous work [Magdalinos et al. 2006], we demonstrate the applicability of FEDRA in a widely distributed setting, where data is not available at one centralized location but instead it is horizontally fragmented over a set of independent nodes in the network.

Symbol	Description
n	Dimensionality of original space
k	Dimensionality of projection space
d	Cardinality of dataset
p_i or P_i	Data point
l_i or L_i	Landmark point
p'_i	Data point in the new space
$p_{i,j}$	The j-th coordinate of p_i
$d_p(p_i, p_j)$	Minkowski distance between p_i and p_j, $d_p(p_i, p_j) = (\sum_{l=1}^{n}(p_{i,l} - p_{j,l})^p)^{\frac{1}{p}}$
$d(p_i, p_j)$	Euclidean distance between p_i and p_j ($p = 2$)
$d_p(p_i, p_j)$	Distance between p_i and p_j in original space
$d'_p(p_i, p_j)$	Distance between p_i and p_j in projection space
ℓ_p^d	The d-dimensional Euclidean space with Minkowski distance metric p

Table I. Overview of basic symbols.

(6) By means of an extensive evaluation on large-scale datasets, we validate the behavior of FEDRA by comparing it against other state-of-the-art landmark-based dimensionality reduction techniques.

The rest of this paper is structured as follows: Section 2 provides a brief survey of related work in dimensionality reduction algorithms. In Section 3, we present FEDRA and describe in detail the embedding algorithm. Then, we provide an analysis of our theoretical findings in Section 4. The extensions of our basic algorithm are presented in Section 5. In Section 6, we demonstrate the results of the experimental evaluation. Finally, we conclude the paper and sketch future research directions in Section 7.

2. RELATED WORK

In this section, we provide an overview of the area of dimensionality reduction. We commence by providing a classification scheme for dimensionality reduction algorithms coupled with various metrics for the evaluation of their results. Furthermore, we outline the most dominant linear techniques and present in details a small subset, namely landmark-based methods that are prominent for their efficiency in terms of consumption of computational resources. Due to the emergence of distributed knowledge discovery, we report the latest results in the area of distributed dimensionality reduction and elaborate on the extensibility and applicability of state-of-the-art algorithms in distributed environments. The section concludes with a comparative assessment of the analyzed algorithms in terms of time and space requirements.

In the following, we assume that the dataset is composed of d data objects represented as points in the n-dimensional *original space* that are going to be embedded in a k-dimensional *projection space*, with k significantly lower than n ($k \leq 0.1n$). The Minkowski distance between two points p_i and p_j in the original space is depicted as $d_p(p_i, p_j)$, while $d'_p(p_i, p_j)$ denotes their distance in the projection space. In the case of the Euclidean distance we drop the subscript and simply use $d(p_i, p_j)$. For a complete overview of the basic symbols used in the following, we refer to Table I.

2.1 Classification Scheme and Quality Measures

2.1.1 *Classification Scheme.* Dimensionality reduction problems can be broadly classified into three distinct categories [Carreira-Perpinan 1997]. *Hard problems*, where data is defined in a space consisting of hundreds or even thousands of coordinates and drastic dimensionality reduction is required, possibly of orders of magnitude, *Soft problems*, where the requirement for reduction is milder, and *Visualization problems*, where data of high dimensionality is mapped to few dimensions, such that its structure becomes perceivable by humans. The algorithms that solve these problems are classified with respect to the way they manage data [de Silva and Tenenbaum 2002]. *Linear algorithms* embed any object in the identified low-dimensional space by deriving a linear combination of its coordinates. This procedure implies that high-dimensional data lay on an approximately linear manifold of significantly lower dimensionality. On the other hand, *non linear methods* assume that such global linearity does not exist and operate on small fractions of the high-dimensional manifold that can be perceived as locally linear. If we consider the dimensionality of the projection space, then reduction methods are distinguished between *global* and *local* [Lian and Chen 2009]. *Global methods* embed data in a common low-dimensional space while *local methods* project small data partitions to a dimensionality which is calculated by the corresponding partition's local statistics. Finally, depending on whether or not the pairwise distances of points are exactly retained in the projection space, dimensionality reduction algorithms can also be classified as *approximate* or *exact*. In the context of this work we will primarily focus on the family of approximate, linear, global dimensionality reduction methods and specifically on one of its subsets, namely landmark-based algorithms.

2.1.2 *Quality Metrics.* We also provide an overview of appropriate quality metrics for the evaluation of dimensionality reduction algorithms.

Distortion. While there exist different methods for assessing the quality of an algorithm, the most popular metric is *distortion* [Hjaltason and Samet 2003]. Distortion quantifies the change in the distance between any two points p_i,p_j due to the projection and is defined as the lowest c_1c_2 value with c_1,$c_2 > 1$, which guarantees that:

$$\frac{1}{c_1} d_p(p_i, p_j) \leq d'_p(p_i, p_j) \leq c_2 d_p(p_i, p_j) \tag{1}$$

Stress. Distortion implies the existence of theoretic upper and lower bounds to the distance deviation induced by an algorithm. However, the derivation of explicit bounds may not be possible for some algorithms, while others may exhibit worse theoretic bounds compared to their actual behavior. In such cases, an application-oriented metric like *stress* is employed. Stress quantifies the capability of an algorithm to approximate the original pairwise distances, by comparing the original set of distances with the one obtained in the projection space. Stress is calculated by formula 2.

$$Stress = \sqrt{\frac{\sum_{i=1}^{d} \sum_{j=1}^{d} (d_p(p_i, p_j) - d'_p(p_i, p_j))^2}{\sum_{i=1}^{d} \sum_{j=1}^{d} d_p(p_i, p_j)^2}} \tag{2}$$

Task-related Metrics. Another approach of indirectly assessing the quality of a dimensionality reduction algorithm is to compare the performance of a data mining task (i.e., clustering or classification) prior and after the application of dimensionality reduction.

Typical examples include the *Clustering Preservation Ratio* (CPR), the *Relative Classification Ability Maintenance* (RCAM) and the *Relative Clustering Disability Degradation* (RCDD). CPR [Gabriela and Martin 1999] validates an embedding with respect to its ability to maintain the original cluster structure. Assuming that data labels are known in advance, CPR employs a nearest neighbor classification scheme and measures how many cluster labels have changed due to the repositioning of objects by the projection. In the same spirit, RCAM and RCDD [Magdalinos et al. 2009] quantify the amelioration of the performance of classification and clustering algorithms, due to dimensionality reduction. Finally, for the specific case of nearest neighbor retrieval, the *Pruning Power* (PP) and the *Computational Cost* (CC) metrics [Lian and Chen 2009] can be employed. PP measures the number of objects that are pruned in the low-dimensional space using the triangle inequality without introducing false dismissals while CC measures the number of distance computations that take place in the original high-dimensional space, after the completion of the pruning phase.

2.2 Prominent Dimensionality Reduction Techniques

Dimensionality reduction can be simply viewed as a transformation that embeds data in a low-dimensional space. One of the key issues however is the definition of the corresponding transformation matrix. The latter is accomplished with the use of linear algebra techniques which operate in the heart of most algorithms. Eigendecomposition, QR factorization and Singular Value Decomposition (SVD) comprise such examples [Stewart 2001].

One of the initial dimensionality reduction methods is multidimensional scaling (MDS) often referred to as classic MDS [Togerson 1958]. MDS embeds data in a low-dimensional space by projecting on a space spanned by the eigenvectors that correspond to the k largest eigenvalues of the data cross product matrix, XX^T. Principal Components Analysis (PCA) [Chakrabarti 2002] is a closely related method to MDS that derives the corresponding eigenvectors from the data covariance matrix, $\frac{1}{n}\overline{X}^T\overline{X}$, where $\overline{X}$ signifies data matrix X with means subtracted across dimensions. MDS requires $O(d^3)$ space and $O(d^2)$ time while PCA $O(n^3)$ and $O(n^2)$ respectively. Linear Disciminant Analysis (LDA) [Swets and Weng 1996] is a technique closely related to PCA, in the sense that they both project points on a set of axis that best discriminate the data. However, contrary to PCA that maximizes data covariance, LDA attempts to best discriminate data classes. The poor scaling ability of classic LDA in conjunction with its high quality results inspired the definition of many alternatives such as Nonparametric LDA [Li et al. 2009], Rotational LDA [Sharma and Paliwal 2008] and QR-based LDA [Ye et al. 2004]. Like PCA, LDA requires $O(n^3)$ time and $O(n^2)$ space. The direct application of SVD on X resulted in methods such as Correspondence analysis (CA) [Payne and Edwards 1999] and Latent Semantic Indexing (LSI) [Deerwester et al. 1990].

Due to the fact that the eigenanalysis and singular value decomposition of a matrix are quite expensive in terms of computational resources, numerous methods have attempted to approximate their results. Instead of using the leading singular vectors of the original data matrix A, in [Drineas et al. 2006] the authors choose directly a subset of columns and rows from the actual dataset and perform an approximation of the original data matrix through multiplication CUR. C and R are matrices populated with columns and rows from A respectively while U is defined as a product of C,R and A. Despite its simplicity, this method yields results of high quality since it induces a bounded error slightly larger than that of

SVD. Apart from its simplicity, the key properties of CUR decomposition are its low time and space requirements; assuming the selection of k rows and columns, time complexity is upper-bounded by $O(k^3)$ while memory load reaches $O(d+n)$. By introducing a new row and column selection procedure [Mahoney and Drineas 2009], the authors managed to further reduce the approximation error without additional computational load, thus further ameliorating the quality of the decomposition.

Trying to solve the high time requirements of MDS, Faloutsos and Lin introduced FastMap [Faloutsos and Lin 1995]. FastMap is an alternative to MDS which employs elementary Euclidean geometry and achieves high quality results with considerably lower time requirements $O(dk)$. A significant drawback of FastMap is its memory requirements, which reach $O(d^2)$. The latter is partially addressed in a variation of the algorithm that takes as input the original points, thus losing its dimensionality agnostic nature. Then, memory requirements are reduced to $O(d(k+n))$, however computational complexity rises since it requires computation of high-dimensional distances[1].

A powerful yet extremely simple and computationally efficient dimensionality reduction method is Random Projection. Random Projection comprises a special case of *data oblivious* technique [Ailon and Chazelle 2010] since, contrary to almost all other approaches, defines a transformation matrix without using any direct or indirect information from the underlying dataset. Indeed, data points are embedded in R^k with the use of a randomly generated matrix (R_{kxn}) through multiplication $\frac{1}{\sqrt{k}}XR^T$. The idea of the projection is based on the Johnson-Lindenstrauss lemma.

Lemma 1. **Johnson-Lindenstrauss:** For any $0 < \varepsilon < 1$ and any integer d, let k be a positive integer such that $k \geq 4(\varepsilon^2/2 - \varepsilon^3/3)^{-1} lnd$. Then for any set V of d points in R^n there is a map $f : R^n \rightarrow R^k$ such that for all $u, v \in V : (1-\varepsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\varepsilon)\|u-v\|^2$. Further, this map can be found in randomized polynomial time.

An elementary proof of this lemma is provided in [Dasgupta and Gupta 2003]. Since the embedding of a dataset from ℓ_2^n to ℓ_p^k ($p \geq 1$) is acquired through a matrix multiplication procedure, time requirements are upper-bounded by $O(dkn)$. Memory requirements are low, $O(kn)$, since the algorithm requires only the constant existence of the random matrix in main memory. Addition of a new point results in $O(kn)$ computational overhead. Random Projection is immune to massive additions of points, because it does not employ data-dependent information for the embedding, such as distance metrics between processed data, which can be affected by subsequent additions.

Random Projection can be significantly accelerated in practice by employing an appropriately defined sparse projection matrix. In [Achlioptas 2001], two simple distributions are proposed that prove rather robust and can easily be applied on large datasets. Running time is then significantly reduced because R comprises either a full matrix of $+/-1$ or a sparse matrix with approximately $\frac{2}{3}$ of its cells set to 0. The *Fast Johnson-Lindenstrauss Transform* (FJLT) introduced in [Ailon and Chazelle 2010; 2006] manages to produce an even sparser matrix, thus further accelerating the procedure. FJLT reduces the number of non-zero elements in R by introducing an additional *Fast Fourier Transform* based preprocessing step. Thus, overall time requirements are then upper-bounded by $O(dnlogn + nlogd\epsilon^{-2})$, however the latter is achieved at the expense of guaranteeing

[1]In the context of this paper we follow the first variation which is the one appearing in the original publication [Faloutsos and Lin 1995].

distance distortion bounds only for projecting from ℓ_2^n to ℓ_p^k, $p = \{1, 2\}$, and not for generic Minkowski distance functions.

2.3 Landmark-based Dimensionality Reduction

In order to address the high memory requirements of MDS, the landmark-based projection methodology has been introduced. Instead of mapping all data simultaneously to the projection space, landmark-based algorithms initially extract a small fraction of points which are embedded in the projection space. Subsequently, based on the assumption that these points remain fixed (landmarks in the projection space), the rest of the dataset is projected by employing distance preservation techniques. The first method obeying this paradigm was Triangulation-based Sequential Mapping (TSM) [Lee et al. 1977].

The most prominent algorithm of this methodology is Landmark Multidimensional Scaling (Landmark MDS) [de Silva and Tenenbaum 2004; 2002], which directly addresses the scalability problem of MDS. Initially, Landmark MDS selects f points (landmark points), on which classic MDS is applied, with the constraint $f > k$. Afterwards, a distance-based triangulation procedure, which uses as input distances to already embedded landmark points, determines the projection of the remaining points. PCA can optionally be employed to align the result to the principal axes of the data set. An obvious question is related to the landmarks selection process. Although random selection produces results of acceptable quality, the authors additionally propose the MAXMIN heuristic. In MAXMIN, the first landmark is randomly picked from the set of objects while a new landmark is selected provided that it maximizes the minimum distance to any of the already selected landmarks. Following our notation, LMDS requires $O(fd)$ memory. The time requirements vary depending on the setup selected. Assuming random selection in the first step and no normalization, time complexity is $O(kfd + f^3)$, otherwise (heuristic selection of landmarks and PCA alignment) it is $O(kfd + f^3 + k^2d + k^3)$. Finally the addition of a new point necessitates the execution of only the last step of the algorithm, resulting in $O(f(n + k))$ extra load.

An approach combining the simplicity of FastMap with the paradigm of landmark-based projection is Vantage Objects [Vleugels and Veltkamp 1999]. The idea of Vantage Objects is quite simple; the embedding of a point p is identified by concatenating its distances to a set of preselected reference objects (henceforth called vantage objects). The j-th coordinate of point p_i is attributed the distance of p_i to vantage point V_j, $p'_{i,j} = d(p_i, V_j)$. The selection of Vantage Objects is accomplished either randomly or heuristically. In [Vleugels and Veltkamp 1999], the use of the MAXMIN heuristic is suggested similarly to LMDS. Trying to identify methodologies for the selection of vantage objects, the authors of [Hennig and Latecki 2003] came up with a larger set of heuristics. Although these proposals provide high quality results, they are resource-consuming and therefore their application is prohibitive in large-scale datasets.

Metric Map [Wang et al. 2005] is a recent approach similar in spirit with Vantage Objects and FastMap. The intuition behind Metric Map is to employ $2k$ objects as reference points and use them for the embeding of the whole dataset in the target space. The algorithm initially maps the small data sample of $2k$ points on the base vectors of a pseudo-Euclidean, $2k$-dimensional space. Then by employing a customised distance function, Metric Map calculates the sampled points squared distance matrix (D) and establishes the target space through the eigendecomposition of D. Finally, each remaining points is mapped in R^k by using its custom squared distances to all reference points. Similar to FastMap, Metric Map

is also agnostic towards the initial dimensionality of the dataset and requires as input only distance information and the target dimensionality. Space requirements for Metric Map are upper-bounded by $O(k^2)$. Time requirements are analogous to $O(dk^2 + k^3)$, while the cost of adding a new point is $O(k^2)$.

BoostMap [Athitsos et al. 2008] is another algorithm that uses reference points, in order to embed data in the low-dimensional space. BoostMap defines a number of embeddings following the methodology of FastMap and then treats each embedding as a classifier that predicts whether a point X is closer to reference points A or B. The combination of these weak classifiers results in the definition of a strong one with the use of Adaboost [Freund and Schapire 1995] which finally provides the embedding in the projection space. The algorithm requires $O(dT)$ time, where T is the size of the sampled training set (the various triplets) and $O(d)$ space. The addition of a new point necessitates $O(k)$ distance computations.

