FEDRA: A Fast and Efficient Dimensionality Reduction Algorithm

Panagis Magdalinos, Christos Doulkeridis and Michalis Vazirgiannis *

Abstract

Contemporary data-intensive applications generate large datasets of very high dimensionality. Data management in high-dimensional spaces presents problems, such as the degradation of query processing performance, a phenomenon also known as the curse of dimensionality. Dimensionality reduction (DR) tackles this problem, by efficiently embedding data from high dimensional to lower dimensional spaces. However, the large scale and dynamism of generated data calls for methods of low time and space complexity, features that are hardly combined in the majority of existing DR algorithms. Motivated by this fact, in this paper we propose FEDRA, a fast and efficient dimensionality reduction algorithm that uses a set of landmark points to project data to a lower dimensional Euclidean space. FEDRA is both faster and requires less memory than other comparable algorithms, without compromising the projection's quality. We theoretically assess the quality of the resulting projection and provide a bound for the error induced in pairwise distances. Furthermore, we present two extensions of FEDRA that improve the quality of the projection, suitable for applications that can tolerate higher processing costs. We prove the validity of our claims both theoretically and experimentally, by comparing our algorithm against prominent approaches, such as FastMap, LMDS, PCA, SVD and Random Projection.

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. In such high dimensional spaces, query processing performance degrades, a phenomenon known as the *curse of dimensionality* [11]. Thus, typical data mining tasks, such as clustering or classification become ineffective [5], and data mining applications rely on dimensionality reduction (DR) as a pre-processing step for solving these problems.

Dimensionality reduction algorithms address this challenge, by projecting data from the original high dimensional space R^n to a new, lower dimensional space R^k (usually $k << n$). The objective of the DR methodology is to retain the distances between points or other statistical properties in the lower dimensional space. Nevertheless, the vast amount of generated data dictates methods that are both fast and exhibit low memory requirements. These features are hardly combined in the majority of existing DR algorithms. Therefore, there is a need for provably fast and effective algorithms for efficient processing of datasets of both high dimensionality and cardinality.

In this context, we propose FEDRA, a fast and efficient dimensionality reduction algorithm that follows the general principles of the *landmark-based* DR paradigm [9]. FEDRA selects randomly a limited number of k points (henceforth called *landmarks*) of the initial space and provides their embedding in R^k, such that the pairwise distances of all landmarks are totally preserved. Each of the remaining points is projected to R^k by requiring that its distances from all landmark points are preserved. Using this simple projection methodology FEDRA achieves quality of results comparable to the most prominent DR algorithms, while being faster and consuming less space in memory.

The contribution of this work is manifold:

- We propose a new algorithm, FEDRA, which exhibits lower time and space requirements than other prominent DR algorithms, while successfully reproducing the original data structure. Furthermore, we show that FEDRA is applicable with the use of any Minkowski distance metric d_p ($p > 1$), making it attractive for applications that cannot employ the Euclidean distance.
- We provide theoretical guarantees for FEDRA's projection quality and offer a quantitative analysis of the induced error. Moreover we derive upper and lower bounds for the error in approximating the original pairwise distances in the projected space.
- We introduce two algorithms, one for landmark selection and one for the actual data embedding, to further improve the projection quality, suitable for applications that tolerate higher processing costs. In addition, we define an analytical expression which enables the offline assessment of the quality of the resulting projection.
- Through extensive experiments on very large datasets, both in terms of dimensionality and cardinality, we demonstrate that the quality of results of FEDRA is comparable to several prominent DR algorithms, while exhibiting lower complexity and costs than any other of the employed algorithms.

*All authors affiliated with Athens University of Economics and Business

The rest of this paper is organized as follows: in section 2, we review the most prominent DR algorithms. We motivate the need for a new fast and efficient algorithm in section 3, through an analysis of the complexity of existing techniques. In section 4, we present FEDRA, while in section 5 we theoretically analyze our algorithm's performance and quality. In section 6, we present two extensions of the basic algorithm for better quality of results, when higher processing costs can be tolerated. Additionally, we offer an offline algorithm that assesses the quality of a projection and re-initializes FEDRA, if necessary. The experimental study is presented in section 7. Finally, in section 8, we conclude the paper and sketch future research directions.

2 Related Work

Dimensionality reduction problems can be broadly classified into three distinct categories [6]. *Hard problems*, where data is defined in a space consisting of hundreds/thousands of coordinates and drastic dimensionality reduction is necessary, *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.

While there exist different methods of assessing the quality of the output of a DR algorithm, the most usually employed evaluation metric is *stress*. Simply stated, *stress* evaluates the preservation of pairwise distances in the projected space and is defined as:

$$\sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d}(d_{ij}^{(n)^2}-d_{ij}^{(k)^2})/\sum_{i=1}^{d}\sum_{j=1}^{d}d_{ij}^{(n)^2}}$$

where $d_{ij}^{(n)}$,$d_{ij}^{(k)}$ denote the Euclidean distance between points i and j in the n-dimensional and k-dimensional space respectively. Another application-oriented method is the comparison of clustering and classification performance prior and after DR. For the subsequent analysis, we assume that the goal is to project d data vectors defined in R^n into the R^k subspace while preserving their pairwise distances.

One of the initial methods proposed was multidimensional scaling (MDS) often referred to as classic MDS [23]. Simply stated, the algorithm initially embeds all points randomly in the new space and iteratively repositions them, in order to minimize the exhibited stress value. Despite its merits, MDS exhibits two serious drawbacks, namely high computational complexity and high memory requirements. Any subsequent addition of a data point in the projection results in $O(d)$ overhead [12].

Principal Components Analysis (PCA) [21] is closely related to MDS and considered as the best method in terms of the quality of the produced projection. Given d data vectors defined in R^n and represented by a two dimensional matrix $X_{d\times n}$, PCA projects them in R^k by computing the product $X_{d\times n}P_{k\times n}^T$, where $P_{k\times n}$ contains as row elements the eigenvectors of the covariance matrix that correspond to its k largest eigenvalues and $X_{d\times n}$ the original data matrix with means subtracted across each dimension.

The high quality results produced by MDS impacted subsequent research efforts significantly, which focused mainly in resolving its scaling problems. Faloutsos and Lin introduced FastMap [12], an algorithm that successfully addressed the computational complexity issue. FastMap embeds data from space R^n to R^1 by projecting on a hyperplane perpendicular to the line defined by the two most distant points (henceforth referred to as pivots) in the processed hyperplane. Iterative application of the latter, results in the embedding of the processed point in the R^k space. The projection is carried out based on the Pythagorean Theorem and its generalization, the cosine law.