Sparse Map [Gabriela and Martin 1999] is a landmark-based algorithm that operates using a powerful embedding technique, namely Lipschitz embeddings [Bourgain 1985]. Lipschitz embeddings require the definition of $log_2^2 d$ data subsets organized in a matrix format with $O(log_2 d)$ rows, with row i having $O(log_2 d)$ sets of cardinality 2^i. The embedding of an object in the projection space requires the computation of $log_2^2 d$ coordinates, where the i-th coordinate identifies the minimum distance of the processed object from any of the points of the $\lfloor (i-1)/(log_2 d) + 1 \rfloor$ subset. In order to speed up the computations and reduce the dimensionality of the resulting embedding, Sparse Map introduces a number of heuristics. The reduction of high dimensional distance computations is accomplished by approximating it with the use of the already derived low-dimensional coordinates. Additionally, given a fixed value for k, Sparse Map iteratively employs the stress metric in order to identify a subset of k features from the obtained embedding that provides the lowest stress value. Time and space complexity of Sparse Map are $O(dlog_2 d)$ and $O(dlog_2^2 d)$ respectively, however this bound can be misleading in practice, since the actual requirements vary depending on the implementation of the various heuristics.

2.4 Distributed Dimensionality Reduction

The large number of distributed applications that appeared since the beginning of the decade in conjunction with the high rate of data generation have highlighted the inapplicability of centralized approaches in current research problems. Therefore it becomes obvious that a paradigm shift, towards the decentralization of data mining methods, is required in order to address these problems. This paradigm shift will also have a significant effect on the area of dimensionality reduction which comprises an important step for data preprocessing.

Distributed dimensionality reduction algorithms assume data distributed across a set of nodes and the existence of some kind of network organization scheme. The simplest case are structured peer-to-peer networks, where organization exists by construction. In such networks, a distributed hash table (DHT) determines the peer where each data object is stored. Examples include Chord [Stoica et al. 2001] and CAN [Ratnasamy et al. 2001]. In unstructured P2P networks, the organization may be induced by means of physical topology (i.e., a router) or by means of a hierarchical scheme [Doulkeridis et al. 2007]. In both cases however, a node undertakes all computations that have to be performed centrally. The most prominent approaches in the area are adaptations of PCA [Kargupta et al. 2000; Qi et al. 2004; Qu et al. 2002]. Two distributed alternatives of Fastmap [Abu-Khzam et al.

2002] have also been proposed, but their application relies heavily on the synchronization of network nodes, thus they can only be applied in controllable laboratory environments. Recently, K-Landmarks [Magdalinos et al. 2006] has been proposed as a promising solution for distributed dimensionality reduction in unstructured peer-to-peer networks.

It has to be stressed out at this point that almost all current landmark-based dimensionality reduction approaches can be applied in a distributed environment. Assuming the existence of a hierarchical organization scheme, each peer selects a set of landmark points and forwards it to an aggregator node. The latter applies the core part of the algorithm and forwards the result to all subsuming nodes. Finally, each node projects its local data independently from the rest. Landmark MDS, Vantage Objects and Metric Map can be directly employed in such context. Finally, it is worth mentioning that Random Projection is also applicable in network environments. Indeed, the Johnson-Lindenstrauss lemma and its independence of any data related metric allows a single node to generate the projection matrix and forward it to all nodes, thus significantly minimizing the required network bandwidth.

2.5 Comparative Assessment

In the context of this paragraph, we provide a comparative assessment of the aforedescribed algorithms. Since we focus on hard dimensionality reduction problems on datasets of particularly high cardinality, we exclude methods that exhibit time or space requirements analogous to or higher than $O(d^2)$ or $O(n^3)$. It is therefore natural to focus on the family of landmark-based dimensionality reduction algorithms. Table II provides an overview of the requirements induced by landmark-based algorithms. Each algorithm is presented with respect to its time and space requirements for the projection of d points from R^n to R^k. In the last column, we provide the cost of adding a new point to an existing embedding. We use T to denote the cardinality of the test dataset employed by BoostMap.

Algorithm	Time	Space	Addition
Landmark MDS	$O(kfd + f^3)$	$O(fd)$	$O(fn + fk)$
Vantage Objects	$O(dk)$	$O(nk)$	$O(k)$
SparseMap	$O(dlog_2 d)$	$O(dlog_2^2 d)$	$O(log_2^2 d)$
MetricMap	$O(dk^2 + k^3)$	$O(k^2)$	$O(k^2)$
BoostMap	$O(dT)$	$O(d)$	$O(k)$
Random Projection	$O(dkn)$	$O(kn)$	$O(kn)$

Table II. Time and space requirements of landmark-based dimensionality reduction algorithms.

Despite its high quality results, SparseMap exhibits prohibitively large space and time requirements as well as poor scaling ability. For example, in a dataset of 10^6 points, SparseMap necessitates the definition of approximately 324 subsets of objects, 18 of which would reach a cardinality of 2^{18}. Even if we speed up the process by approximating the original distance through the use of the derived coordinates, the identification of k features that exhibit low stress value is extremely difficult to be applied in practice. The reason is quite simple; following the methodology proposed in [Gabriela and Martin 1999] we should sample 10% of the dataset, or 10^5 points and define a matrix of 10^{10} elements. Obviously this induces a huge memory load while in parallel its frequent application (at least k times) requires considerable time. BoostMap exhibits similar scaling problems due to

its dependence on the size of the training set as well as the requirement of frequent execution of the classification algorithm. On the other hand Landmark MDS, Vantage Objects, Random Projection and Metric Map appear as promising solutions to our problem.

3. THE FEDRA ALGORITHM

In this section, we present FEDRA, a linear dimensionality reduction algorithm that directly addresses the two major disadvantages of classic MDS, namely its high computational complexity and high memory requirements, while exhibiting low stress values and preserving data distribution. It is designed to handle all classes of dimensionality reduction problems, however it emphasizes on hard problems. The intuition of the algorithm follows the landmark-based projection methodology [de Silva and Tenenbaum 2004; Lee et al. 1977; Wang et al. 2005]. However, compared to existing landmark-based dimensionality reduction algorithms and other embedding methods, FEDRA introduces significant advances in terms of time and space requirements. More specifically:

—FEDRA acquires the projection through an iterative set of polynomial equations, thus achieving low computational complexity and memory requirements.

—In comparison to other dimensionality reduction algorithms that are restricted to the Euclidean distance (cf. [Hjaltason and Samet 2003]), our approach is applicable for any Minkowski distance metric. Therefore, FEDRA is appropriate for applications that require the use of more complex distance functions than the Euclidean distance, or necessitate the definition of a mapping from ℓ_p^n to ℓ_p^k where $p \geq 1$.

—The proposed algorithm guarantees that an amount of the initial pairwise distances is exactly sustained, in spite of the projection.

—Finally, FEDRA establishes a bound for the error introduced due to the dimensionality reduction, thus providing theoretical guarantees for the quality of the projection.

3.1 Theory Underlying FEDRA

Before delving into the details of FEDRA, we provide a theorem that sets the methodological and practical foundations of our algorithm. Theorem 2 comprises the cornerstone of FEDRA and encapsulates in a coherent manner the methodology of the algorithm as well as the main concepts related to its application. For ease of presentation, we omit the proof of the theorem and provide it in the Appendix (Section 8). In the context of this section, we argue about the key implications of FEDRA through an illustrative example.

THEOREM 2. *A set of $k+1$ points $p_i, i = 1, ..., k+1$, described only by their pairwise distances which have been defined with the use of a Minkowski distance metric p, can be embedded in R^k without distortion through the following equations:*

$$p'_{i,j} = \begin{cases} \mid p'_{i,j} \mid^p - \mid p'_{i,j} - p'_{j+1,j} \mid^p + \sum_{f=1}^{j-1} \mid p'_{i,f} \mid^p - \sum_{f=1}^{j-1} \mid p'_{i,f} - p'_{j,f} \mid^p \\ +d_p(p_{j+1}, p_i)^p - d_p(p_i, p_1)^p = 0 & if\, j \leq i-2 \\ (d_p(p_i, p_1)^p - \sum_{f=1}^{i-2} \mid p'_{i,f} \mid^p)^{\frac{1}{p}} & if\, j = i-1 \\ 0 & otherwise \end{cases} \quad (3)$$

Additionally the embedding is determined in polynomial time.

Theorem 2 requires the exact preservation of points' pairwise distances, captured by the following set of equations: $d'(p_i, p_j) = d(p_i, p_j)$, $j = 1...k+1, i = 1...k+1$. The system of equations is obviously non-linear, since even for the Euclidean distance we need to solve a second-order equation. Although confusing at first sight, its solution is in essence quite simple and is based in practice on the axioms of Euclidean geometry. In order to understand its rationale, we provide an illustrative step-by-step example of the embedding of 4 points $\{p_1, p_2, p_3, p_4\}$ that reside in an unknown high-dimensional space into R^3 (Fig.1).

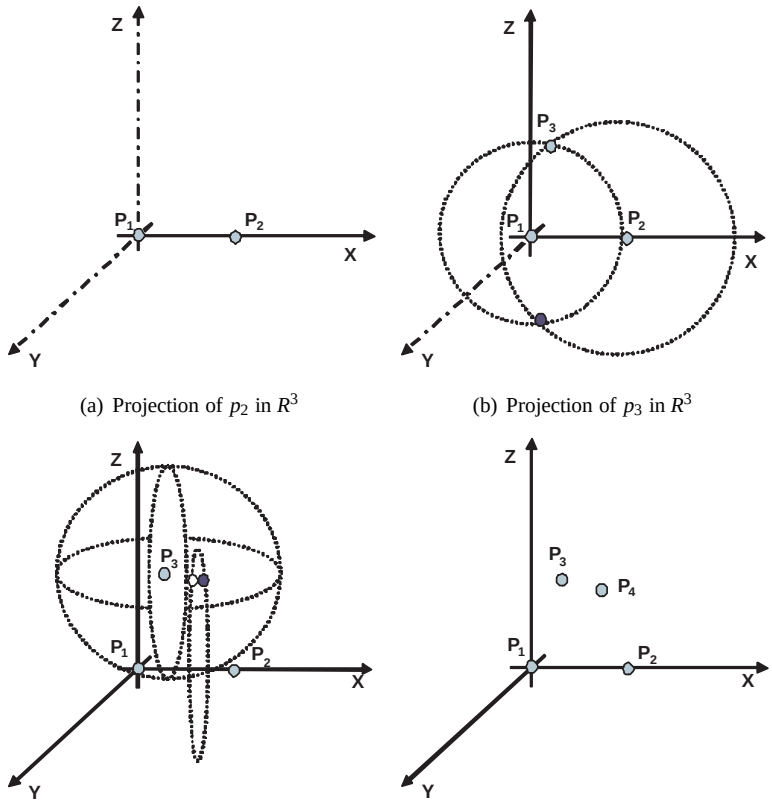
The projection of the first point p_1 is quite simple, as no constraints are imposed on its exact position yet. For reasons of simplicity, we choose to embed it in the beginning of the coordinates system, at point O. Naturally, the projection of one single point implies that no axes are required. Point p_2 must be projected on the circumference of a sphere with center O and radius $d(p_1, p_2)$, in order to preserve its distance to p_1 in the original space. For simplicity reasons, we choose to assign p_2 the coordinates $(d(p_1, p_2), 0, 0)$, as depicted in Fig. 1(a). Note that we are using only one axis, therefore the other axes are depicted with dotted lines.

The embedding of p_3 should satisfy simultaneously $d'(p_1, p_3) = d(p_1, p_3)$ and $d'(p_2, p_3) = d(p_2, p_3)$. These requirements depict two circles with centers p_1 and p_2 and radii $d(p_1, p_3)$ and $d(p_2, p_3)$ respectively. The intersection of these circles provides the embedding of p_3 in the projection space, as shown in Fig. 1(b). Two possible depictions of p_3 are identified, both symmetric with respect to the line defined by p_1 and p_2. We randomly select one to be the desired projection of p_3.

In the final step, we embed the fourth point p_4. The embedding should satisfy simultaneously $d'(p_1, p_4) = d(p_1, p_4)$, $d'(p_2, p_4) = d(p_2, p_4)$ and $d'(p_3, p_4) = d(p_3, p_4)$. These equations describe intersecting spheres in R^3. The intersection of two spheres results in the definition of a circle, which in turn intersects with the third sphere in two points (entrance and exit points), both symmetric with respect to the plane defined by points p_1, p_2, and p_3 (Fig.1(c)). We choose again one of the two possible depictions at random and obtain the mapping of p_4 in R^3, as illustrated in Fig.1(d).

Generalizing this embedding methodology, distance relations between $k+1$ points can be expressed with at most k independent variables, therefore these points can be embedded in R^k without distortion. The key remaining issue is the identification of the intersecting points of the hyperspheres. This task is not trivial, especially in the general case of any Minkowski distance metric. However, recall that we are using $i-1$ non-zero coordinates for the projection of the i-th point, therefore the derived embedding is in the form of a lower triangular matrix. Consequently, we can make the problem easier by exploiting this structure as well as calculating one coordinate at a time.

Recall that the first point p_1 is placed at O with coordinates $(0, 0, ..., 0)$. This signifies that for any point p_i the corresponding hypersphere will be of the form $\mid p'_{i,1} \mid^p + \mid p'_{i,2} \mid^p +...+ \mid p'_{i,k} \mid^p = d_p(p_1, p_i)^p$. Similarly, if we consider the second embedded point p_2 the equation would be $\mid p'_{i,1} - p'_{2,1} \mid^p + \mid p'_{i,2} \mid^p +...+ \mid p'_{i,k} \mid^p = d_p(p_2, p_i)^p$. In order to identify the intersection of these hyperspheres, we subtract the second equation from the first and get $\mid p'_{i,1} \mid^p - \mid p'_{i,1} - p'_{2,1} \mid^p = d_p(p_1, p_i)^p - d_p(p_2, p_i)^p$. The system can be easily solved with the use of Newton-Raphson method, thus deriving the coordinate $p'_{i,1}$. Consequently, for the computation of the j-th coordinate of p'_i we simply subtract the $(j+1)$-th equation from the first and identify the single root of a p-order equation of the



(a) Projection of p_2 in R^3 (b) Projection of p_3 in R^3

(c) Projection of p_4 in R^3 (d) All points in R^3

Fig. 1. Projecting 4 points $\{p_1, p_2, p_3, p_4\}$ from a high-dimensional space to R^3 using FEDRA.

form $| x |^p - | x - a |^p - d = 0$ where $d \in R$ and $p \in N\backslash\{0\}$[2]. The (i-1)-th (last) coordinate of p'_i is calculated by substituting all obtained coordinates in the first equation and solving for $p'_{i,i-1}$.

The cost of this procedure is polynomial. The requirements for the computation of the $\frac{(k+1)^2}{2}$ coordinates are O(ck^2) where c is the cost of the method employed for determining the root of the previous equation (i.e., Newton Rapshon). It has to be stressed out at this point that we have intentionally omitted further analysis related to the symmetric projections, as well as the existence of intersection between the hyperspheres. Both issues are discussed in the theoretic analysis of FEDRA in Section 4.

3.2 Landmark-based Dimensionality Reduction Algorithm

FEDRA requires as input the projection dimensionality (k), the pairwise distances between the points of the dataset (D) and the employed Minkowski distance metric (p). The only requirement is that the triangular inequality is sustained in the original space.

[2]The proof that equation $| x |^p - | x - a |^p - d = 0$ has a single root is provided in the Appendix.

Initially, k points are selected from the dataset that are going to be used as landmark points in the subsequent projection phase. This set of points defines the landmarks set L. We map the first landmark point $l_1 \in R^n$ to $O \in R^k$. All remaining landmarks (l_i,$i = 2...k$) are projected, by requiring that their distances to already projected landmarks are equal to those in the original space. Essentially, we employ the methodology of Theorem 2 and derive the set of equations (4) for the landmarks embedding procedure.

$$l'_{i,j} = \begin{cases} \mid l'_{i,j} \mid^p - \mid l'_{i,j} - l'_{j+1,j} \mid^p + \sum_{f=1}^{j-1} \mid l'_{i,f} \mid^p \\ - \sum_{f=1}^{j-1} \mid l'_{i,f} - l'_{j+1,f} \mid^p + d_p(l_{j+1}, l_i)^p - d_p(l_1, l_i)^p = 0 & \text{if j} \leq i-2 \\ (d_p(l_1, l_i)^p - \sum_{f=1}^{i-1} \mid l'_{i,f} \mid^p)^{\frac{1}{p}} & \text{if j} = i-1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

This process, apart from its inherent simplicity and fast convergence, guarantees that landmarks pairwise distances are exactly preserved in the projection space. Following the same principle, we embed all remaining points p_j ($j = 1...d-k$) in the lower dimensional space R^k, requiring that their distances to any landmark point l_i are preserved. The projection is derived by the solution of the non-linear system of equations (5).

$$d'(l_i, p_j) = d(l_i, p_j), i = 1, ..., k \tag{5}$$

Similarly to the landmarks projection process, the coordinates here can be obtained in polynomial time according to (6).

$$p'_{i,j} = \begin{cases} \mid p'_{i,j} \mid^p - \mid p'_{i,j} - l'_{j+1,j} \mid^p + \sum_{f=1}^{j-1} \mid p'_{i,f} \mid^p \\ - \sum_{f=1}^{j-1} \mid p'_{i,f} - l'_{j+1,f} \mid^p + d_p(p_i, l_{j+1})^p - d_p(l_1, p_i)^p = 0 & \text{if j} \leq i-1 \\ (d_p(l_1, p_i)^p - \sum_{f=1}^{k-1} \mid p'_{i,f} \mid^p)^{\frac{1}{p}} & \text{if j} = k \end{cases} \tag{6}$$

Based on this analysis, FEDRA is derived and its pseudocode is presented in Algorithm 1. At first, we randomly select k landmark points (lines 5-8) and embed them in the projection space with the use of equation 4 (lines 12-15). Notice that other heuristic landmark selection techniques can be integrated in the algorithm, by simply replacing function *SelectLandmark()*. We propose such techniques in Section 5. Then, we project each remaining non-landmark point with the use of equation 6 (lines 17-20). The embedded points in R^k are represented as a set P'.