FastMap's agnostic nature towards the original dimensionality of data, makes it an ideal candidate for the processing of collections where only pairwise distance information (and not absolute position) is available. A significant drawback of FastMap is its memory requirements, which reach $O(d^2)$, as it requires the maintenance of all pairwise distances in main memory. 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 to $O(dk(k+n))$. In the context of this paper we follow the first outlined variation which is also the one appearing in the original publication [12].

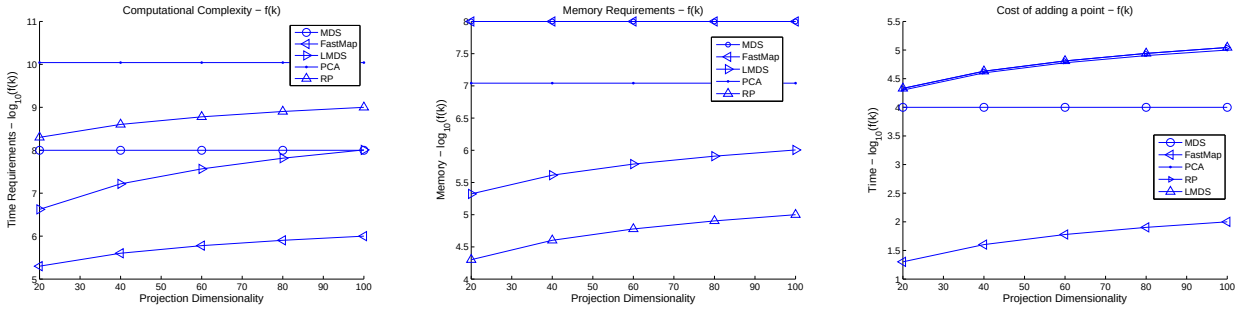
In order to address the high memory requirements of MDS, a slightly different embedding methodology, namely landmark-based projection, has been defined. Instead of trying to map all data simultaneously to the new space, landmark-based projection algorithms initially extract a small fraction of points which are projected in the new space. Based on the assumption that these points remain fixed (landmarks in the new space), the rest of the dataset is projected by employing distance preservation techniques.

The first method obeying this scientific paradigm was Triangulation based Sequential Mapping (TSM) [19]. TSM has been specifically designed and tuned in order to address visualization problems. TSM projects each point independently in a two dimensional Euclidean space using two landmarks. Landmarks are different for each point and their selection is accomplished with two alternative strategies, both operating on the dataset's minimum spanning tree. Subsequent projection is implemented through triangulation techniques. TSM guarantees that $2n-3$ out of the $\frac{n(n-1)}{2}$ overall pairwise distances will be exactly preserved.

The most prominent algorithm of this methodology is Landmark Multidimensional Scaling (LMDS) [9], which directly addresses the scalability problem of MDS. Initially, LMDS selects f landmark points, with the constraint $f > k$, and projects them in the new space with the use of MDS.

Table 1: Comparative assessment of the presented methods.

Algorithm	Computational Complexity	Memory Requirements	Single Point Addition
MDS	$O(d^2)$	$O(d^2)$	$O(d)$
PCA/SVD	$O(n^3 + n^2 d)$	$O(n^3 + nd)$	$O(kn)$
RP	$O(dkn)$	$O(kn)$	$O(kn)$
LMDS	$O(kfd + f^2 + f^3)$	$O(fd)$	$O(f(n+k))$
FastMap	$O(dk)$	$O(d^2)$	$O(k)$



(a) Computational complexity (log scale) (b) Memory requirements (log scale) (c) Cost of adding new point (log scale)

Figure 1: Graphical comparison of the complexity of DR algorithms.

Afterwards a distance-based triangulation procedure, which uses as input distances to already embedded landmark points, determines the projection of the remaining points. Principal components analysis [21] is optionally used to align the result to the principal axes of the data set.

Advances in the area of landmark-based algorithms have resulted in the definition of several algorithms, such as Local Linear Embedding (LLE) [20], Isomap [22][1], and K-Landmarks [16]. All these successfully address the memory requirement issue of MDS, but necessitate polynomial execution time with respect to the dimensionality of the projection space, which may cause problems if k is comparable to n. Nevertheless, they have inspired the definition of FEDRA and they served as basis for its theoretic justification.

Random Projection is another technique used for dimensionality reduction. The data matrix X is embedded in the R^k space with the use of a randomly generated matrix (R) through multiplication $X'_{d\times k} = X_{d\times n} R^T_{k\times n}$. The method is based on the Johnson-Lindenstrauss lemma [14]. An elementary proof of this lemma is provided by Dasgupta and Gupta in [8]. The main remaining issue is the definition of the projection matrix. Achlioptas proposes two simple distributions [3] that prove rather robust and can easily be applied to large datasets.

[1]LLE and Isomap belong in another DR paradigm, namely Non Linear Dimensionality Reduction. Landmark selection is also used in this case, but not strictly obeying the landmark based projection methodology.

3 Motivation

We present a comparative assessment of the most prominent DR methods, in terms of computational complexity, memory requirements and cost of adding a new point in an already projected dataset. Table 1 summarizes these results in terms of $O()$ notation. In order to ensure a fair comparison to LMDS, we assume random selection of landmarks and omit the optional PCA alignment step. The notation is based on the analysis of the previous section.

Using the mathematical formulas of Table 1 we simulate a hard DR problem (projection where $k \leq 0.1n$) and measure the time, space requirements and the cost for a new point addition. We assume that n=1000, d=10000 and k=20...100 (2% - 10% of n). For LMDS, we randomly select the least required number of landmarks, specifically k+1, and omit the optional PCA alignment step. The results are depicted in Figure 1.

A first observation is that FastMap is indeed the algorithm with the lowest time complexity, while mainly Random Projection and LMDS guarantee low memory requirements. However, FastMap fails to address the second drawback of MDS, namely the intensive memory requirements, in the order of $O(d^2)$.

Despite their good results, SVD-based methods also exhibit poor scaling quality [4]. Moreover, the eigen analysis employed in these algorithms results in high computational complexity and high memory requirements. Another prob-

lem is their inability to handle data updates efficiently (i.e. massive additions or deletions of points). Although methods of bounding the error [18] and updating an already existing decomposition when the error surpasses a predefined bound do exist [24], the cost of updating is excessively high, marginally reaching the cost required for the recomputation of the decomposition. On the other hand, Random Projection seems a suitable method regarding scaling, as it is immune to data additions, produces results rather fast and has low space complexity.

According to this analysis we conclude that the ideal dimensionality reduction method should exhibit quality of results equal to that of PCA, space complexity lower than or equal to Random Projection or at least LMDS and time complexity lower than or at least equal to FastMap. Such a method should also provide solution to hard dimensionality reduction problems on datasets of high cardinality. This discussion motivates the design of a new DR algorithm that successfully addresses all of these requirements.