At this point, it should be stressed that the order in which the landmarks are selected does not affect the projection. The only effect is a simple shift of the coordinates of all points, however the projection remains the same, since it is based on the initial pairwise distances and not on actual coordinates.

4. THEORETIC PROPERTIES

In this section, we present the theoretic properties of FEDRA. At first, we analyze its computational complexity in a comparative way (Section 4.1), against the state-of-the-art approaches presented in Section 2. Afterwards, we geometrically interpret the methodology of FEDRA (Section 4.2), and prove that for every point processed by our algorithm there always exists at least one possible embedding in the lower dimensional space (Section

Algorithm 1 FEDRA.

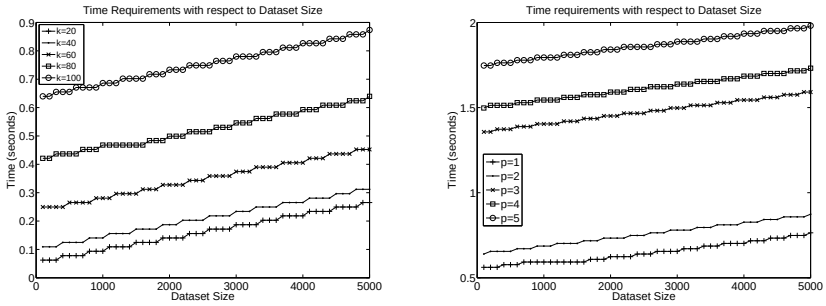
Input: Projection dimensionality (k), data distances in $R^n(D)$, distance metric (p)
Output: New dataset in R^k (P')
Initialize set of landmarks L={∅}
Initialize new dataset P'={∅}
for $i = 1$ **to** k **do**
  $l_i \leftarrow$ SelectLandmark()
  $L \leftarrow L \cup l_i$
end for
Initialize set of projected landmarks L'={∅}
Set $l'_1 = O \in R^k$
$L' \leftarrow L' \cup l'_1$
for $i = 2$ **to** k **do**
  $l'_i \leftarrow$ Calculate coordinates using Eq.(4)
  $L' \leftarrow L' \cup l'_i$
end for
P'=$P' \cup L'$
for $i = 1$ to all remaining points p_i **do**
  $p'_i \leftarrow$ Calculate coordinates using Eq.(6)
  $P' \leftarrow P' \cup p'_i$
end for

4.3). Finally, we assess the quality of the produced projection and provide a lower and upper bound of the distortion induced by applying FEDRA in the original (high-dimensional) dataset (Section 4.4).

4.1 Computational Complexity

Based on the algorithmic description, FEDRA requires O(ck) for the projection of a point in the low-dimensional space R^k. Therefore, overall time requirements for the projection of d points in R^k are exactly $O(cdk)$. We employ parameter c in order to capture the requirements posed by the solution of the equation $| x |^p - | x - a |^p - d = 0$. Consequently c is indirectly dependent on the Minkowski distance metric p as well as on the convergence rate of the employed method. In particular, when FEDRA is employed with the Euclidean metric ($p = 2$), c is equal to 1 since the solution of the equation is $x = \frac{d+a}{2a}$. In addition, our approach exhibits lower memory requirements than Landmark MDS and Random Projection. The space complexity of FEDRA is analogous to $O(k^2)$, because it requires $\frac{k^2}{2}$ in the first step (landmarks pairwise distances) and $\frac{k^2}{2} + k$ in the second (embedded landmark coordinates and distances between processed points and the landmarks in the original space).

In order to practically validate the linear dependence of FEDRA on the size of the dataset as well as the overhead of the Newton-Rapshon method we run two simple experiments. We generated a random dataset of 5000 1000-dimensional points and projected it in a lower dimensional space of dimensionality equal to 2%, 4%, 6%, 8% and 10% of initial dimensions. In the context of the first experiment, we assessed the dependence of FEDRA on the size of the processed dataset. In order to accomplish that, we fixed $p = 2$ and initiated the procedure with a dataset of 100 points and progressively augmented it by adding each



(a) Time requirements with respect to dataset size for various values of k

(b) Time requirements with respect to dataset size for various values of p

Fig. 2. Experimental assessement of FEDRA's time requirements.

time 100 more instances. Fig. 2(a) illustrates the required time where, as expected from theory, FEDRA exhibits a linear dependence on the size of the dataset. Based on this result, our approach is ideal for large datasets that call for drastic dimensionality reduction. In our second experiment, we fixed $k = 100$ (i.e., 10% of initial dimensions) and varied the value of p from 1 to 5. The obtained results are presented in Fig.2(b). Obviously, time requirements are affected by p; in particular time requirements are almost doubled when changing distance metric from $p = 2$ to $p = 3$. The latter is attributed to the convergence requirements of Newton-Raphson. However, the results are very encouraging, since the required time for projecting 5000 points for $p = 1$ and 2 is less than a second, while for $p = 5$ it marginally reaches 2 seconds.

FEDRA is indifferent to the initial dimensionality of the dataset, and this property makes it appropriate for datasets where only similarity/distance information is available. This is usually the case when objects either cannot be represented in a vector space or such a representation does not exist and only pairwise distances are available. Further, the subsequent addition of a point in an already existing projection results in O(ck) additional load, while it is as fast and efficient as in FastMap (when evaluated with the Euclidean distance). Although FJLT-based Random Projection is faster than our algorithm it provides guaranteed distortion bounds only when projecting from ℓ_2^n to ℓ_p^k with $p = \{1, 2\}$, whereas FEDRA provides corresponding bounds while projecting from ℓ_p^n to ℓ_p^k for $p \geq 1$ (Section 4.4 and Appendix). Concluding, FEDRA successfully addresses the high time and space requirements of MDS and emerges as an efficient solution in cases of hard dimensionality reduction problems on large datasets.

When compared to other methods, FEDRA exhibits the advantageous combination of fast and simple arithmetic computations. An intuitive example is derived when comparing FEDRA with SVD with respect to their time requirements as well as their implementation requirements. Notice that the first step of our procedure can be alternatively replaced by applying SVD on a set of k randomly selected landmarks. Assume that the set of landmarks defines matrix X_{kxn}. The latter can be projected in R^k through the transformation $X'_{kxk} = X_{kxn} Q^T_{kxn}$ where the columns of Q are the singular vectors of X. The relationship between the inner products matrix of the projected data (C') and the inner products matrix

of the original data (C) is given by the following computations:

$$C' = X'_{kxk}X'^{T}_{kxk} = X_{kxn}Q^{T}_{kxn}(X_{kxn}Q^{T}_{kxn})^{T} = X_{kxn}X^{T}_{kxn} = C \tag{7}$$

Moreover, each cell (i, j) of C is populated by the value $x_i x_j^T$ and based on equality $C = C'$ we conclude that $x_i x_j^T = x'_i x'^T_j$. Then the new distance between points x'_i,x'_j is:

$$d'(x_i, x_j) = \sqrt{\sum_{f=1}^{k}(x'_{if} - x'_{jf})^2} = \sqrt{\sum_{f=1}^{k}(x'^2_{if} + x'^2_{jf} - 2x'_{if}x'_{jf})} = \sqrt{x'_i x'^T_i + x'_j x'^T_j - 2x'_i x'^T_j} \tag{8}$$

But since $x_i x_j^T = x'_i x'^T_j$ the new distance can also be expressed as:

$$d'(x_i, x_j) = \sqrt{x_i x_i^T + x_j x_j^T - 2x_i x_j^T} = d(x_i, x_j) \tag{9}$$

Hence, the projection of k points from R^n to R^k with the use of SVD produces exactly the same results as FEDRA for $p = 2$. These results however were anticipated due to Theorem 2. Consequently, the key remaining issue is the time required for the projection. FEDRA manages to embed the dataset with $O(ck^2)$ computations whereas SVD requires $O(kn^2)$ or $O(k^3)$ in case we provide as input a kxk distance matrix. Arguably, if we consider the second case, for small values of k the difference might be negligible; still however FEDRA is preferable to SVD due to its implementation simplicity. Contrary to our algorithm which acquires the embedding through a set of equations, SVD -in its simplest form- requires a series of Householder transformations followed by the QR decomposition of a bidiagonal matrix([Stewart 2001]).[3]

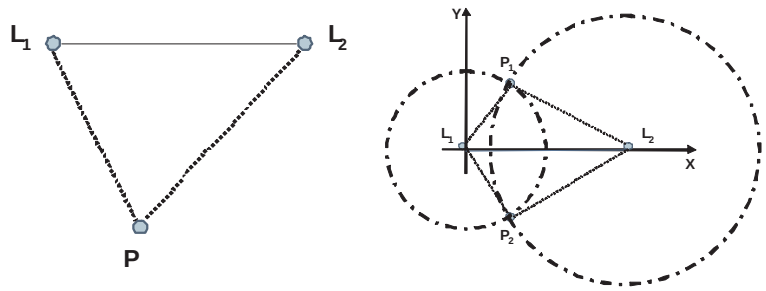
Intuitively, FEDRA owes its low memory and computational requirements to the minimization criterion employed. Instead of trying to minimize the distance discrepancies between all projected points (stress minimization criterion), FEDRA minimizes the distance deviation between the landmarks and the point under projection. One could argue that this simplification results in deteriorating projection quality. However, existing theory [de Silva and Tenenbaum 2004] and experiments (cf.Section 6) suggest that this simplification produces results of acceptable quality. Additionally, in the following paragraphs, we provide theoretical bounds regarding the distance distortion induced by FEDRA and prove that a percentage of initial pairwise distances are exactly preserved, in spite of the projection.

4.2 Geometric Interpretation

The core idea of FEDRA is the exact preservation of k distances per non-landmark point. This is achieved by requesting that distances from landmark points are exactly retained, which is captured in equations $d'_p(l_i, p) = d_p(l_i, p)$, $i = 1...k$. Each equation describes a hypersphere with center l'_i and radius $d_p(l_i, p)$. The algorithm essentially searches for the common trace of the k hyperspheres, which is the projection of point p in R^k.

An illustrative example of the embedding is provided in Fig. 3(a). In this elementary case, we project three points from an unknown high-dimensional space to R^2. Two of the points are employed as landmarks while the remaining one (P) is processed as non-landmark. Two circles are defined by applying FEDRA. The common trace of these circles

[3]If $d >> n$ it is preferable to first compute the QR decomposition of the input matrix and then according to the aforedescribed methodology calculate the SVD of R.



(a) Initial state in high-dimensional space (b) Projecting to R^2

Fig. 3. Projecting with FEDRA from a high-dimensional space to R^2.

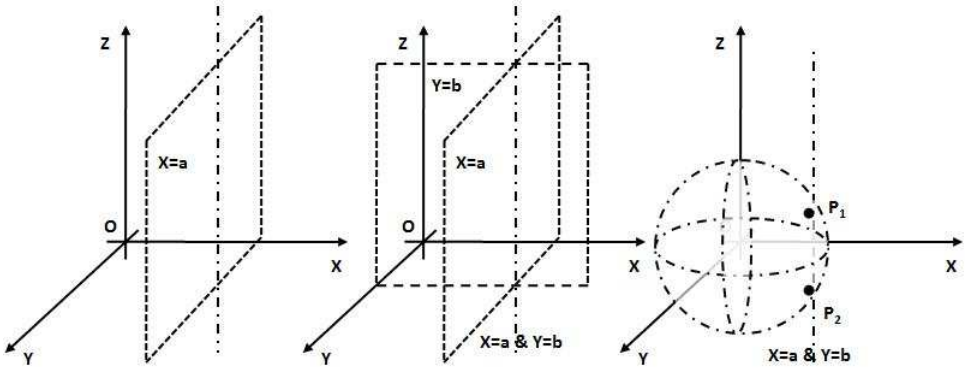


Fig. 4. Geometric interpretation of FEDRA for embedding in a 3-dimensional space.

provides the depiction of P in R^2. As shown in Fig. 3(b), two potential depictions exist, both symmetric with respect to the line defined by L_1 and L_2. Taking also into account equation set (6) this observation can be generalized; each point can be embedded in two possible places, both symmetric with respect to the hyperplane defined by the selected landmarks. In terms of arithmetics, this is because we do not compute the actual value of the k-th coordinate, but its absolute value.

From a methodological point of view this fact appears due to the methodology employed for the solution of the system of non-linear equations. Essentially, by subtracting any equation from the first we define the hyperplane on which the corresponding variable lies. For example, in Fig. 4, a step-by-step example is depicted of projecting a point P in R^3 from an unknown n-dimensional space. In the first step of the procedure, we calculate the value of coordinate X and derive value a. Consequently, any point on the plane $X = a$ can be the projection of P in R^3. By performing the same task for dimension Y we calculate value b. The intersection of these two planes defines a line, and any point belonging to this line satisfies both requirements. In the final step, we search for all points P of the line that additionally satisfy the prerequisite $d'_p(P, O) = |P'| = d_p(L_1, P)$. These points correspond to the intersection of line $X = a, Y = b$ with a sphere centered in O with radius

$d_p(L_1, P)$. There are at most two points satisfying these requirements, P_1 and P_2, one of which becomes the projection of P in R^3.

4.3 Existence of Embedding

Generalizing the last observation, the intersection of the $k-1$ hyperplanes defines a line in R^k. Consequently, the two possible values of the k-th variable depict the intersecting points of this line with a hypersphere centered in the beginning of the coordinates system, with radius the distance of the point under projection from the first landmark in the original space. A natural question that arises is whether there exists a case that the line has no intersecting point with the hypersphere. However, the next theorem guarantees that there always exists at least one intersecting point (the line is either adjacent to or intersects with the sphere), provided that the triangular inequality is sustained in the original space.

Recall that each hyperplane corresponds to the intersection of two hyperspheres (i.e., the intersection of two spheres defines a circle which in turn belongs to a two-dimensional plane). Consequently, the line captures the intersection of $k-1$ hyperspheres. Finally, if the line does not intersect with the last hypersphere it means that one of the hyperspheres has no intersection with the last one. Therefore, in the next theorem we prove that any two hyperspheres defined by FEDRA will have an intersection, provided that the triangular inequality is sustained in the original space.

THEOREM 3. *For any non-linear system of equations defined by FEDRA, there always exists at least one solution, provided that the triangular inequality is sustained in the original space.*

PROOF. By contradiction. Assume that there exists no solution for the system of equations defined by FEDRA, hence there exists no common trace between the defined hyperspheres.

Let p be a point being projected and l_i, l_j ($i \neq j$) any two landmarks. These three points define triangle $l_i p l_j$ in R^n. Without loss of generality we assume that:

$$d_p(l_i, p) \leq d_p(l_j, p) \tag{10}$$

Consequently, based on the triangular inequality we derive:

$$d_p(l_i, p) - d_p(l_j, p) \leq d_p(l_j, l_i) \leq d_p(l_i, p) + d_p(l_j, p) \tag{11}$$

In addition, Theorem 2 guarantees that distances between point p and all landmarks are exactly preserved in R^k, thus deriving:

$$d_p(l_i, p) = d'_p(l_i, p) \tag{12}$$

Landmarks pairwise distances are also exactly preserved, meaning that the following equality holds true for any pair of landmarks:

$$d_p(l_i, l_j) = d'_p(l_i, l_j) \tag{13}$$

Since there exists no common trace between the defined hyperspheres, it implies that one of the following situations has occurred; either one hypersphere is enclosed inside the

other or they are far apart and do not intersect. We examine each case separately in the following.

If one hypersphere is enclosed inside the other:

$$d'_p(l_j, l_i) < d'_p(l_i, p) - d'_p(l_j, p) \tag{14}$$

and based on equations 12 and 13:

$$d_p(l_j, l_i) < d_p(l_i, p) - d_p(l_j, p) \tag{15}$$

which contradicts with equation 11.