4 The FEDRA Algorithm

In this section, we present FEDRA, a 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 the data distribution. It is designed to handle all classes of dimensionality reduction problems, however it emphasizes on the hard ones. The intuition of the approach is based on the landmark-based projection methodology [9, 16, 19]. However FEDRA introduces significant advances in terms of computational cost and projection quality, compared to existing landmark-based algorithms and other DR methods. More specifically:

- FEDRA acquires the projection through an iterative scheme of polynomial equations, thus achieving low computational complexity and memory requirements.
- In comparison to several DR algorithms that are restricted to the Euclidean distance, our approach is applicable to any Minkowski distance metric ($p > 1$).
- Additionally, the proposed method guarantees the exact preservation of a significant amount of the initial pairwise distances.
- Finally, FEDRA establishes a bound for the error introduced due to the reduction, thus providing theoretical guarantees for the quality of the projection.

In the following analysis, we assume that the dataset is composed of d n-dimensional points that are going to be embedded in R^k, with k significantly lower than n ($k \leq 0.1n$). The Minkowski distance between two points x_i and x_j in an n-dimensional space is depicted as $d_p^{(n)}(\overrightarrow{x_i}, \overrightarrow{x_j})$.

4.1 Landmark-based DR Algorithm. Initially, k landmarks are selected from the dataset. The simplest selection approach is to randomly pick the points from the dataset, which incurs $O(k)$ time cost. The algorithm embeds the selected set of landmarks L into R^k. The first processed landmark (randomly selected from L) is mapped to the beginning of the coordinates systems axis, while for each of the remaining landmarks it is required that distances between already embedded landmarks and the processed one are retained. In the final step, each remaining non-landmark point is projected independently with respect to the rest of the dataset by requiring that its distances to the landmarks remain the same in the embedding space. This results in the definition of a non-linear system of equations. However we show that, despite its non-linear nature, the system in question can be solved easily in linear polynomial time.

We first provide an overview of the algorithm in a descriptive manner for ease of presentation. FEDRA requires as input only the projection dimensionality (k), the pairwise distances between the points of the dataset (D) and the employed Minkowski distance metric (p). One necessary condition for the successful execution of the approach is that the triangular inequality is sustained in the original space.

In the first step, k points are randomly selected populating set $L = \{l_1, l_2, ...l_k\}$. The first landmark point, l_1, is projected at point $l_1^{(k)}$=(0, 0, ..., 0). Since l_2 should retain its distance to l_1, it is mapped at point ($d_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{l_2})$, 0, ..., 0), where $d_p^{(n)}(\overrightarrow{l_i}, \overrightarrow{l_j})$ denotes the Minkowski distance of landmarks i and j in the n-dimensional space. The third landmark is embedded under the constraint that its distances to all already projected landmarks are retained ($d_p^{(n)}(\overrightarrow{l_i}, \overrightarrow{l_3}) = d_p^{(k)}(\overrightarrow{l_i}, \overrightarrow{l_3})$ for $i = 1, 2$). This leads to the following system of non-linear equations (l_{ij} denotes the j-th coordinate of l_i):

$$l_{31}^{(k)p} + l_{32}^{(k)p} + ... + l_{3k}^{(k)p} = d_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{l_3})^p \quad (1)$$

$$(l_{31}^{(k)} - l_{21}^{(k)})^p + l_{32}^{(k)p} + ... + l_{3k}^{(k)p} = d_p^{(n)}(\overrightarrow{l_2}, \overrightarrow{l_3})^p \quad (2)$$

The solution of this system is trivial considering the fact that only two non-zero coordinates are required for our prerequisites to hold true. The latter is based on the fact that the pairwise distances of k objects can be exactly preserved in a $k-1$ dimensional space. Consequently, we simplify the system by assuming that $l_{3j}^{(k)} = 0$ for each $j \geq 3$. Generalizing, the number of non-zero coordinates required for the projection of the i-th landmark is $i-1$. Following the same technique during the subsequent landmarks projections, the j-th coordinate of the i-th landmark is provided by the following equations (equation set A) ($i = 1..k, j = 1..k-1$):

$$\mathrm{l}_{ij}^{(k)} = \begin{cases} \mathrm{d}_p^{(n)}(\overrightarrow{l_{j+1}}, \overrightarrow{l_i})^p - d_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{l_i})^p - \\ \sum_{f=1}^{p} \binom{p}{f} l_{ij}^{(k)p-f}(-l_{j+1j}^{(k)f}) + \\ \sum_{f=1}^{j-1}[l_{if}^{(k)p} - (l_{i,f}^{(k)} - l_{j+1f}^{(k)})^p] = 0 & \text{if } j \leq i-2 \\ (\mathrm{d}_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{l_i})^p - \sum_{f=1}^{i-2} l_{i,f}^{(k)p})^{\frac{1}{p}} & \text{if } j = i-1 \\ 0 & otherwise \end{cases}$$

The cost of finding the p-1 roots of the resulting polynomial with the use of a simple Horner scheme requires O(p) time. Consequently, since the computed non-zero coordinates for all landmarks in this step are $k(k-1)/2$, the cost of embedding the landmarks is $O(pk^2)$[2]. As the resulting polynomials describing the value of each dimension may have up to p-1 roots, there exists a problem of which root to select.

Figure 2 provides a simple example of the projection of three points from R^3 to R^2 with $p = 2$. During the first step, landmarks (depicted as stars) are projected in the new space. The first one is placed on the beginning of the coordinate system. The second landmark is mapped on the x axis at $(d_o, 0)$, where d_o signifies their distance in the original space. The third point can be embedded in two different positions, both symmetric with respect to the line defined by the two initial landmarks. Symmetric coordinates are derived in this case due to the square root computation employed for the calculation of the last coordinate. Since roots depict the various symmetric coordinates that a point can be embedded in the lower dimensional space, it suffices to choose any of them, providing that the same selection scheme is employed for all remaining points.

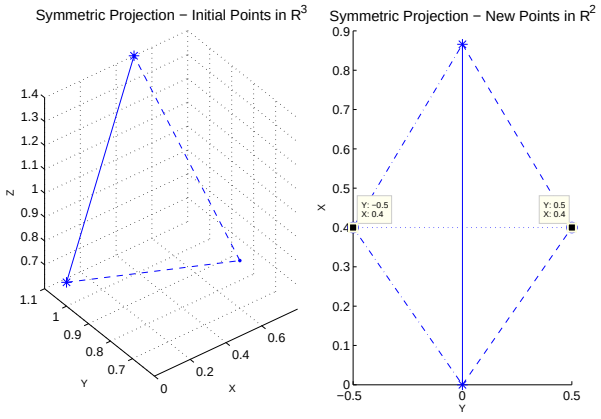


Figure 2: Symmetric projection from R^3 to R^2. Stars depict the landmark points and squares the two possible projections. Axes are rotated to ease illustration.

In the final step, each remaining point is projected independently of the rest of the dataset by requiring that its distances to the landmarks are sustained in the projection space. This defines the following non-linear system of equations for any point x:

$$d_p^{(k)}(\overrightarrow{x}, \overrightarrow{l_i}) = d_p^{(n)}(\overrightarrow{x}, \overrightarrow{l_i}) \text{ for } i = 1...k \quad (3)$$

Assuming the projection of point x_i in the new space and analyzing equation (3) in the same way as equation (2) we obtain the equations that provide the projection of point x_i ($i = 1...d, x_i \notin L$) in the low dimensional space (equation set B):

$$\mathrm{x}_{ij}^{(k)} = \begin{cases} \mathrm{d}_p^{(n)}(\overrightarrow{l_{j+1}}, \overrightarrow{x_i})^p - d_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{x_i})^p - \\ \sum_{f=1}^{p} \binom{p}{f} x_{ij}^{(k)p-f}(-l_{j+1j}^{(k)f}) + \\ \sum_{f=1}^{j-1}[x_{if}^{(k)p} - (x_{i,f}^{(k)} - l_{j+1f}^{(k)})^p] = 0 & \text{if } j < k \\ (\mathrm{d}_p^{(n)}(\overrightarrow{l_1}, \overrightarrow{x_i})^p - \sum_{f=1}^{i-2} x_{i,f}^{(k)p})^{\frac{1}{p}} & \text{if } j = k \end{cases}$$

The computation of the k coordinates for each of the remaining $d-k$ points results in $O(pdk-pk^2)$ computational cost. Based on this example, FEDRA is straightforwardly derived and presented in Algorithm 1. 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 the coordinates.

Algorithm 1 FEDRA.

Input: Projection dimensionality (k), data distances in $R^n(D)$, Distance Metric (p)
Output: new dataset in R^k (P)
Create set of landmarks LS=$\emptyset$
Create new dataset P=$\emptyset$
for $i = 1$ **to** k **do**
  select landmark point p_i
  LS = $LS \cup p_i$
end for
Create set of projected landmarks PLS=$\emptyset$
for $i = 1$ to all landmarks l_i **do**
  l_i = Calculate landmark coordinates using the equations of Set A
  PLS=$PLS \cup l_i$
end for
P=$P \cup PLS$
for $i = 1$ to all remaining points x_i **do**
  x_i=Calculate remaining points coordinates using the equations of Set B
  P=$P \cup x_i$
end for

5 Theoretical Evaluation

After having introduced the FEDRA algorithm, we proceed to present its theoretical evaluation. We first analyze its

[2]In the general case, factor p is significantly lower than k and does not affect the overall cost

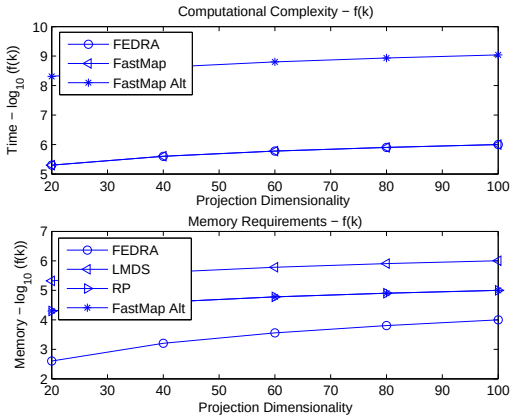


Figure 3: FEDRA's computational complexity and memory requirements.

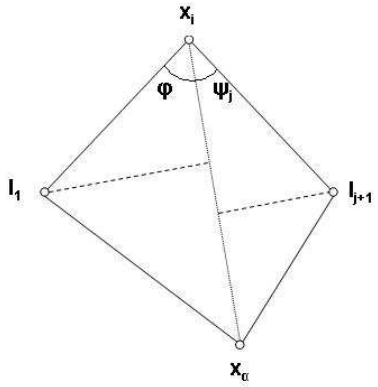


Figure 4: Application of the cosine law, in order to determine the error bound.

computational complexity, then we prove that the error induced by the projection is bounded, and finally we assess the projection's quality.

5.1 Computational Complexity. Based on the algorithmic description, FEDRA exhibits slightly better computational complexity than FastMap (both evaluated with the Euclidean distance), and additionally significantly lower memory requirements than LMDS and Random Projection. FEDRA requires k^2 space in the first step (landmarks pairwise distances) and $(k^2+k)/2$ in the second (embedded landmark coordinates and distances between processed points and the landmarks in the original space), which is analogous to $O(k^2)$.

The latter is clearly depicted in Figure 3. All measurements are computed in accordance with the configuration outlined in section 3, in order to ensure a fair comparison. The alternate FastMap setup (FastMap Alt) has also been included in the graph to ensure completeness.

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 is as fast and efficient as in FastMap[3]. Concluding, FEDRA successfully addresses the two major disadvantages of MDS and appears as an efficient solution in cases of hard dimensionality reduction problems on large datasets.

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 significant deterioration of the projection's quality. However, existing theory [9] and experiments suggest that this simplification is acceptable. Moreover, in the following paragraphs, we prove that the distance between any two non-landmark points in the new space is dominated by their distance in the original space, while a significant percentage of initial pairwise distances remains unaltered in spite of the projection. The analysis is done for the Euclidean distance metric ($p=2$) however its generalization can be straightforwardly derived.

5.2 Error Bound. The new distance between any two points x_i, x_a can be defined as $d_2^{(k)}(\overrightarrow{x_i},\overrightarrow{x_a}) = \sqrt{\sum_{c=1}^{k}(x_{i,c}^{(k)}-x_{a,c}^{(k)})^2}$. By approximating the new distance in the k-dimensional space we can assess the quality of the projection. This is done with the use of the equations of Section 4 for $p=2$ as follows: $x_{i,j}^{(k)} - x_{a,j}^{(k)} = (d_2^{(n)}(\overrightarrow{l_{j+1}},\overrightarrow{x_i})^2 - d_2^{(n)}(\overrightarrow{l_1},\overrightarrow{x_i})^2 - d_2^{(n)}(\overrightarrow{l_{j+1}},\overrightarrow{x_a})^2 + d_2^{(n)}(\overrightarrow{l_1},\overrightarrow{x_a})^2 - 2\sum_{f=1}^{j-1} l_{j+1,f}^{(k)}(x_{i,f}^{(k)} - x_{a,f}^{(k)}))/2l_{j+1,j}^{(k)}$ for $i,j < k$ whereas for $i,j = k$: $x_{i,j} - x_{a,j} \leq \sqrt{d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{l_1})^2 + d_2^{(n)}(\overrightarrow{x_a},\overrightarrow{l_1})^2}$