If the hyperspheres are far from each other and they do not intersect:

$$d'_p(l_i, p) + d'_p(l_j, p) < d'_p(l_j, l_i) \tag{16}$$

and based on equations 12 and 13:

$$d_p(l_i, p) + d_p(l_j, p) < d_p(l_j, l_i) \tag{17}$$

which again contradicts with equation 11. Thus, in both cases, the triangular inequality is violated in the original space. To conclude, the system in question always has a solution, provided that the triangular inequality is sustained in the initial, high-dimensional space. □

4.4 Quality Assessment

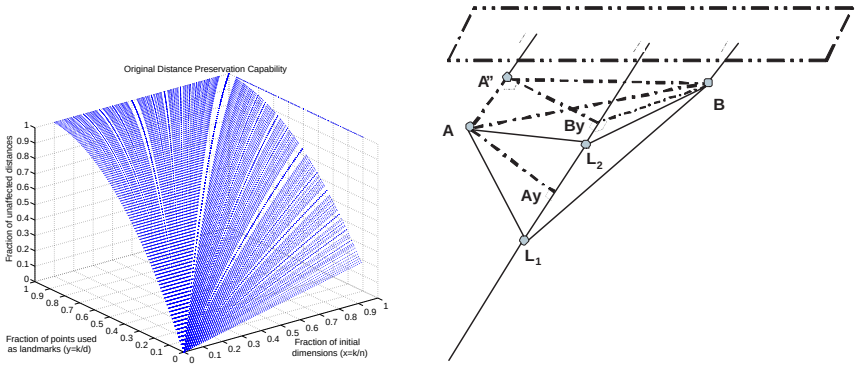
4.4.1 *Distance Preservation.* FEDRA guarantees that a certain amount of pairwise distances are exactly preserved. Indeed, the landmarks' selection and projection phases preserve exactly $\frac{k(k-1)}{2}$ pairwise distances. The subsequent embedding of the remaining $(d-k)$ data points retains another $(d-k)k$ distances. However, the latter is misleading since distance preservation is also affected by the value of n. In order to overcome this burden, we distinguish two cases, specifically $d \leq n$ and $d > n$. In the subsequent analysis we will assume that x signifies the dimensionality of the projection space as a fraction of the number of initial dimensions ($\frac{k}{n}$) while y corresponds to the dimensionality of the projection space as a fraction of the number of employed landmarks ($\frac{k}{d}$).

When $d \leq n$, according to Theorem 2 any $k \geq d-1$ retains all distances. Consequently in a space of $d-1$ dimensions we exactly preserve $\frac{d(d-1)}{2}$ distances. Additionally any value fo k satisfying $d-1 \leq k \leq n$ also guarantees exact distance preservation. Assuming that $d-1 \simeq d$ the percentage of pairwise distances that remain unaffected, due to the projection, is:

$$f(x,y) = \frac{k(k-1)+2(d-k)k}{d(d-1)} = \frac{k}{d}\frac{2d-k-1}{(d-1)} = y\frac{2d-k-1}{d-1} = y\frac{2-y-1/d}{1-1/d} \tag{18}$$

Given the fact that d is usually excessively large ($1 \ll d$) we can easily ignore $\frac{1}{d}$, since its value is close to zero, thus deriving the following:

$$f(x,y) = \begin{cases} y(2-y) & \text{if } 0 < y < 1, 0 < x < 1 \\ 1 & \text{if } y \geq 1 \end{cases} \tag{19}$$



(a) FEDRA's original distances maintenance ability.

(b) Points A,B, L_1 and L_2 in a high-dimensional space.

Fig. 5.

On the other hand, if $d > n$ we obtain exact distance preservation for $n \leq k \leq d$. Therefore, equation 19 is transformed as:

$$f(x,y) = \begin{cases} y(2-y) & \text{if } 0 < y < 1, 0 < x < 1 \\ 1 & \text{if } x \geq 1 \end{cases} \tag{20}$$

This analysis proves that a percentage of distances remains unaltered, in spite of the projection. Fig. 5(a) shows the fraction $f(x,y)$ of distances that are not modified, because of the projection. The x-axis depicts the dimensionality of the projection space calculated as a fraction x of the number of the initial dimensions. The y-axis corresponds to the percentage of points that are employed as landmarks (y). For example, if the projection dimensionality is equal to 30% of the initial dimensions ($x = 0.3$) and the number of initial dimensions is equal to the dataset cardinality ($d = n$) then the embedding acquired by FEDRA will not affect 51% of the initial distances.

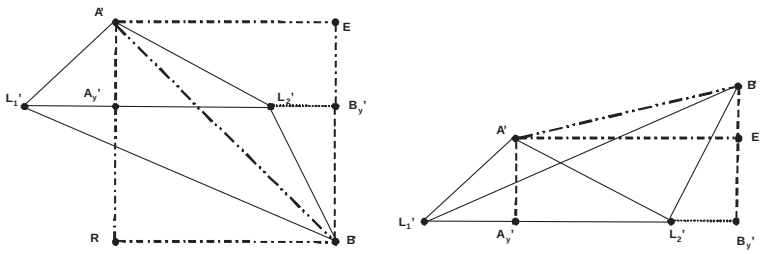
4.4.2 *Distortion.* We will now attempt to go one step further and calculate the distortion induced due to the projection to the rest of the pairwise distances. For this purpose we will use two points A and B, and study their projection with the use of two random landmarks points, L_1 and L_2 (Fig. 5(b)). For ease of presentation, we drop the formal distance notation and signify the distance between points X and Z as XZ and also assume that $p = 2$.

Each point together with the two landmarks forms a triangle. The linear segments AA_y and BB_y correspond to the altitudes of triangles L_1AL_2 and L_1BL_2 respectively. Using the cosine law on triangle L_1AL_2 we derive the length of L_1A_y.

$$L_1A_y = x = \frac{L_1A^2 + L_1L_2^2 - L_2A^2}{2L_1L_2} \tag{21}$$

Analogously, we derive from triangle L_1BL_2 the length of L_1B_y.

$$L_1B_y = y = \frac{L_1B^2 + L_1L_2^2 - L_2B^2}{2L_1L_2} \tag{22}$$



(a) Projection on different planes (b) Projection on the same plane

Fig. 6. Projection symmetric with respect to line L_1L_2

Segment $A''B$ is the orthogonal projection of AB on a plane perpendicular to the line defined by the landmarks. Obviously, since AA'' is parallel to L_1L_2, it is equal to A_yB_y. By applying the Pythagorean theorem on triangle $AA''B$ we derive:

$$AB^2 = A''B^2 + (y - x)^2 \tag{23}$$

Additionally with the use of the triangular inequality on $A''B_yB$ we derive:
$A''B \leq A''B_y + B_yB \Rightarrow$
$A''B \leq AA_y + BB_y \Rightarrow$
$A''B^2 \leq AA_y^2 + B_yB^2 + 2AA_yBB_y$

Consequently, by employing equality 23, we provide an upper bound of the initial distance AB:

$$AB^2 \leq AA_y^2 + BB_y^2 + 2AA_yBB_y + (x - y)^2 \tag{24}$$

where $AA_y = \sqrt{AL_1^2 - x^2}$ and $BB_y = \sqrt{BL_1^2 - y^2}$ are obtained by the application of the Pythagorean theorem on AL_1A_y and BL_1B_y respectively.

While projecting, FEDRA will replicate the two triangles. Consequently, we have two ways of projecting that are depicted in Fig.6. The projections of the original points A,B are depicted as A' and B' respectively, while L_1' and L_2' are the projections of the landmark points in the new, two-dimensional space. Due to the fact that FEDRA guarantees the exact replication of the two triangles, we know that $A'A_y'$=AA_y, $B'B_y'$=BB_y, $B'L_1'$=BL_1 and $A'L_1'$=AL_1. Consequently, the new distance $A'B'$ can be calculated by applying the Pythagorean theorem on triangle $A'EB'$. In the first case (Fig.6(a)), we calculate $A'B'^2$=$(y-x)^2 + (AA_y - BB_y)^2$, while in the second (Fig.6(b)), $A'B'^2$=$(y - x)^2 + (AA_y + BB_y)^2$. By combining the last relation with the bound of AB obtained in the high-dimensional space (Eq. 24), we have that $AB \leq A'B'$. Notice also that the squares of the two new distance values differ only by $4AA_yBB_y$. Consequently, a unique lower bound can be obtained through relation:

$$AB^2 - 4AA_yBB_y \leq A'B'^2 \tag{25}$$

The upper bound can be simply derived by the triangular inequality on triangle $A''B_yB$ (Fig.5(b)). Since $A''B_y = AA_y$ and $A''AB$ is orthogonal we have that:

$|AA_y - BB_y| \leq A''B \Rightarrow$
$(AA_y - BB_y)^2 \leq A''B^2 \Rightarrow$
$(AA_y - BB_y)^2 + (y-x)^2 \leq A''B^2 + (y-x)^2 \Rightarrow$
$(AA_y - BB_y)^2 + (y-x)^2 \leq AB^2 \Rightarrow$
$(AA_y - BB_y)^2 + (y-x)^2 + 4AA_yBB_y \leq AB^2 + 4AA_yBB_y \Rightarrow$
$(AA_y + BB_y)^2 + (y-x)^2 \leq AB^2 + 4AA_yBB_y$

and since $(AA_y - BB_y)^2 + (y-x)^2 \leq (AA_y + BB_y)^2 + (y-x)^2$ we derive a unique upper bound:

$$A'B'^2 \leq AB^2 + 4AA_yBB_y \tag{26}$$

Consequently, by combining equations 25 and 26 we derive observation 1 which provides a bound for the new distance $A'B'$ in the case of the Euclidean distance metric.

OBSERVATION 1. *Using any two landmarks L_1, L_2, FEDRA can project any two points A, B in a given low-dimensional space while guaranteeing that their new distance $A'B'$ will be bounded according to:*

$$AB\sqrt{1 - \frac{4AA_yBB_y}{AB^2}} \leq A'B' \leq AB\sqrt{1 + \frac{4AA_yBB_y}{AB^2}} \tag{27}$$

where AA_y, BB_y are the lengths of the altitudes of triangles L_1AL_2, L_1BL_2 respectively.

Another way of approximating the error induced by the projection is with the use of the cosine law. As discussed previously, FEDRA essentially replicates the manifold defined by $k+1$ high-dimensional points in the new lower dimensional space R^k. In our previous example (Fig.5(b)), this signifies that angles $\widehat{AL_1L_2}$ and $\widehat{BL_1L_2}$ (i.e ψ, ω respectively) will remain unaltered (Fig.6). Consequently the new distance can be calculated as $A'B'^2 = AL_1^2 + BL_1^2 - 2BL_1AL_1\cos(\psi + \diagup - \omega)$ while the original distance was $AB^2 = AL_1^2 + BL_1^2 - 2BL_1AL_1\cos(\phi)$, where $\phi = \widehat{AL_1B}$. Adding and subtracting value $2BL_1AL_1\cos(\phi)$ from the first relation we obtain $A'B'^2 = AB^2 - 2BL_1AL_1(\cos(\psi + \diagup - \omega) - \cos(\phi))$. Since $-1 \leq cos(x) \leq 1$ we can bound $A'B'$ as:

$$AB\sqrt{1 - \frac{4BL_1AL_1}{AB^2}} \leq A'B' \leq AB\sqrt{1 + \frac{4BL_1AL_1}{AB^2}} \tag{28}$$

Relation 28 is extremely important for the derivation of the following observation.

OBSERVATION 2. *Given a point X in a high-dimensional space R^n, all points that are at most r from X in R^n are projected in R^k in a circle with center the embedding of X in R^k and radius $r + 2d(L_i, X)$, where L_i is the landmark point closest to X.*

PROOF. From the previous analysis we have that the new distance between point X and unknown point Y is at most $d'(X,Y)^2 \leq d(X,Y)^2 + 4d(X,L_i)d(Y,L_i)$, where L_i is any landmark point. Unfortunately we are not aware of the exact value of $d(Y,L_i)$ but we can substitute it with the larger value that can satisfy our prerequisites, that is $d(X,L_i)+r$. Consequently all points satisfying our prerequisite lay in a circle with center the embedding of X in R^k and radius $r + 2d(X,L_i)$. In order to minimize the diameter of the circle, we choose to assess the distance with the use of the closest landmark. □

Based on equation 27, we derive the distortion product c_1c_2 for FEDRA which is equivalent to $\sqrt{\frac{AB^2+4AA_yBB_y}{AB^2-4AA_yBB_y}}$. Attempting a head-to-head comparison with Random Projection, the corresponding ϵ in the case of FEDRA is exactly $\frac{4AA_yBB_y}{AB^2}$ which is different for every pair of points. Unfortunately, FEDRA is unable to provide a global bound ϵ for all points since it is a *data-aware* method. Contrary to Random Projection that define an explicit *data-oblivious* distortion bound ϵ using only the target dimensionality (k) and the cardinality of the dataset (d), FEDRA exploits information from the underlying dataset (i.e., landmarks distances) in order to produce the embedding. Therefore the error due to the projection is directly related to the selected landmarks. The latter is not strange to algorithms of its genre; for example FastMap's low distortion bound for projecting in one dimension is $AB^2-(x-y)^2 = A'B'^2 \leq AB^2$ with x,y being defined by the employed pivot points, similar to FEDRA.

The aforedescribed analysis can be generalized for the case of any Minkowski distance metric by taking into account the general expression for the Pythagorean theorem and the cosine law. The generalization is provided in the Appendix (Section 8, observations 3,4 and lemma 6).

5. FEDRA EXTENSIONS

In this section we present three extensions to the basic algorithm. The first is a heuristic – directly derived from the theoretic properties of FEDRA – that enables the selection of landmarks that improve the quality of the embedding (Section 5.1). The second is a complementary heuristic, which is employed to further improve the quality of the produced results (Section 5.2). Finally, the application of FEDRA in a widely distributed environment, such as a large-scale peer-to-peer networks, is also presented (Section 5.3).

5.1 Landmark Selection Heuristic

So far, we have implicitly assumed that landmarks are randomly selected. Although this approach produces results of acceptable quality, we introduce a heuristic which is able to intentionally select a set of landmarks that minimizes the distortion induced by the embedding.

Based on the analysis of Section 4.4, we have identified the bound for the distance distortion between any two points A,B as $\sqrt{AB^2-4AAyBBY} \leq A'B' \leq \sqrt{AB^2+4AAyBBY}$. The minimization of the induced distortion implies the minimization of the product $4AAyBBY$, which in turn is achieved by the simultaneous minimization of AAy and BBy. However, recall that $AAy = \sqrt{AL_1^2-x^2}$ and $BBy = \sqrt{BL_1^2-y^2}$, thus both values are minimized when $AL_1 - x \to 0$ and $BL_1 - y \to 0$. Considering the case of A and substituting x with $\frac{L_1A^2+L_1L_2^2-L_2A^2}{2L_1L_2}$ we obtain:

$$AL_1 - \frac{L_1A^2+L_1L_2^2-L_2A^2}{2L_1L_2} = \frac{(L_1A-L_1L_2-L_2A)(L_1A-L_1L_2+L_2A)}{2L_1L_2} \to 0 \qquad (29)$$

Consequently, the minimization is achieved when $L_2A \simeq L_1A - L_1L_2$ or $L_2A \simeq L_1L_2 - L_1A$. The first condition occus when landmarks are selected in such a way that they exhibit minimum distance from each other. The latter is intuitively verified by considering a random triangle L_1AL_2. If L_1L_2 is small compared to L_1A then L_2A will be approximately

equal to L_1A, thus $L_2A \simeq L_1A - L_1L_2$ holds true. Our landmark selection algorithm works in the following way. We select the first landmark at random and then we iteratively select a new one by requiring that it minimizes the overall distance from all previously selected landmarks. Assuming we have selected $f-1$ landmarks, the f-th will be point p that satifies $argmin \sum_{j=1}^{f-1} d_p(p, l_j)$.

This procedure however is costly for datasets of high cardinality (millions of records), since all data points need to be processed before a new landmark is selected. Therefore, we propose a more efficient strategy based on sampling. We draw uniformly S data samples of cardinality C and apply the proposed heuristic. In the end, the set with the minimum distance sum is retained as the global landmark set. The procedure yields a memory cost of $O(Cn)$, $C \in N$ and time requirements analogous to $O(Sk)$.

It has to be stressed out at this point that this procedure is inherently heuristic, therefore it cannot always guarantee that the selected set of landmarks will be the optimal one. A potential case of failure may appear if one or more landmark points are outliers. The latter is due to the fact that any outlier landmark (L_o) when combined with another landmark L_i will not validate expression $L_oA \simeq L_iA - L_iL_o$ since $L_iA < L_iL_o$. A simple remedy to this deficiency is the re-application of the landmark selection process each time with a different starting point.