In Figure 4, by applying the cosine law on the triangles defined by (x_i,l_1,x_a) and (x_i,l_{j+1},x_a) we obtain for $i,j<k$:$x_{i,j}^{(k)} - x_{a,j}^{(k)} = d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})(\cos\phi d_2^{(n)}(\overrightarrow{l_{j+1}},\overrightarrow{x_i}) - \cos\psi_j d_2^{(n)}(\overrightarrow{l_1},\overrightarrow{x_i}) - \sum_{f=1}^{j-1} l_{j+1,f}^{(k)}(x_{i,f}^{(k)} - x_{a,f}^{(k)}))/l_{j+1,j}^{(k)}$ where ϕ , ψ_j signify the angles defined by lines $\overrightarrow{x_i x_a}, \overrightarrow{x_i l_1}$ and $\overrightarrow{x_i x_a}, \overrightarrow{x_i l_{j+1}}$ respectively.

By defining $D_j = (\cos\phi d_2^{(n)}(\overrightarrow{l_{j+1}},\overrightarrow{x_i}) -$

[3]Evaluated with the Euclidean distance

$\cos\psi_j d_2^{(n)}(\overrightarrow{l_1},\overrightarrow{x_i}))$ the expression can be simplified and depicted as $x_{i,j}^{(k)} - x_{a,j}^{(k)} = d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})(D_j - \sum_{f=1}^{j-1} l_{j+1,f}^{(k)}(x_{i,f}^{(k)} - x_{a,f}^{(k)}))/l_{j+1,j}^{(k)}$

Notice that the sum of differences $x_{i,f}^{(k)} - x_{a,f}^{(k)}$ can be recursively computed with the use of the initial formula. The resulting expression is of the form $d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})\sum_{f=1}^{j} F(D_f)$, where $F(D_f)$ signifies the linear combination of D_f with the landmark coordinates in the new space. Consequently the expression is also defined as $x_{i,j}^{(k)} - x_{a,j}^{(k)} = d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})(D_j - \sum_{f=1}^{j} F(D_f))/l_{j+1,j}^{(k)}$

Finally, $d_2^{(k)}(\overrightarrow{x_i},\overrightarrow{x_a}) = \sqrt{\sum_{c=1}^{k}(x_{i,c}^{(k)} - x_{a,c}^{(k)})^2} \leq \sqrt{\sum_{c=1}^{k-1}(x_{i,c}^{(k)} - x_{a,c}^{(k)})^2} + \sqrt{(x_{i,k}^{(k)} - x_{a,k}^{(k)})^2} \leq \sqrt{k-1}d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})\sqrt{\sum_{c=1}^{k-1} A_c^2} + \sqrt{d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{l_1}) + d_2^{(n)}(\overrightarrow{x_a},\overrightarrow{l_1})}$. where $A_c = D_c - \sum_{f=1}^{c} F(D_f))/l_{c+1,c}^{(k)}$.

A significant first result of this analysis is that the resulting embedding is primarily influenced by the points' original pairwise distances and significantly less by the landmarks themselves, showing that the landmarks' random selection process is acceptable.

We now define $\Delta_{i,a} = |d_2^{(k)}(\overrightarrow{x_i},\overrightarrow{x_a}) - d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a})|$ the variation between the distances of points x_i,x_a in the original and lower dimensional space and obtain $(\sqrt{(k-1)\sum_{c=1}^{k-1} A_c^2} - 1)d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a}) + \sqrt{d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{l_1}) + d_2^{(n)}(\overrightarrow{x_a},\overrightarrow{l_1})} \leq \Delta_{i,a} \leq (\sqrt{(k-1)\sum_{c=1}^{k-1} A_c^2} + 1)d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{x_a}) + \sqrt{d_2^{(n)}(\overrightarrow{x_i},\overrightarrow{l_1}) + d_2^{(n)}(\overrightarrow{x_a},\overrightarrow{l_1})}$. Thus we derive both an upper and a lower bound for the pairwise distance variation due to the projection between any two points of the dataset.

5.3 Quality Assessment. The previous analysis bounds the error in the distance deviation between any two points due to the projection while highlighting the acceptability of the landmarks' random selection process. We will now attempt to go one step further and calculate the exact amount of pairwise distances for which this analysis is applicable ($\Delta_{i,a} \neq 0$). By construction, FEDRA guarantees that a certain amount of pairwise distances will be exactly preserved. Indeed, the landmarks' selection and projection phases preserve exactly $\frac{k(k-1)}{2}$ pairwise distances while the subsequent embedding of the remaining $(n-k)$ data points retains another $(n-k)k$ distances. Assuming that *p* signifies the dimensionality of the projection space as a percentage of the number of the original dimensions, the percentage *x* of pairwise distances that will remain unaffected due to the projection is given by the subsequent analysis.

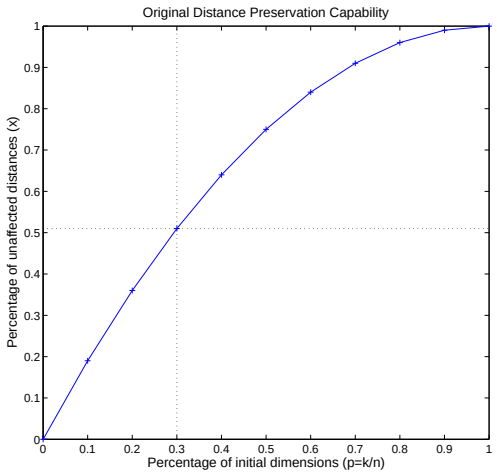


Figure 5: FEDRA's original distances maintenance ability.

$$x=\frac{2nk-k^2-k}{(n-1)n}=\frac{k}{n}\frac{2n-k-1}{(n-1)}$$
$$x=p\frac{2n-k-1}{n-1}=p\frac{2-p-1/n}{1-1/n}$$

Assuming that n is excessively large ($1 \ll n$) we can easily ignore $\frac{1}{n}$ since its value will be close to zero, thus deriving the following:

$$x=p(2-p)\ ,\ 0 < p \leq 1$$

Despite the simplicity of this analysis, it proves that a significant percentage of distances will remain unaltered despite the projection. Figure 5 shows the percentage *x* of distances that are not modified, because of the projection. The x-axis depicts the dimensionality of the projection space calculated as a percentage of the number of the original dimensions (*p*). For example, if $\frac{k}{n} = p = 0.3$ then projecting with FEDRA will not affect 51% of the initial distances.

6 FEDRA Extensions

In this section we introduce three algorithms, complementary to the FEDRA approach. The first two aim at improving the projection quality and are suitable for applications that can tolerate higher processing costs. The third one is an offline evaluation procedure which assesses the quality of the projection and triggers, if necessary, the re-initialization of FEDRA.

6.1 Intentional Landmark Selection. The theoretic analysis previously presented signifies that the pairwise distances of all points in the new space are partially influenced by the landmarks positions in the original space. Consequently, a natural question is whether significant benefits can be achieved by employing another selection scheme. For this purpose, we introduce a greedy algorithm that intentionally chooses landmark objects, in such a way that they increase

Algorithm 2 Projection Quality Evaluation.

Input: Original Distances ($d_p^{(n)}$), New Distances ($d_p^{(k)}$), Threshold (t)
Output: True/False
Set counter = 0
for $i = 1$ **to** $d - k$ **do**
  for $j = i$ **to** $d - k$ **do**
    if ($\frac{|d_p^{(n)}(\overrightarrow{x_i},\overrightarrow{x_j}) - d_p^{(k)}(\overrightarrow{x_i},\overrightarrow{x_j})|}{d_p^{(n)}(\overrightarrow{x_i},\overrightarrow{x_j})} \longmapsto 1$) **then**
      counter = counter + 1
    end if
  end for
end for
if counter $>$ t **then**
  return False
end if
return True

the quality of the projection. We refer to this heuristic as Intentional Landmark Selection heuristic (ILS).

The basic intuition behind this approach is that landmarks should ideally be distributed over the whole dataset. The algorithm works in the following way. The landmark set is initialized by randomly selecting the first landmark object l_1. The next landmark object l_2 would be the one that exhibits maximum distance from l_1. This process iteratively selects as next landmark l_i the object that maximizes: $\sum_{j=1}^{i-1} d(l_i, l_j)$. The algorithm terminates when k landmarks have been selected.

As already mentioned, distances between any point and the landmarks remain unaffected during projection. Therefore data distribution is preserved if landmarks are selected so that they span throughout the whole dataset.

6.2 Distance Minimization. Another issue, straightforwardly rising from the main analysis of the algorithm is the root selection process. A polynomial of $p - 1$ degree has at most $p - 1$ roots so it is normal to question the validity of the random selection process proposed. Although the assessment analysis of the previous section and the experiments suggest that random selection is acceptable, in this paragraph we present a heuristic which provides a solution to this issue.

The heuristic is based on the stress minimization criterion of MDS and emphasizes on the subsequent embedding of the remaining points. Recall that each point can be embedded to more than one points in the projection space, due to the degree (p) of the equations. Instead of using a specific root selection scheme, the algorithm greedily selects the point that minimizes the distance deviation from already projected points (not landmarks). We refer to this heuristic as Distance Minimization Heuristic (DMH). Despite its effectiveness, this heuristic comes with a cost, since its computational complexity is in the order of $O((d-k)pd)$. Table 2 provides the complexity of FEDRA and its extensions in a comparable way to Table 1.

Table 2: The FEDRA algorithm assessment.

Algorithm	Computational Complexity	Memory Requirements	Addition of data
FEDRA	$O(pdk - pk^2)$	$O(k^2)$	$O(pk)$
FEDRA-DMH	$O((d-k)pd)$	$O(dk)$	$O(pk)$
FEDRA-ILS	$O(pdk - pk^2)$	$O(dk)$	$O(pk)$

6.3 Offline Quality Assessment. The independent projection of each point with respect to the other non-landmark points is one of the factors for FEDRA's reduced complexity. However, this simplification may sometimes come with a cost, as it cannot always guarantee that pairwise distances between non-landmark points are also preserved. The latter is due to the fact that the new distance is approximated by the result of a linear combination of the initial distance as well as the distances between the selected landmarks. Therefore, there exists one potential case of failure. The latter is depicted in the assessment analysis when $D_j = 0$, or $\cos\phi d_2^{(n)}(\overrightarrow{l_{j+1}}, \overrightarrow{x_i}) = \cos\psi_j d_2^{(n)}(\overrightarrow{l_1}, \overrightarrow{x_i})$ for $j = 1..k$. In general, this situation scarcely occurs, as data would have to be symmetrically divided by a given set of k randomly chosen points. Nevertheless we provide an offline detection algorithm, which detects a problematic situation and triggers the re-initialization of FEDRA.

The proposed evaluation algorithm takes as input both the original ($d_p^{(n)}$) and new distances ($d_p^{(k)}$) and a threshold value which signifies the maximum number of points (t) which can be falsely embedded without deteriorating the overall quality of the projection. By evaluating expression $\frac{|d_p^{(n)}(\overrightarrow{x_i},\overrightarrow{x_j}) - d_p^{(k)}(\overrightarrow{x_i},\overrightarrow{x_j})|}{d_p^{(n)}(\overrightarrow{x_i},\overrightarrow{x_j})} \longmapsto 1$ for all points i,j we calculate whether the t value is exceeded. In case of an invalid projection the algorithm is re-initialized with a different set of landmark points. A formal presentation of the algorithm is presented in Algorithm 2.

The added value of the approach lays in the fact that it is executed offline, that is, after the actual reduction has terminated. Despite its high computational complexity, $O((d-k)^2)$ this heuristic can be efficiently implemented, in order to impose minimum memory overhead. Instead of having both distance arrays residing constantly in memory throughout the evaluation procedure, in each step one can load only the required fraction of information, that is only one line from each array, thus resulting in $O(d)$ memory

load.

7 Experimental Evaluation

In this section we present the experimental evaluation of FEDRA, which indeed verifies the expected performance and makes it an attractive solution for hard DR problems.

We carried out experiments on a number of datasets acquired by the UCI Machine Learning Repository [1]. We compare FEDRA to other well known approaches in order to assess the quality of the projection. Therefore, we focus on the how well data distribution features are maintained in the lower dimensional space. More specifically we compare the classification and clustering results of kNN [17] and K-Means [7] algorithms respectively, between the original and the target low dimensional space. kNN is initialized with $k = 10$ in all experiments, while K-Means with the number of classes of each dataset. Although kNN initialization is not optimal, it does not affect the validity of our results, due to the fact that we are measuring relative classification preservation.

We assess the performance of FEDRA by comparing its performance to that of the algorithms described in Section 2, with respect to the following metrics:

1. Stress: Projection quality, measuring the algorithm's original distance preservation capability. Each algorithm aims at minimizing this criterion.

2. Relative Classification Ability Maintenance[4] (RCAM): Classification quality metric measuring the capability of each algorithm in retaining or ameliorating the classification results of kNN through projection to the target space. It is defined as: Correctly Classified Instances in R^k/Correctly Classified Instances in R^n. Values close to or higher than 1 signify acceptable behavior.