We have intentionally ignored the second case since it implies a more laborious and computational expensive approach. Condition $L_2A \simeq L_1L_2 - L_1A$ requires that landmarks are choosen so as to exhibit minimum distances from a set of points and maximum distances from another set. Considering a dataset that enjoys a cluster structure with clusters well separated and far from each other, this condition is valid when we employ the cluster centroids as landmarks. Assuming that L_1 is the center of the cluster in which A is situated and L_2 is the center of another cluster then L_1A is small while L_2L_1 is approximately of the same length as L_2A. Unfortunately, in order to guarantee fast computation of the landmarks we need to be aware in advance of this structure or at least be supplied with specific data statistics. However, in general, we will be obliged to execute a clustering algorithm which will result in significant load to the system. Consequently we decide to ignore this approach.

5.2 Projection Heuristic

The independent projection of each point with respect to the other non-landmark points is one of the factors for the reduced complexity of FEDRA. However, this simplification may sometimes come at a cost, as it cannot always guarantee that pairwise distances between non-landmark points are also well-approximated. The latter is due to the fact that the new distance is calculated by the result of a linear combination of the initial distance as well as the distances between the selected landmarks. This potential case of failure is depicted in the analysis of Section 4.4, where it is obvious that two closely situated points in the original space may end up far apart in the projection space if the lengths of the altitudes of the corresponding triangles are large. However, this would only be a significant problem, if it would occur for every pair of landmarks; consequently, this situation rarely appears in practice. Nevertheless, we provide a fast heuristic which detects such a problematic situation and defines the best possible embedding for each point.

The proposed evaluation algorithm (Algorithm 2) takes as input the original distances (D), the set of already projected non landmark points (NLP), the set of projected landmarks (PL), the Minkowski distance metric (p) and the point under projection x and tries to find

Algorithm 2 Projection Heuristic

1: **Input:** Original Distances (D), Minkowski metric (p), Point under projection (x), Projected non landmark points (NLP), Projected landmark points (PL)
2: **Output:** x'
3: $NL \leftarrow$ randomly select k points from NLP
4: Calculate x_1 and x_2 the two possible embeddings of x
5: Set $C_1 \leftarrow 0, C_2 \leftarrow 0$
6: **for** $i = 1$ **to** k **do**
7: Select nl_i from NL
8: Calculate d_1 and d_2 from $d'_p(nl_i, x_1)$ and $d'_p(nl_i, x_2)$
9: **if** $|d_1 - d_p(nl_i, x)| \leq |d_2 - d_p(nl_i, x)|$ **then**
10: $C_1 \leftarrow C_1 + 1$
11: **else**
12: $C_2 \leftarrow C_2 + 1$
13: **end if**
14: **end for**
15: **if** $C_1 \leq C_2$ **then**
16: $x' \leftarrow x_2$
17: **else**
18: $x' \leftarrow x_1$
19: **end if**

the embedding that minimizes the distance distortion between point x and k randomly selected, already projected, non-landmark points. The added value of this heuristic lays in the fact that it guarantees minimum distortion for additional $dk - k^2$ pairwise distances, thus further ameliorating FEDRA's quality. Moreover, overall time and space requirements are analogous to $O(dk)$ and $O(k^2)$ respectively[4], thus not posing any significant overhead to the basic algorithm.

5.3 Distributed Dimensionality Reduction with Landmark Points

The fact that FEDRA operates with only a fraction of the overall dataset and achieves results of high quality promotes it as an attractive candidate for application in a distributed context. As already stated in Section 2, the area of distributed knowledge discovery poses a number of new challenges that primarily originate from the fact that no network element can gather all available data. Unfortunately, existing work in the area of distributed dimensionality reduction fails to provide a robust solution. Algorithms based on eigen analysis deteriorate and need to recompute the decomposition, in the case that many new points are added. The two adaptations of FastMap require a high amount of exchanged messages, thus they work well only when node availability and intercommunication are guaranteed, otherwise the synchronization of network nodes is practically impossible.

It is therefore obvious that a distributed dimensionality reduction algorithm should combine the salient features of the aforementioned techniques, in terms of network load, algorithmic complexity and quality of results, while being immune to subsequent changes in

[4]Contrary to the pseudocode of Algorithm 2, the actual input is only x since all other information is provided through pointers to the permanent storage medium. During the execution of the algorithm we only have to occupy $2k + k^2$ space on main memory –x_1, x_2 and k k-dimensional points.

the processed data (i.e., massive addition or deletion of points). Moreover, the algorithm should be adaptable to potential network failures, as well as topology changes. Finally it has to apply to the full extent of distributed applications, starting from controllable laboratory environment and reaching large-scale peer-to-peer networks.

Inspired by our previous work [Magdalinos et al. 2006], we introduce the distributed application of FEDRA. Obviously, the distributed extension of FEDRA bares similarities with K-Landmarks with respect to the decentralization methodology, however differentiates significantly with respect to the implementation of each step as well as the exhibited time requirements. Moreover, FEDRA is applicable with any Minkowski distance metric, contrary to K-Landmarks that is confined to the Euclidean distance. The corresponding extension is presented in Algorithm 3. The only assumption made is the existence of a hierarchical network overlay, where an aggregator node exists [Doulkeridis et al. 2007]. The aggregator uses k landmark points (L) sampled from the whole network (lines 9-11) and projects them to R^k using equation 4. The original set of landmark points and the generated mapping (L') are forwarded to all nodes (line 19), which in turn project local points independently (lines 25-28).

The proposed algorithm differs significantly from other widely employed distributed dimensionality reduction approaches, since it achieves the projection of the vast majority of points independently from the rest, implying that only the projection of few landmarks is done in a centralized manner. Moreover, it is not affected by any changes in the network topology or any subsequent data unavailability since all points are projected with respect to the landmark points. Consequently, no re-computation of the projection is required, in order to guarantee the preservation of projection quality. Finally, the network load imposed is lower than the load of other algorithms. The network cost of the application of FEDRA in a distributed environment is $O(nkM)$ where M is the number of peers in the network, while distributed PCA [Qi et al. 2004] necessitates $O(Mn^2 + nkM)$. However, there exists one disadvantage; the agnostic nature of centralized FEDRA towards the initial dimensionality of the dataset is lost, since points pairwise distances cannot be known in advance.

6. EXPERIMENTS

In this section we present the experimental evaluation of FEDRA, which verifies the expected performance. Thus, FEDRA emerges as an attractive solution for hard dimensionality reduction problems on large-scale datasets. The aim of the experimental assessment process is threefold:

(1) To validate the effectiveness and efficiency of FEDRA on hard dimensionality reduction problems and highlight its scalability.

(2) To demonstrate the enhancement of a typical data mining task, such as clustering, due to the application of FEDRA.

(3) To experimentally show the merits of FEDRA in a distributed setup, where restrictions are usually imposed on the amount of data that can be exchanged.

The obtained results prove the suitability and viability of our algorithm for problems where data is described by hundreds of coordinates and applying a clustering algorithm is highly demanding in terms of time and space requirements.

Algorithm 3 Distributed FEDRA

Input: Projection dimensionality (k), node id (i), number of landmark points of node i (k_i), local dataset defined in R^n (P)
Output: local dataset defined in R^k (P')
Initialize new dataset P'={∅}
Initialize set of landmarks L={∅}
Initialize set of projected landmarks L'={∅}
Initialize local set of landmarks L_i={∅}
$L_i \leftarrow$ RandomLandmarkSelection()
if node is aggregator **then**
  for $j = 1$ **to** all nodes **do**
    Receive k_j landmarks (L_j) from node j
    $L \leftarrow L \cup L_j$
  end for
  Set $l'_1 = O \in R^k$
  Set $L' \leftarrow L' \cup l'_1$
  for $j = 2$ **to** k **do**
    $l'_j \leftarrow$ Calculate coordinates using Eq. (4)
    $L' \leftarrow L' \cup l'_j$
  end for
  Communicate L, L' to all nodes
else
  Send L_i to aggregator
  Receive L, L'
end if
$P' \leftarrow P' \cup L'$
for $j = 1$ to all remaining points p_j **do**
  $p'_j \leftarrow$ Calculate coordinates using Eq. (6)
  P'=$P' \cup p'_j$
end for

6.1 Experimental Methodology

In the context of FEDRA's validation, we run a series of dimensionality reduction experiments.

Algorithms and Datasets. We compare the performance of FEDRA against FastMap, Metric Map, Landmark MDS, Random Projection, Vantage Objects and PCA. We employed eight real world and artificial datasets in our experiments. Four of them were acquired from the UCI Machine Learning Repository[5]. More specifically Ionosphere, Segmentation, Musk and Synthetic Control were used. Ionosphere contains radar observations of earth's ionosphere while Segmentation and Musk contain high-level numeric-valued attributes corresponding to images and molecules respectively. Finally, the Synthetic Control is a set of synthetically generated control charts. Another four, of particularly high cardinality and dimensionality, were acquired from the Pascal Large Scale Challenge[6] that took

[5]http://archive.ics.uci.edu/ml/
[6]http://largescale.first.fraunhofer.de/about/

place in ICML 2008. To the best of our knowledge these datasets are the largest that have ever been employed for the experimental assessment of a dimensionality reduction algorithm. The datasets together with their properties are summarized in Table III. All datasets were embedded in a space of dimensionality equal to 2%, 4%, 6%, 8% and 10% of their initial dimensions. In the case of the three smaller UCI datasets, where these values are unattainable, we set the lower dimensionality to 3, 4, 5, 6 and 7 respectively.

Dataset	Cardinality	Dimensionality	Classes	Description
Ionosphere	351	34	2	Radar Observations
Segmentation	2100	19	7	Image Segmentation Data
Musk	476	166	2	Molecules Data
Synthetic Control	600	60	6	Synthetic dataset
alpha	500000	500	2	Pascal Large Scale '08
beta	500000	500	2	Pascal Large Scale '08
gamma	500000	500	2	Pascal Large Scale '08
delta	500000	500	2	Pascal Large Scale '08

Table III. Datasets used in the experiments.

Due to the fact that the application of PCA is infeasible on the Pascal datasets, we employ the covariance aggregation scheme of Global PCA (GPCA) [Qi et al. 2004]. The latter is based on the simple observation that given globally centered data the eigenvector u of matrix $(m-1)cov(X)+(p-1)cov(Y)$ is also an eigenvector of $(m+p-1)cov([X^T Y^T]^T)$ where cov denotes the covariance matrix, X, Y are the data samples and m, p the respective cardinalities. Consequently, our implementation of PCA necessitates three passes over the whole dataset, one for calculating the global mean, a second for the calculation of the covariance matrix and a final one for the projection of the dataset.

In order to calculate the various heuristics on the Pascal datasets we iteratively drew 10 uniform random samples equal to 1% of the original dataset (5000 instances). For each sample, we apply the heuristic and retain the sample that best satisfies the corresponding conditions. In the case of the MAXMIN heuristic of LMDS, where we seek to maximize the landmarks minimum distances, we choose to retain the sample that maximizes the sum of distances. On the contrary, for FEDRA's landmark selection heuristic, we maintain the one that minimizes the overall sum. Obviously, this strategy may not yield results equal to those when the heuristic is applied on the whole dataset, yet it still provides results of acceptable quality. For the projection heuristic of our algorithm we randomly sample k already projected points and use them for the projection. Finally, for Random Projection we employ the two distributions of [Achlioptas 2001] while we follow the simple implementation methodology described in [Ailon and Chazelle 2010].

All algorithms (with the exception of PCA) were executed 10 times. Consequently, all reported values correspond to the obtained mean. Fastmap was deployed only with the 4 small datasets and evaluated against FEDRA with respect to the exhibited time and stress values. Although FastMap cannot scale for datasets of significant cardinality, we employed it as an evaluation benchmark of our algorithm's effectiveness and efficiency. The statistical significance of all experiments has been verified with a t-test with confidence level set to 0.99.

Metrics. In order to support the efficiency claims made earlier in the paper, we report the execution time of FEDRA and compare it against the corresponding requirements of

the other algorithms. Following the concept of the application-oriented evaluation metrics (see Section 2.1), we certify the effectiveness of FEDRA by comparing its original distance maintenance capability with the one exhibited by other landmark-based algorithms. The comparison is accomplished through the computation of *stress*. Due to the fact that the computation of stress requires large amounts of time and space, we employed the four UCI datasets for this purpose.

Furthermore, in order to demonstrate FEDRA's capability to enhance the quality of a clustering algorithm, we evaluated the original and projected datasets with the use of *k*-Means. Each result of *k*-Means is evaluated according to the *Purity* (*P*) metric. Purity considers the mapping of a cluster C_i ($i = 1...a$) to a class S_j ($i = 1...a$) based on the highest observed overlap. The quality of this assignment is measured by counting the number of correctly classified instances and dividing by the total number of instances (N). Purity is formally defined as $\frac{1}{N}\sum_{i,j=1}^{a} max(|C_i \cap S_j|)$.

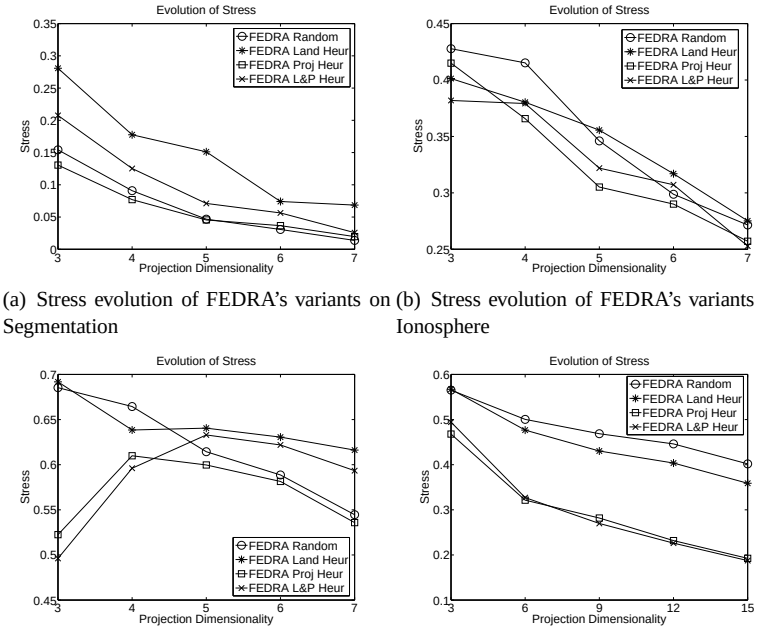
For each algorithm, we present its clustering quality maintenance capability, which is defined as $\frac{P_n}{P_o}$ where P_n is the purity score obtained in the projection space, while P_o corresponds to the value obtained from the original dataset. All clustering experiments were repeated 10 times and we report here the obtained mean values. Again, their statistical significance has been verified with a t-test with confidence level set to 0.99. Finally, we measure the time requirements of *k*-Means in the projection space and compare it against those in the original space. The algorithm is obviously accelerated because of the reduced dimensionality, however it still exhibits different time requirements for each of the embedded datasets. This is due to the fact that each algorithm produces a different embedding, which affects the convergence rate of *k*-Means. In all experiments, we employ the Euclidean distance metric.

Distributed Setup. Using the distributed variation of FEDRA, we also consider enhancing the quality of distributed clustering. In order to execute these experiments, we assume that a large dataset is distributed among the nodes of a peer-to-peer network and the task is to derive the global clustering model, without imposing significant network load. All algorithms assume the existence of a star overlay network, where each peer communicates its sample (or result) to an aggregator node that undertakes the task of performing any subsequent computations.

We use the Pascal datasets and employ as reference the Purity values obtained by the distributed *k*-Means algorithm of [Datta et al. 2006] (DKMeans). The assessment methodology of DKMeans follows the same principles as the one of *k*-Means. Additionally, in all cases, we measure the communication cost imposed by the execution of the various dimensionality reduction algorithms. All experiments took place in a simulated peer-to-peer environment of 500 nodes where topology was randomly generated with nodes being connected with 5% probability. Again reported results correspond to the mean value of 10 executions.

Experimental Setup and Source Code. All algorithms have been implemented in MATLAB R2009a. For the experiments we used four Intel Core 2 Quad processors at 2.4GHz with 4GB of RAM running Ubuntu Linux v.9.04. *k*-Means and DKMeans have been implemented in Java[7].

[7]The full source code accompanied by deployment instructions can be found at http://www.db-net.aueb.gr/panagis/TKDD2009/



(a) Stress evolution of FEDRA's variants on Segmentation

(b) Stress evolution of FEDRA's variants on Ionosphere

(c) Stress evolution of FEDRA's variants on Synthetic Control

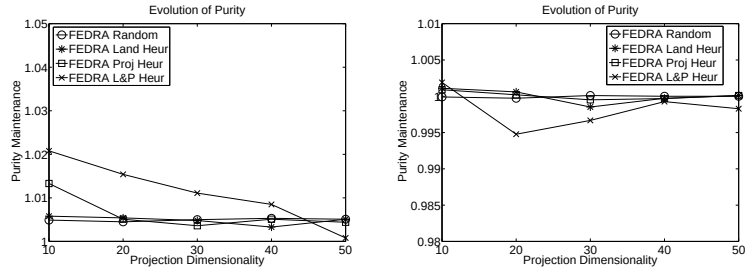
(d) Stress evolution of FEDRA's variants on Musk

Fig. 7. Assessment of FEDRA's variants with respect to the obtained stress values.

6.2 Sensitivity Analysis of FEDRA

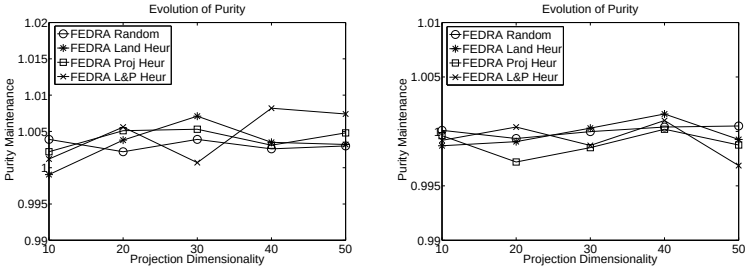
We first evaluate the effect of FEDRA's heuristics (one for the landmark selection process and another one for the projection) on the basic algorithm. Their combination yields four different variants of FEDRA. The first is the basic FEDRA algorithm which employs random landmark selection and random projection. The second variant employs intentional landmark selection process (denoted as *FEDRA Land Heur*) and projects all data randomly, while the third variant uses assisted projection and random landmarks (denoted as *FEDRA Proj Heur*). Finally, FEDRA can be deployed by employing both the landmark selection and projection heuristics (denoted as *FEDRA L& P Heur*). The purpose of our sensitivity analysis is to provide an empirical study regarding the efficiency of these heuristics as well as the cost induced by their execution.

Regarding the quality of the produced embedding in terms of the exhibited stress, we notice that all configurations exhibit approximately the same behavior. In the case of the Segmentation dataset (Fig.7(a)), the random and assisted projection setups provide the best results, while intentional landmark selection exhibits slightly larger values due to inappropriate selection of landmarks. Indeed, the intentional landmark selection process, due to its heuristic nature, cannot guarantee that the best set of landmarks will be selected and therefore that the subsequent projection phase will be significantly enchanced. In general, provided that we pick a set of closely positioned landmarks, the intentional landmark selection is expected to exhibit better behaviour than the random FEDRA configuration, a fact that appears in Figure 7(d).



(a) Purity evolution of FEDRA's variants on alpha

(b) Purity evolution of FEDRA's variants on beta



(c) Purity evolution of FEDRA's variants on gamma

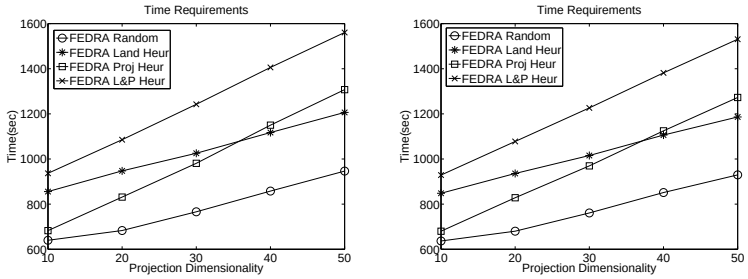
(d) Purity evolution of FEDRA's variants on delta

Fig. 8. Assessment of FEDRA's configurations with respect to relative clustering quality maintenance

However, as k grows – and therefore the number of selected landmarks – the results of all variants tend to decrease and converge to approximately the same value. Same results have been obtained by the Ionosphere dataset (Fig.7(b)), where the assisted projection produced slightly better results. The merit of both heuristics however is demonstrated in the first two iterations with the Synthetic control dataset (Fig.7(c)), where the embeddings produced by the assisted projection heuristic were of significantly better quality than those of the other configurations. On the other hand, in the last two iterations we notice that the variants employing the intentional landmark selection exhibit a slight deterioration in the stress (0.05-0.07), again due to inappropriate selection of landmarks. Finally, the musk dataset comprises an excellent example of the power of the projection heuristic as well as the power of the landmarks random selection (Fig.7(d)). The latter highlights that the projection heuristic comprises the key enabling component for stress minimization.

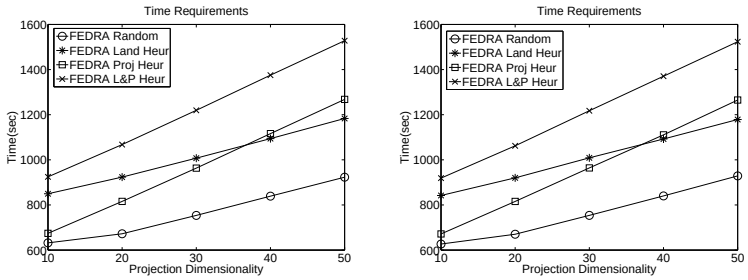
Similar results were obtained in the evaluation of the clustering quality of the produced embedding. All configurations of the algorithm exhibit approximately the same behavior. In the case of the alpha dataset (Fig.8(a)), the assisted projection heuristics produced results of higher quality than the rest, however the amelioration cannot be considered significant. In the remaining datasets (Figs.8(b), 8(c), 8(d)), the results were approximately the same.

As expected, the time requirements of the four variants differ significantly (Figs.9(a), 9(b), 9(c), 9(d)). The basic algorithm necessitates the least amount of time, while the use of both heuristics requires significantly more time. The most noticeable fact however is the behavior of the assisted projection setup compared to the intentional landmark selection. According to theory, we would expect both configurations to behave approximately



(a) Time requirements of FEDRA's variants for alpha

(b) Time requirements of FEDRA's variants for beta



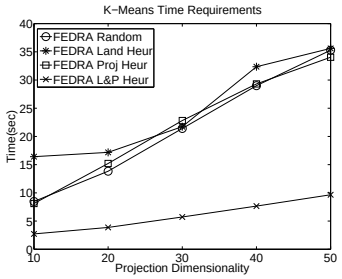
(c) Time requirements of FEDRA's variants for gamma

(d) Time requirements of FEDRA's variants for delta

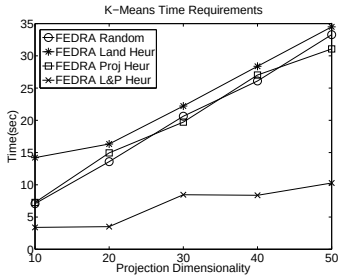
Fig. 9. Assessment of FEDRA's configurations with respect to the exhibited time requirements. Time is measured in seconds

the same. Although this cannot be justified theoretically, it is due to the experimentation strategy we employed. Recall that the landmark selection process requires drawing a number of samples, a fact which is directly translated to disk-accesses. The latter is absent in the case of assisted projection, where we use the already projected points which reside in memory.

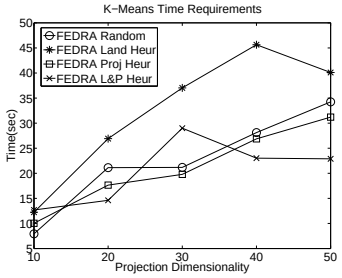
Finally, we would like to validate the quality of the produced embedding in terms of the convergence speed of *k*-Means. Obviously, the obtained cluster structures are of equal quality, however the convergence of *k*-Means is influenced by the separation of the clusters. Well-separated clusters enable the algorithm to produce results faster, while a more fuzzy layout necessitates additional loops. This experiment highlights the power of the combination of both heuristics, where in two cases (Figs.10(a) and 10(b)) they enable the *k*-Means algorithm to converge significantly faster that the other variants. It should be stressed out that in both experiments, when dimensionality reaches 50, *k*-Means converges almost 4 times faster on the dataset produced by the assisted projection. Intuitively, this fact is explained using the underlying theory as presented in Section 5. The landmark heuristic guarantees that the selected landmark set provides a good approximation of the points pairwise distances. However, the addition of the projection heuristic directs FEDRA to place points originally situated near in the high-dimensional space nearer in the projection space. On the other hand, distant points are projected far from each other. Essentially, the combination of the landmarks selection and points projection heuristics enables FEDRA to produce an embedding that best discriminates clusters, thus enabling the faster



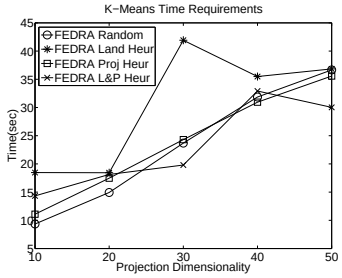
(a) Time requirements of *k*-Means for alpha



(b) Time requirements of *k*-Means for beta



(c) Time requirements of *k*-Means for gamma



(d) Time requirements of *k*-Means for delta

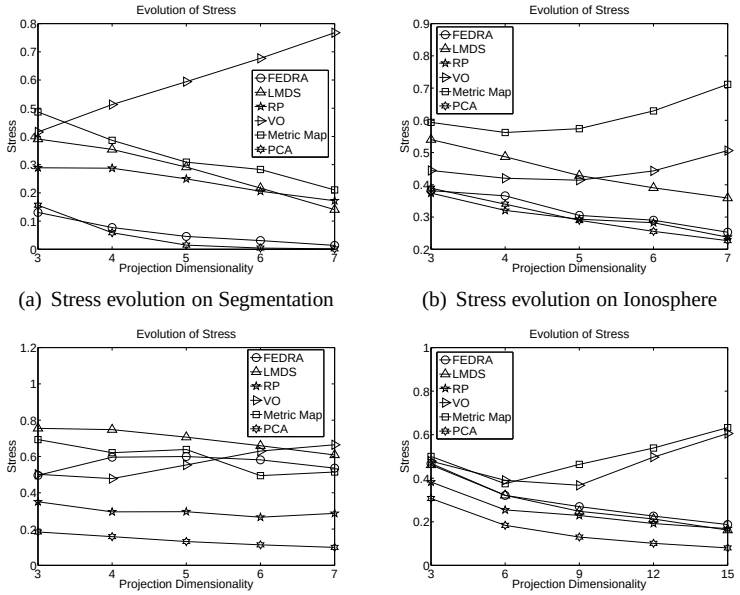
Fig. 10. Assessment of FEDRA's configurations with respect to the exhibited time requirements of *k*-Means. Time is measured in seconds

convergence of *k*-Means. The latter is also depicted in the case of the two other datasets, although in less extent (Figs.10(c) and 10(d)).

We conclude that the best configuration for FEDRA is the one combining the landmarks selection and points projection heuristics. However, it is worth pointing out the fact that the basic FEDRA appears as a good compromising solution, especially with respect to the imposed time requirements. Indeed, the basic algorithm manages to maintain clustering quality, while exhibiting low stress values and the lowest possible time requirements. On the other hand, the obtained results signify that landmarks do not ameliorate significantly the resulting embedding. Although the intentional landmark selection process enhances the results, the required cost may not always justify its application.

6.3 Comparative Study

After having analyzed FEDRA and its variants, we proceed to compare its performance with that of well-known linear dimensionality reduction algorithms. We compare FEDRA against FastMap, Metric Map, Landmark MDS (LMDS), Random Projection (RP), Vantage Objects (VO) and PCA. The algorithms were chosen specifically for their low time and space requirements, while PCA is employed as benchmark, due to its high quality results. Landmark MDS was deployed both with random landmark selection as well as MAXMIN, while in both cases the selected landmarks were two times more than the projection dimensionality ($f = 2k$). Additionally, we used both distributions for Random Projection and the four different configurations of FEDRA. However, for ease of presentation, we report here only the best results obtained from each algorithm.



(a) Stress evolution on Segmentation (b) Stress evolution on Ionosphere

(c) Stress evolution on Synthetic Control (d) Stress evolution on Musk

Fig. 11. Comparative assessment of all methods with respect to stress

(a) Stress values obtained while evaluating FEDRA and FastMap on the Segmentation and Ionosphere datasets.

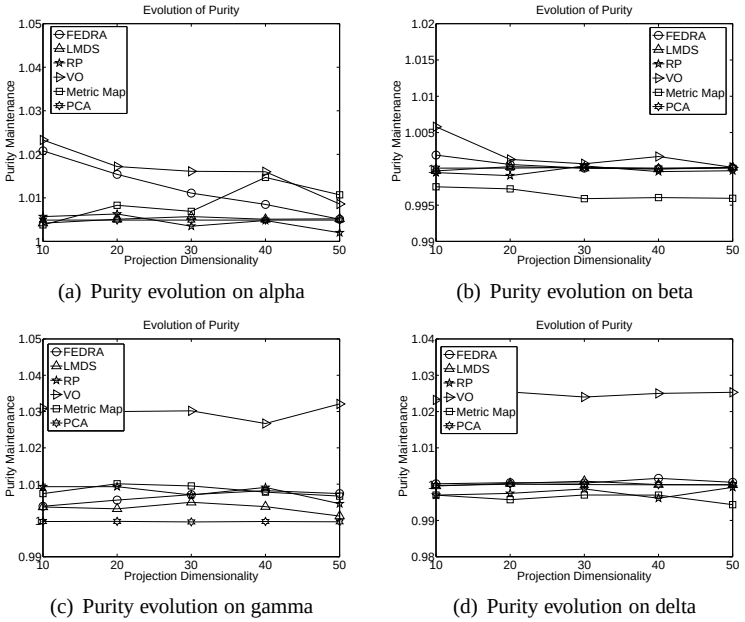
	Segmentation					Ionosphere				
	k=3	k=4	k=5	k=6	k=7	k=3	k=4	k=5	k=6	k=7
FastMap	0.14	0.12	0.10	0.08	0.05	0.47	0.44	0.36	0.33	0.30
FEDRA	0.15	0.09	0.05	0.03	0.01	0.43	0.42	0.35	0.30	0.27
FEDRA Land Heur	0.28	0.18	0.15	0.07	0.07	0.40	0.38	0.36	0.32	0.28
FEDRA Proj Heur	0.13	0.08	0.05	0.04	0.02	0.41	0.37	0.31	0.29	0.26
FEDRA L& P Heur	0.21	0.13	0.07	0.06	0.03	0.38	0.38	0.32	0.31	0.25

(b) Stress values obtained while evaluating FEDRA and FastMap on the Synthetic control and Musk datasets.

	Synthetic					Musk				
	k=3	k=4	k=5	k=6	k=7	k=3	k=6	k=9	k=12	k=15
FastMap	0.29	0.28	0.22	0.19	0.20	0.43	0.27	0.19	0.16	0.15
FEDRA	0.69	0.66	0.61	0.59	0.54	0.57	0.50	0.47	0.45	0.40
FEDRA Land Heur	0.69	0.64	0.64	0.63	0.62	0.57	0.48	0.43	0.40	0.36
FEDRA Proj Heur	0.52	0.61	0.60	0.58	0.54	0.47	0.32	0.28	0.23	0.19
FEDRA L& P Heur	0.50	0.60	0.63	0.62	0.59	0.49	0.33	0.27	0.23	0.19

Table IV. Stress values for FEDRA and FastMap on the various small-size datasets

FEDRA's ability in maintaining distance information while projection becomes evident, when compared against other algorithms. FEDRA managed to provide embeddings of high quality, while sometimes approximating the quality of PCA as depicted in Figs.11(a) and 11(b). In both experiments, FEDRA was slightly outperformed only by PCA. However, FEDRA should not be considered as an alternative to PCA but rather as a fast approxima-



(a) Purity evolution on alpha (b) Purity evolution on beta

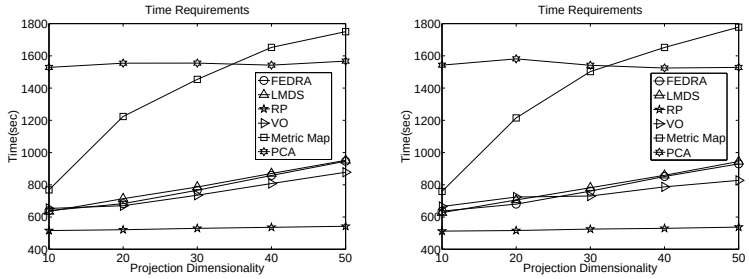
(c) Purity evolution on gamma (d) Purity evolution on delta

Fig. 12. Comparative assessment of all methods with respect to relative clustering quality maintenance

tion algorithm which manages to produce results of acceptable quality compared to those obtained by similar algorithms of its genre. Indeed, studying Figs.11(c) and 11(d), we notice that this time FEDRA is significantly outperformed by PCA, however it produces better results than the other landmark-based algorithms. Both observations have been validated also when performing an experimental comparative assessment with FastMap. In 2 out of 4 cases, FEDRA behaved extremely well, managing to produce results comparable or even better than FastMap (Table IV(a)), while in the remaining datasets FEDRA was outperformed (Table IV(b)). The latter was expected since FastMap essentially comprises a heuristic approximation of PCA[8], therefore it exhibits approximately the same behavior.