3. Relative Clustering Disability Degradation (RCDD): Clustering quality metric that measures the capability of each algorithm in discovering clusters of K-Means through projection. It is defined as: Incorrectly Clustered Instances in R^k/ Incorrectly Clustered Instances in R^n.Values close to or lower than 1 signify acceptable behavior.

In order to test the statistical significance of our results, we have analyzed their variance. ANalysis Of VAriance (ANOVA) is a mathematical process that tests the statistical significance of the differences between the mean values of two or more populations of observations. In order to ensure credible results, ANOVA requires: 1) mutually independent population values, 2) that pairwise exhibited variances of populations are the same and 3) that observations should be normally distributed within populations. The first requirement is essential. On the other hand, any violation of the second or the third will not be a source of significant problems. This is proved by [15], where it is shown that the F statistic[5] is quite robust against violations of these assumptions. Based on the aforementioned observations and our experimental setup, we use ANOVA to evaluate the observed stress values. Classification and clustering experiments cannot be validated by ANOVA, due to the fact that the first requirements is not satisfied. The validity of the classification experiments has been verified with 10-fold cross validation.

The datasets used for our experiments are presented in Table 3. To the best of our knowledge, the datasets employed in this evaluation are among the largest ones in comparison to experiments found in relevant research work. This verifies our claim for robust behavior of FEDRA. Each dataset was evaluated for 5 different values of the target dimensionality (thus in the relevant graphs we encounter 5 points for each algorithm), which are defined as a fraction of the initial dimensionality of the dataset. Each time the projection dimensionality is increased by 2% of the initial dimensions (starting from 2%)[6]. Each resulting value is the mean of 20 executions. In all experiments we used the Euclidean distance metric.

We compare our algorithm against PCA, SVD, Random Projection, FastMap and LMDS. In the case of Random Projection the random matrix is generated using the distributions provided by Achlioptas in [3]. The first proposed distribution is depicted in the various graphs as RP(1/2) and the second as RP(1/6). On the other hand, LMDS selects either randomly (LMDS RN) or heuristically (LMDS MM) $2k$ landmarks for its initialization.

All experiments have been carried out on a commodity 2.4GHz Pentium IV machine with 1.28GB of RAM. In the first two sets of experiments WEKA [2] is used for both classification and clustering purposes, while in the last set, where memory requirements are extremely high, we employed an implementation of the Gmeans [10] algorithm.

7.1 FEDRA Evaluation. In the first set of experiments we used the Wine dataset to assess FEDRA's (and its variations') performance with regards to Stress, RCAM and RCDD. Our intention was to measure the statistical significance of the differences between the metrics values obtained by the base FEDRA algorithm and its heuristics.

As one can see in Figures 6(a), 7(a), neither the DMH

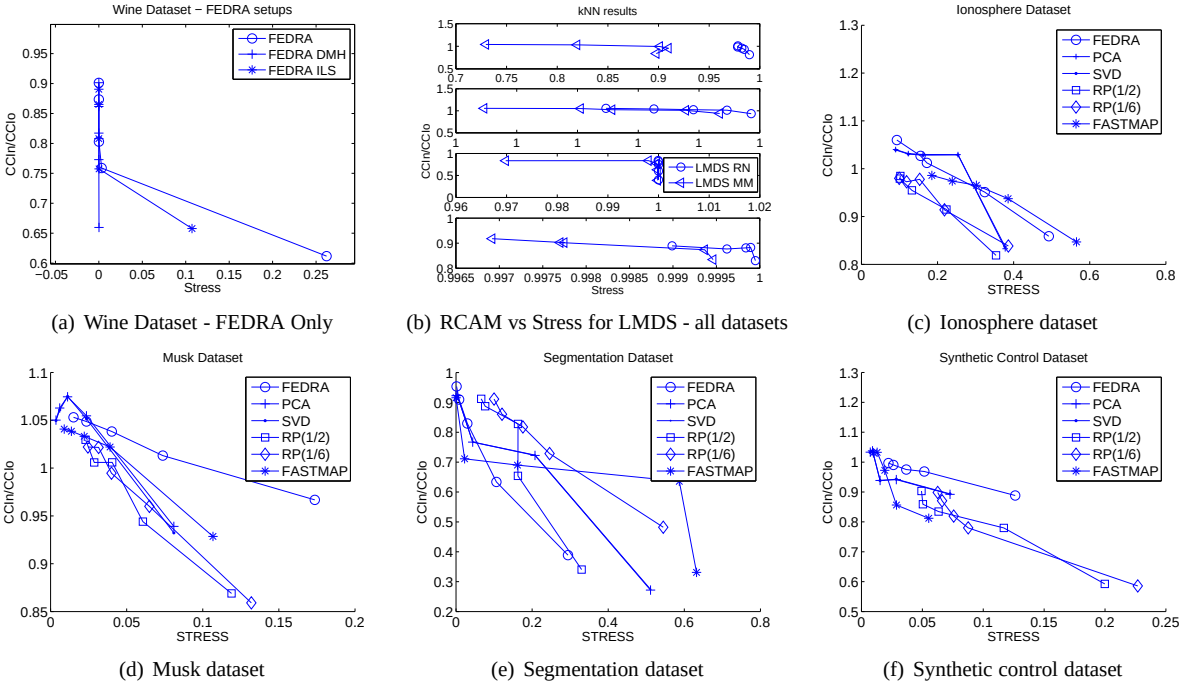
[4] We use the terms maintenance and preservation interchangeably.

[5] F statistic is the ratio of two s squares (i.e. estimates of a population variance, based on the information in two or more random samples). When employed in the procedure entitled ANOVA, the obtained value of F provides a test for the statistical significance of the observed differences among the means of two or more random samples.

[6] If 2% is smaller than 1 then the projection dimensionality is set to 1 and it is increased by one on iteration basis.

Table 3: Datasets used in the evaluation.