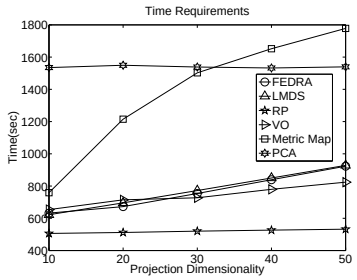
The next experiments however highlights the merits of data preprocessing with dimensionality reduction. Recall that our methodology directs the projection in a space with dimensionality ranging from 2% to 10% of the initial dimensions. Applying this on a 500 coordinates space results projecting in a space ranging from 10 to 50 dimensions. Obviously, the indirect gains of this procedure are immense, considering the fact that the output dataset is going to be used as input for another data mining or knowledge discovery task. The most important outcome however is the clustering maintenance results (Figs. 12(a), 12(b), 12(c), 12(d)), where almost all algorithms managed to maintain the original clustering quality, and even slightly ameliorate it by 2%-3%. The best results were obtained by Vantage Objects, which managed to produce an embedding that exhibited better amelioration of clustering quality than the other approaches in 3 out of 4 datasets. FEDRA behaved approximately equal to the other algorithms with minor deviations which are not

[8]The selection of the most distant objects for the projection by FastMap essentially approximates the selection of the maximum variance axis of PCA

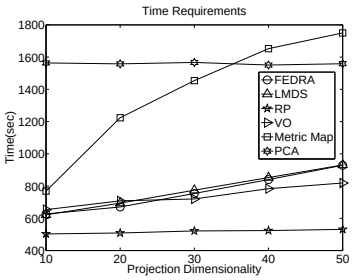


(a) Time requirements for alpha

(b) Time requirements for beta



(c) Time requirements for gamma



(d) Time requirements for delta

Fig. 13. Comparative assessment of all methods with respect to the exhibited time requirements. Time is measured in seconds

(a) Time requirements (in seconds) while evaluating FEDRA and FastMap on the Segmentation and Ionosphere datasets.

	Segmentation					Ionosphere				
	k=3	k=4	k=5	k=6	k=7	k=3	k=4	k=5	k=6	k=7
FastMap	0.27	0.32	0.35	0.35	0.39	0.07	0.09	0.10	0.12	0.13
FEDRA	0.24	0.29	0.34	0.37	0.37	0.08	0.07	0.08	0.07	0.08
FEDRA Land Heur	0.27	0.31	0.34	0.37	0.41	0.09	0.10	0.13	0.11	0.12
FEDRA Proj Heur	0.37	0.45	0.52	0.57	0.62	0.08	0.08	0.10	0.12	0.14
FEDRA L& P Heur	0.40	0.49	0.54	0.65	0.69	0.11	0.14	0.15	0.14	0.15

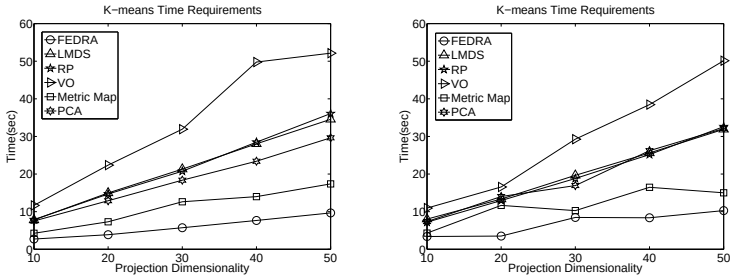
(b) Time requirements (in seconds) while evaluating FEDRA and FastMap on the Synthetic control and Musk datasets.

	Synthetic					Musk				
	k=3	k=4	k=5	k=6	k=7	k=3	k=6	k=9	k=12	k=15
FastMap	0.14	0.16	0.17	0.19	0.23	0.15	0.15	0.19	0.20	0.24
FEDRA	0.11	0.12	0.13	0.15	0.17	0.13	0.14	0.16	0.19	0.22
FEDRA Land Heur	0.15	0.15	0.18	0.18	0.19	0.21	0.21	0.24	0.27	0.31
FEDRA Proj Heur	0.14	0.15	0.19	0.18	0.21	0.15	0.18	0.23	0.27	0.31
FEDRA L& P Heur	0.18	0.20	0.25	0.25	0.27	0.21	0.24	0.29	0.32	0.37

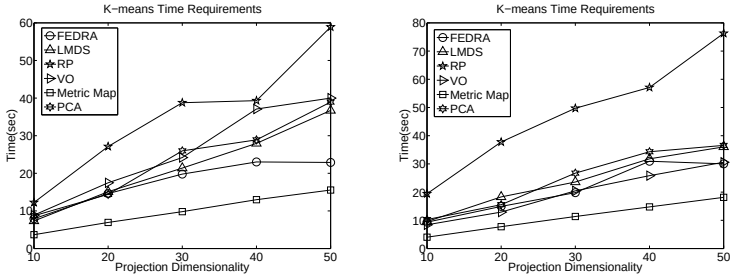
Table V. Time requirements (in seconds) for FEDRA and FastMap on the various small-size datasets

considered significant.

Time requirements follow the theoretic analysis as presented in Section 2.5. The base FEDRA configuration exhibits similar behavior to FastMap and sometimes manages to



(a) Time requirements of *k*-Means for alpha (b) Time requirements of *k*-Means for beta

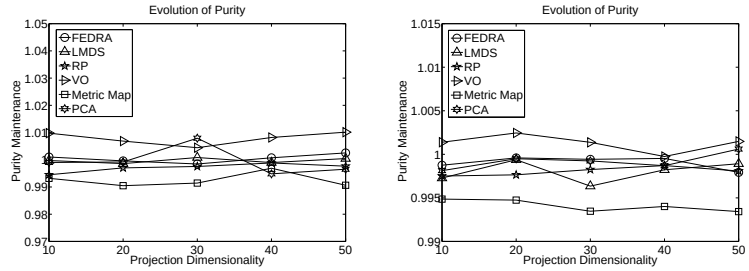


(c) Time requirements of *k*-Means for gamma (d) Time requirements of *k*-Means for delta

Fig. 14. Comparative assessment of all methods with respect to the exhibited time requirements of *k*-Means. Time is measured in seconds

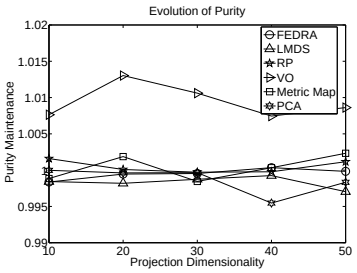
produce results slightly faster. The overall results of this comparison appear in Tables V(a) and V(b). As far as the large datasets are concerned, Random Projection is faster in all experiments, which is expected since the only requirement is the definition of a rather simple projection matrix. On the other hand, PCA is generally the most expensive algorithm in most cases. Metric Map is influenced by the augmentation in the value of *k*. Finally, FEDRA, Vantage Objects and Landmark MDS require approximately the same time. The overall results are provided in Figs. 13(a), 13(b), 13(c), 13(d). All time measurements also encapsulate the time requirements imposed for accessing the hard disk, since the datasets do not reside in main memory.

In the final experiment, we compare the convergence requirements of *k*-Means on the embeddings produced by all algorithms. In this case, two out of four experiments highlight FEDRA's ability to enable faster convergence of *k*-Means, while Vantage Objects require the most time (Figs. 14(a) and 14(b)). In parallel, Metric Map also produces embeddings that support *k*-Means. Additionally, it is worth noticing that the projections obtained by Random Projection were those that required significant additional time in the gamma and delta datasets (Figs. 14(c) and 14(d)). In general, Metric Map and FEDRA exhibited the most stable results and produced consistent results in all datasets. We note at this point that the execution of *k*-Means on the original datasets required 296 seconds for alpha, 324 seconds for beta, 400 seconds for gamma and 383 seconds for delta. Obtaining results of equal quality in less than 10 seconds, obviously comprises a huge acceleration of the algorithm. Of course, based on the reported time requirements, one could argue that the application of *k*-Means on the original dataset is still faster than that of dimensionality reduction. How-

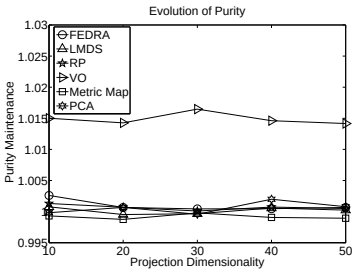


(a) Purity evolution on alpha

(b) Purity evolution on beta



(c) Purity evolution on gamma



(d) Purity evolution on delta

Fig. 15. Comparative assessment with respect to relative distributed clustering quality maintenance

ever, these results were obtained by different programming languages (MATLAB and Java respectively) and therefore cannot be considered directly comparable.

Based on our experimental study, we conclude that FEDRA is a viable solution for hard dimensionality reduction problems. In our experimental validation process, FEDRA managed to combine low stress values with low time requirements and produced embeddings that enabled the fast and accurate convergence of *k*-Means. These features were hardly combined in the majority of the competitive solutions. Despite their high quality results, Vantage Objects and Metric Map exhibit high stress values. Metric Map additionally necessitates significantly more time that Vantage Objects and FEDRA. Additionally, Vantage Objects produced embeddings that slowed down *k*-Means in 3 out of 4 cases. The same results were obtained for Random Projection, thus leading us to the conclusion that FEDRA is an attractive solution in terms of the quality factors we have analyzed so far.

6.4 Distributed Dimensionality Reduction

In the final set of experiments, we validate the applicability of FEDRA in a distributed context, where each node has a fragment of the whole dataset. We compare the distributed version of FEDRA with the corresponding adaptations of all other algorithms. The major assumption made in these simulations is the existence of a star overlay network where the central node undertakes the tasks that need to be carried out centrally. For the Random Projection and Vantage Objects, the aggregator defines the projection matrix or the reference objects and forwards them to all peers. In the case of PCA, all nodes initially compute their local means and forward them to the aggregator. The latter computes the global mean and disseminates it network-wide. Afterwards, all peers calculate their local covariance matrixes and forward them to the aggregator, which finally extact the *k* principal eigen-

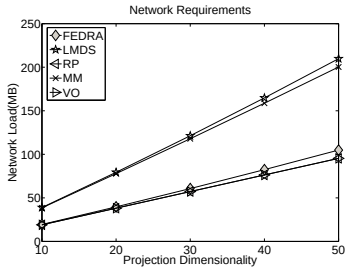


Fig. 16. Network Requirements of all methods for distributed dimensionality reduction

vectors and eigenvalues and sends them back. This methodology is in accordance with the directives of GPCA. For the rest of the algorithms, peers select locally their landmarks and forward them to the aggregator which, in turn, calculates either their projection or the projection matrix and replies with the result. Then, each peer projects each of its local points independently from the rest.

The assessment methodology follows the same principle as before. At first, we validate the clustering quality of the produced embedding and then present the induced network load by the application of dimensionality reduction. We excluded the landmark selection heuristics of FEDRA and LMDS, because on one hand their application would produce additional network load, while on the other hand in our previous experiments gave similar results with the random selection process.

The overall obtained results are provided in Figs. 15(a), 15(b), 15(c) and 15(d). The common characteristic in all graphs is the clustering quality maintenance in every projection. Additionally, Vantage Objects behave slightly better than the other approaches exhibiting an amelioration ranging from 0.5% to 1.5%. FEDRA behaves similarly to the other algorithms and manages to clearly differentiate from Random Projection and Landmark MDS in the alpha and beta datasets. The network load imposed by this operation is presented in Fig. 16. We have intentionally omitted PCA, since the aggregation of the covariance matrix results in an overall cost of approximately 1GB. Although the latter is tolerable, since it amounts less that 50% of the size of the total dataset, it is significantly larger than the load of the other algorithms. Based on this analysis, we conclude that FEDRA can also be applied in a distributed context, producing high quality results comparable to PCA, while also managing to keep network consumption low.

7. CONCLUSIONS AND FURTHER WORK

In the context of this paper we proposed FEDRA, a novel, linear dimensionality reduction algorithm with low time and space requirements. FEDRA embeds data in the new space by following the landmark-based projection methodology, where a limited set of points is used to assist the reduction process. We thoroughly analyzed FEDRA's theoretic properties and based on this analysis we proposed two extensions complementary to the base algorithm. Moreover we introduced the distributed adaptation of FEDRA thus making it an attractive candidate for distributed data preprocessing.

Through extensive experimental validation we highlighted the merits of FEDRA as well as its applicability in hard dimensionality reduction problems. FEDRA produced results comparable to the best algorithms in all assessment experiments thus combined all salient

characteristics that an ideal dimensionality reduction method should have, namely low time and space requirements, minimum distortion values as well as clustering structure and quality preservation. With respect to the latter we observed the acceleration of *k*-Means when it was applied on the projected dataset obtained from our algorithm. In future work we will exploit FEDRA in the context of similarity search and nearest neighbor retrieval in large and distributed databases.

REFERENCES

Abu-Khzam, F., Samatova, N., Ostrouchov, G., Langston, M., and Geist, A. 2002. Distributed dimension reduction algorithms for widely dispersed data. In *International Conference on Parallel and Distributed Computing Systems (PDCS)*. 167–174.

Achlioptas, D. 2001. Database-friendly random projections. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*. 274–281.

Ailon, N. and Chazelle, B. 2006. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*. 557–563.

Ailon, N. and Chazelle, B. 2010. Faster dimension reduction. *Commun. CACM 53*, 2, 97–104.

Athitsos, V., Alon, J., Sclaroff, S., and Kollios, G. 2008. BoostMap: An embedding method for efficient nearest neighbor retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 1, 89–104.

Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. 1999. When is "nearest neighbor" meaningful? In *International Conference on Database Theory (ICDT)*. 217–235.

Bourgain, J. 1985. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics 52*, 1 (March), 46–52.

Carreira-Perpinan, M. A. 1997. A review of dimension reduction techniques. *Technical Report, CS-96-09, University of Shefield*.

Chakrabarti, S. 2002. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kauffman.

Dasgupta, S. and Gupta, A. 2003. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structured Algorithms 22*, 1, 60–65.

Datta, S., Giannella, C., and Kargupta, H. 2006. K-Means clustering over a large, dynamic network. In *SIAM International Conference on Data Mining (SDM)*.

de Silva, V. and Tenenbaum, J. B. 2002. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems (NIPS)*. 705–712.

de Silva, V. and Tenenbaum, J. B. 2004. Sparse multidimensional scaling using landmark points. *Technical Report*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 6, 391–407.

Doulkeridis, C., Nørvåg, K., and Vazirgiannis, M. 2007. DESENT: Decentralized and distributed semantic overlay generation in P2P networks. *IEEE Journal on Selected Areas in Communications 25*, 1 (Jan.), 25–34.

Drineas, P., Kannan, R., Mahoney, M. W., and A, L. 2006. Fast monte carlo algorithms for matrices iii: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing 36*, 1, 184–206.

Faloutsos, C. and Lin, K.-I. 1995. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *International Conference on Management of Data (SIGMOD)*. 163–174.

Freund, Y. and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory (EuroCOLT)*. 23–37.

Gabriela, H. and Martin, F. 1999. Cluster-preserving embedding of proteins. Tech. rep., Center for Discrete Mathematics and Theoretical Computer Science.

Hennig, C. and Latecki, L. J. 2003. The choice of vantage objects for image retrieval. *Pattern Recognition 36*, 9, 2187 – 2196.

Hjaltason, G. R. and Samet, H. 2003. Properties of embedding methods for similarity searching in metric spaces. *IEEE Trans. Pattern Anal. Mach. Intell. 25*, 5, 530–549.

KARGUPTA, H., HUANG, W., SIVAKUMAR, K., PARK, B.-H., AND WANG, S. 2000. Collective principal component analysis from distributed heterogeneous data. In *Proceedings of Principles of Data Mining and Knowledge Discovery (PKDD)*. 452–457.

LEE, R., SLAGLE, J., AND BLUM, H. 1977. A triangulation method for the sequential mapping of points from n-space to two-space. *IEEE Transactions on Computers C-26*, 3 (March), 288–292.

LI, Z., LIN, D., AND TANG, X. 2009. Nonparametric discriminant analysis for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 31*, 4, 755–761.

LIAN, X. AND CHEN, L. 2009. General cost models for evaluating dimensionality reduction in high-dimensional spaces. *IEEE Trans. Knowl. Data Eng. 21*, 10, 1447–1460.

MAGDALINOS, P., DOULKERIDIS, C., AND VAZIRGIANNIS, M. 2006. K-Landmarks: Distributed dimensionality reduction for clustering quality maintenance. In *Proceedings of Principles of Data Mining and Knowledge Discovery (PKDD)*. 322–334.

MAGDALINOS, P., DOULKERIDIS, C., AND VAZIRGIANNIS, M. 2009. FEDRA: A fast and efficient dimensionality reduction algorithm. In *SIAM International Conference on Data Mining (SDM)*. 509–520.

MAHONEY, M. W. AND DRINEAS, P. 2009. CUR matrix decompositions for improved data analysis. *Proceedings of National Academy of Sciences (PNAS) 106*, 697–702.

PAYNE, T. R. AND EDWARDS, P. 1999. Dimensionality reduction through Correspondence Analysis. Tech. rep., AUCS-TR-9910, Carnegie Mellon University.

QI, H., WANG, T., AND BIRDWELL, D. 2004. Global principal component analysis for dimensionality reduction in distributed data mining. *Chapter 19 in Statistical Data Mining and Knowledge Discovery, CRC Press*, 327–342.

QU, Y., OSTROUCHOV, G., SAMATOVA, N., AND GEIST, A. 2002. Principal component analysis for dimension reduction in massive distributed data sets. In *5th International Workshop on High Performance Data Mining*.

RATNASAMY, S., FRANCIS, P., HANDLEY, M., KARP, R., AND SCHENKER, S. 2001. A scalable content-addressable network. *ACM SIGCOMM*, 161–172.

SHARMA, A. AND PALIWAL, K. K. 2008. Rotational linear discriminant analysis technique for dimensionality reduction. *IEEE Trans. Knowl. Data Eng. 20*, 10, 1336–1347.

STEWART, G. W. 2001. *Matrix algorithms Vol I, II*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

STOICA, I., MORRIS, R., KARGER, D., KAASHOEK, F. M., AND HARI. 2001. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM*.

SWETS, D. AND WENG, J. 1996. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. 18*, 831–836.

TOGERSON, W. 1958. *Theory and methods of scaling*. Wiley.

VLEUGELS, J. AND VELTKAMP, R. 1999. Efficient image retrieval through vantage objects. *Pattern Recognition 35*, 69–80.

WANG, J., WANG, X., SHASHA, D., AND ZHANG, K. 2005. MetricMap: an embedding technique for processing distance-based queries in metric spaces. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35*, 5 (Oct.), 973–987.

YANG, Q. AND WU, X. 2006. 10 Challenging Problems in Data Mining Research. *International Journal of Information Technology & Decision Making 5*, 4, 597–604.

YE, J., YE, J., H.XIONG, LI, Q., XIONG, H., PARK, H., JANARDAN, R., AND KUMAR, V. 2004. IDR/QR: An incremental dimension reduction algorithm via QR decomposition. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. 364–373.

8. APPENDIX.

In the context of the appendix we provide some further analysis of key issues which indirectly support the definition and analysis of FEDRA. At first we provide the proof of Theorem 4 which comprises a fundamental step for the successfull application of our algorithm. Furthermore we provide the proof of Theorem 2. In the end, we provide the extension of the Pythagorean theorem and the Cosine Law for any Minkowski distance metric and present the generalized distortion study. Although most observations presented here can

be considered elementary, they have been included in order to ensure completeness (i.e., support the provision of a standalone document).

THEOREM 4. *Any equation of the form* $f(x) = \mid x \mid^n - \mid x - a \mid^n - d$ *where* $a \in R\backslash\{0\}$, $d \in R$ *and* $n \in N\backslash\{0\}$ *has a single root in R.*

PROOF. $f(x)$ is differentiatable and continuous in R since it is polynomial. Consequently in order to have a single root it must exhibit a sign change in R while also being monotonous. The successful validation of these two requirements would signify that $f(x)$ intersects with axis X at a single point, thus has a single root. In order to define the derivative $f'(x)$ of $f(x)$ we distinguish two cases, according to the value of a.

—$a > 0 \Rightarrow$
  —$f'(x) = n \mid x \mid^{n-1} - n \mid x - a \mid^{n-1} > 0$ when $x > 0$, $x > a$
  —$f'(x) = n \mid x \mid^{n-1} + n \mid x - a \mid^{n-1} > 0$ when $x \geq 0$, $x < a$
  —$f'(x) = -n \mid x \mid^{n-1} + n \mid x - a \mid^{n-1} > 0$ when $x \leq 0$, $x < a$

—$a < 0 \Rightarrow$
  —$f'(x) = n \mid x \mid^{n-1} - n \mid x - a \mid^{n-1} < 0$ when $x \geq 0$
  —$f'(x) = -n \mid x \mid^{n-1} - n \mid x - a \mid^{n-1} < 0$ when $x \leq 0$, $x > a$
  —$f'(x) = -n \mid x \mid^{n-1} + n \mid x - a \mid^{n-1} < 0$ when $x < 0$, $x < a$

Consequently $f(x)$ is monotonous in R and specifically is ascending when $a > 0$ and descending when $a < 0$. Next we must prove that there exists a single root. In order to accomplish that we will use the Bolzano theorem. We define $c = \frac{d}{|a|^n}$ and distinguish four cases, according to the values of d and c:

—$d > 0, 0 < c < 1 \Rightarrow$
  —$f(a) = |a|^n - c|a|^n = (1 - c)|a|^n > 0$
  —$f(0) = -|a|^n - c|a|^n = (-1 - c)|a|^n < 0$

—$d > 0, c > 1 \Rightarrow$
  —$f(a) = |a|^n - c|a|^n = (1 - c)|a|^n < 0$
  —$f(ca) = |ca|^n - |ca - a|^n - c|a|^n > 0$ since $(|c|^n - |c - 1|^n - c)|a|^n > 0$ which necessitates $|c|^n - |c - 1|^n - c > 0$ or equivalently $1 - \mid 1 - \frac{1}{c} \mid^n - \frac{c}{|c|^n} > 0$. Assuming that $\frac{c}{|c|^n} \simeq 0$ then our relation holds true since $\mid 1 - \frac{1}{c} \mid^n < 1$

—$d < 0, c < -1 \Rightarrow$
  —$f(a) = |a|^n - c|a|^n = (1 - c)|a|^n > 0$
  —$f(0) = -|a|^n - c|a|^n = (-1 - c)|a|^n < 0$

—$d < 0, -1 < c < 0 \Rightarrow$
  —$f(a) = |a|^n - c|a|^n = (1 - c)|a|^n < 0$
  —$f(-ca) = \mid -ca|^n - \mid -ca - a|^n - c|a|^n > 0$ since $(|c|^n - |c + 1|^n - c)|a|^n > 0$ which necessitates $|c|^n - |c + 1|^n - c > 0$ or equivalently $1 - \mid 1 + \frac{1}{c} \mid^n - \frac{c}{|c|^n} > 0$. Assuming that $\frac{c}{|c|^n} \simeq 0$ then our relation holds true since $\mid 1 + \frac{1}{c} \mid^n < 1$

Based on the latter and with the use of the Bolzano theorem we conclude that $f(x)$ has a root in $(-\infty, +\infty)$which is single since our function is monotonous in R. In particular

—if $d > 0$ and $0 < \frac{d}{|a|^n} < 1$ then the root lays in (0,a)

—if $d > 0$ and $\frac{d}{|a|^n} > 1$ then the root lays in (a,ca)

—if $d < 0$ and $\frac{d}{|a|^n} > -1$ then the root lays in (0,a)

—if $d < 0$ and $-1 < \frac{d}{|a|^n} < 0$ then the root lays in $(a,-ca)$

□

THEOREM 5. *A set of $k+1$ points $p_i, i = 1,...,k+1$, described only by their pairwise distances which have been defined with the use of a Minkowski distance metric p, can be embedded in R^k without distortion through the following equations:*

$$p'_{i,j} = \begin{cases} |\ p'_{i,j}\ |^p - |\ p'_{i,j} - p'_{j+1,j}\ |^p + \sum_{f=1}^{j-1} |\ p'_{i,f}\ |^p - \sum_{f=1}^{j-1} |\ p'_{i,f} - p'_{j,f}\ |^p & \\ +d_p(p_{j+1}, p_i)^p - d_p(p_i, p_1)^p = 0 & \text{if } j \leq i-2 \\ (d_p(p_i, p_1)^p - \sum_{f=1}^{i-2} |\ p'_{i,f}\ |^p)^{\frac{1}{p}} & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

Additionally the embedding is determined in polynomial time.

PROOF. By denoting as p_i the i-th point of the dataset, the aforedescribed requirements are precisely captured by the system of non linear equations (31).

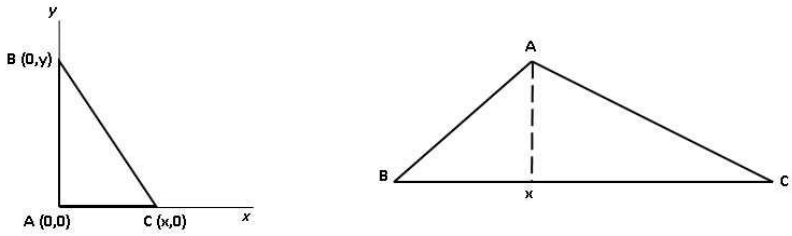
$$d'_p(p_i, p_j) = d_p(p_i, p_j),\ j = 1...k+1,\ i = 1...k+1 \quad (31)$$

Despite its non linear nature, this system can be solved fast through an iterative set of polynomial equations. We employ the following technique for achieving this goal. The first point p_1 is mapped in the beginning of the coordinates system, thus is attributed coordinates $O = (0, 0, ..., 0)$. The second point is projected under the requirement that $d'_p(p_1, p_2) = d_p(p_1, p_2)$, which essentially can be expressed as $d'_p(O, p_2) = d_p(p_1, p_2)$. Obviously, p_2 can be mapped at any point lying on the circumference of a hypersphere with center p_1 and radius $d_p(p_1, p_2)$, however we choose to embed it at $(d_p(p_1, p_2), 0, 0, ..., 0)$. Essentially our choice is a simple verification of the fact that the distance between two points can be expressed in an 1D space. Having projected p_2 we proceed with p_3. In this case, our system (32) is augmented with one additional equation .

$$d'_p(p_3, p_i) = d_p(p_3, p_i),\ i = 1, 2 \quad (32)$$

The reader may notice that the system in question has an infinite number of solutions. Indeed, we have k unknowns and only two equations. Geometrically, our equations define two hypersheres; any of the points that lay in their intersection can be the projection of p_3 in R^k. In order to overcome this, we calculate as before, only the minimum number of coordinates that are needed in order for our prerequisites to hold true and assign a zero value to the rest. The miminum number of non-zero coordinates is set to $i-1$, where i is the index of the point under projection. Consequently, by expanding both equations and subtracting the second from the first, we assign to the first coordinate of p_3 the single root (recall Theorem (4)) of $|\ p'_{3,1}\ |^p - |\ p'_{3.1} - p'_{2,1}\ |^p - d_p(p_3, p_1)^p + d_p(p_3, p_2)^p = 0$. The second coordinate is derived by substituting $p'_{3,1}$ in the first equation, thus deriving $|\ p'_{3,2}\ | = (d(p_3, p_1)^p - |\ p'_{3,1}\ |^p)^{\frac{1}{p}}$.

Adhering to the above methodology we define the non linear system (33) for the third landmark and proceed accordingly in order to define the embedding of p_4 in R^k.



(a) Orthogonal triangle BAC (b) Triangle ABC with its altitude Ax

Fig. 17. .

$$d'_p(p_4, p_i) = d_p(p_4, p_i), i = 1, 2, 3 \tag{33}$$

We expand all three equations and calculate $p'_{4,1}$ and $p'_{4,2}$ by subtracting the second and third equations respectively from the first. The final non-zero coordinate, $p'_{4,3}$, is defined by substituting the calculated values in the first equation. By iteratively applying this procedure for all $k-1$ points we manage to embed them in R^k. Based on this methodology we derive the set of equations (34).

$$p_{i,j} = \begin{cases} \mid p'_{i,j} \mid^p - \mid p'_{i,j} - p'_{j+1,j} \mid^p + \sum_{f=1}^{j-1} \mid p'_{i,f} \mid^p - \sum_{f=1}^{j-1} \mid p'_{i,f} - p'_{j,f} \mid^p \\ +d_p(p_{j+1}, p_i)^p - d_p(p_i, p_1)^p = 0 & \text{if } j \leq i-2 \\ (d_p(p_i, p_1)^p - \sum_{f=1}^{i-2} \mid p'_{i,f} \mid^p)^{\frac{1}{p}} & \text{if } j = i-1 \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

The cost of this procedure is polynomial. The requirements are $O(ck^2)$ where c is the cost of the method employed for determining the root of the equation. □

Observation 3. *Given an orthogonal triangle $\widehat{BAC}$ and a Minkowski distance metric p, the length of the hypotenuse BC of the triangle is given by $BC^p = AB^p + AC^p$.*

Proof. Assuming triangle BAC as in Fig. 17(a) the lengths of its sides are $AC = x$, $AB = y$, $BC = (|x|^p + |y|^p)^{\frac{1}{p}}$. Obviously, we conclude that $BC^p = AB^p + AC^p$. □

Observation 4. *Given a triangle $\widehat{BAC}$ and a Minkowski distance metric p, the length of the segment Bx, where x is the intersecting point of the altitude of angle $\widehat{A}$ with line BC is given by the single root of $Bx^p - (BC - Bx)^p - AB^p + AC^p = 0$*

Proof. Assume triangle ABC as in Fig. 17(b). The application of the Pythagorean theorem on triangle ABx gives $AB^p = Ax^p + Bx^p$. Accordingly on triangle ACx we derive $AC^p = Ax^p + (BC - Bx)^p$ since $BC = Bx + Cx$. Subtracting the second equation from the first we derive $Bx^p - (BC - Bx)^p - AB^p + AC^p = 0$ which according to theorem 4 has a single root in R. □

Lemma 6. Using any two landmarks L_1, L_2 and a Minkowski distance metric p, FEDRA can project any two points A, B in a given low-dimensional space while guaranteeing

that their new distance $A'B'$ will be bounded by $(AB^p - \Delta)^{\frac{1}{p}} \leq A'B' \leq (AB^p + \Delta)^{\frac{1}{p}}$ with $\Delta = (AA_y + BB_y)^p - (AA_y - BB_y)^p$ where AA_y, BB_y are the lengths of the altitudes of triangles L_1AL_2, L_1BL_2 respectively.

Proof. We will now extend the analysis of Section 4 with respect to the distortion for any Minkowski distance metric p. Towards this end we will employ Observations 3 and 4. Following the same analysis as in Paragraph 4.4.2 but using the previous generalizations, in the high-dimensional space we obtain that $AB^p = A''B^p + (x-y)^p$ and $AB^p \leq (AA_y + BB_y)^p + (x-y)^p$ where x and y are obtained from the solution of equations $x^p - (L_1L_2 - x)^p + AL_2^p - AL_1^p = 0$ and $y^p - (L_1L_2 - y)^p + BL_2^p - BL_1^p = 0$ respectively which according to Theorem 4 have a single root in R. Using Observation 3 we also obtain $AAy = (AL_1^p - x^p)^{\frac{1}{p}}$ and $BBy = (BL_1^p - y^p)^{\frac{1}{p}}$.

The new distance between A', B' is either $A'B'^p = (x-y)^p + (AA_y - BB_y)^p$ or $A'B'^p = (x-y)^p + (AA_y + BB_y)^p$. We define as $\Delta = (AA_y + BB_y)^p - (AA_y - BB_y)^p$ the difference between the obtained values and obtain the lower bound of $A'B'$ as $(AB^p - \Delta)^{\frac{1}{p}} \leq A'B'$.

The upper bound can be derived similarly. We have that $|AA_y - BB_y| \leq A''B \Rightarrow (AA_y - BB_y)^p \leq A''B^p \Rightarrow (AA_y - BB_y)^p + (y-x)^p \leq A''B^p + (y-x)^p \Rightarrow (AA_y - BB_y)^p + (y-x)^p \leq AB^p \Rightarrow (AA_y - BB_y)^p + (y-x)^p + \Delta \leq AB^p + \Delta \Rightarrow (AA_y + BB_y)^p + (y-x)^p \leq AB^p + \Delta$ and since $(AA_y - BB_y)^p + (y-x)^p \leq (AA_y + BB_y)^p + (y-x)^p$ we derive the upper bound $A'B'^p \leq AB^p + \Delta$.

Consequently, $(AB^p - \Delta)^{\frac{1}{p}} \leq A'B' \leq (AB^p + \Delta)^{\frac{1}{p}}$. □