Dataset	Objects	Dimensions	Classes	Description
Wine	178	13	3	Wine chemical observations
Segmentation	2100	19	7	Image segmentation data
Musk	476	617	2	Molecules descriptions
Ionosphere	351	34	2	Ionosphere observations
Synthetic Control	600	60	6	Synthetic Control dataset
Connect-4	67577	42	3	A set of connect-4 games
CovType	581012	54	7	Forest cover type data



(a) Wine Dataset - FEDRA Only (b) RCAM vs Stress for LMDS - all datasets (c) Ionosphere dataset

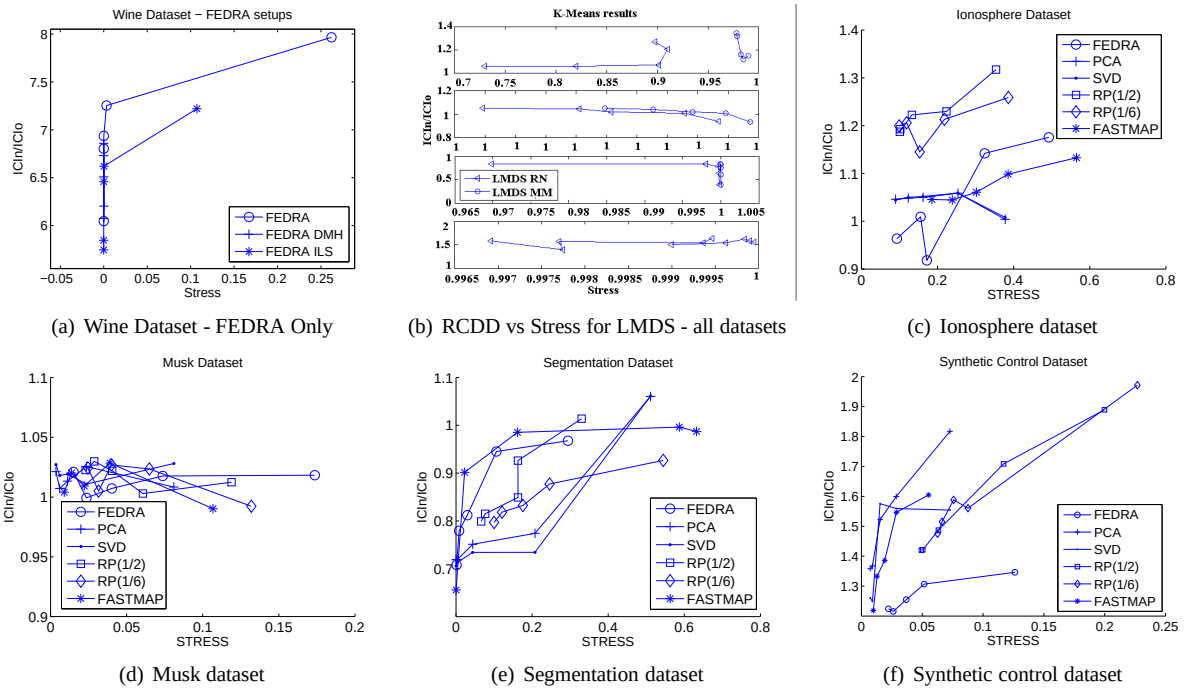
(d) Musk dataset (e) Segmentation dataset (f) Synthetic control dataset

Figure 6: Relative classification ability maintenance (RCAM) depicted as a function of stress.

implementation nor the ILS selection of landmarks ameliorated the results significantly in most cases. In addition, variance analysis of the obtained results suggested that with a confidence level of 95% our hypothesis, equality between results mean values, is maintained in 4 out of 5 projection loops. This means that regardless of the employed FEDRA's variation, stress remains the same within a confidence level of 95%. We come to the same conclusion regarding the respective experiments for RCDD and RCAM. Consequently, based on these results, we employ only the base FEDRA algorithm in the rest of our experiments.

7.2 Classification Ability Preservation. In the second set of experiments we evaluated all approaches against four datasets and depicted their RCAM values versus the exhibited stress. The employed representation depicts algorithms with low stress and high RCAM values in the upper left part of the graph. Moreover, although not explicitly shown in the graphs, higher stress values correspond to lower projection dimensionality. In other words, the rightmost point of a line in the charts corresponds to $k = 10\%n$ and the leftmost point to $k = 2\%n$. Consequently, stress is minimized as the projection dimensionality increases towards 10% of the original one. According to [13] we would expect an amelioration of the classification results of kNN because of dimensionality reduction. However, due to the fact that the latter research has made rather broad assumptions, we expect a general tendency of reaching 100% of preservation or a small fraction of improvement.

The first remarkable result was the very high stress value generated by LMDS. Despite this, LMDS (Figures 6(b) and 7(b)) produced fair results with respect to the RCAM measure, with high stability, but always lower than the maximum RCAM values achieved by the other approaches. FEDRA on the other hand clearly outperforms Random Projection in all experiments, while achieving better classification ability



(a) Wine Dataset - FEDRA Only (b) RCDD vs Stress for LMDS - all datasets (c) Ionosphere dataset

(d) Musk dataset (e) Segmentation dataset (f) Synthetic control dataset

Figure 7: Relative clustering disability degradation (RCDD) depicted as a function of stress.

Table 4: Experiments with large datasets.

Algorithm	RFM (CovType)	RCAM (Connect-4)	RCDD (Connect-4)
FEDRA	100%	98%	82%
RP(1/2)	98%	99%	81%
RP(1/6)	95%	94%	82%

maintenance than most of the other algorithms. Specifically, in Figure 6(d) (musk dataset), FEDRA clearly outperforms all other algorithms in 3 out of 5 projections, while it performs slightly worse than SVD and PCA for $k = 8\%n$ and $k = 10\%n$. Exactly the same behavior, but with significantly higher RCAM values with respect to the rest of the algorithms, appears in Figure 6(f). Another notable fact is that the stress exhibited by FEDRA is always significantly lower than the one exhibited by Random Projection. In Figure 6(c), FEDRA is shown to outperform all other approaches in the final projection, but it is being supplanted by PCA, SVD and FastMap in 2 out 5 projections. Similar conclusions are drawn by the results in Figure 6(e).

7.3 Clustering Disability Degradation. In this set of experiments we evaluate all approaches with the previously presented datasets and plot their RCDD values as a function of stress. The results highlight algorithms with low stress and low RCDD values in the lower left part of the graph. This metric's best values are those closer to 1, conveying that the clustering quality is either the same or better than the one exhibited by the clustering algorithm in the original space.

As one can see in Figure 7(b), LMDS exhibits low stress while its clustering quality degrades in almost three out of four cases. Only for the segmentation dataset we notice an improvement of the clustering quality, where for a particular projection we obtain a quality of almost 100%.

FEDRA successfully addresses the challenges posed by this new setup as it achieves the best quality update for the synthetic control and the ionosphere datasets (Figures 7(c), 7(f)). In addition, as far as the former is concerned, FEDRA constantly exhibits very favorable values of both stress and improvement of clustering ability. The experiments with the musk dataset (Figure 7(d)) indicated that all algorithms exhibit similar behavior, with FEDRA providing slightly better and more stable results with respect to the RCDD measure. Finally, in Figure 7(e), although FEDRA is initially outperformed by SVD and PCA, it finally achieves high quality results in both stress and RCDD measures. It is worth mentioning in this case FastMap's excellence in the fifth projection, where it outperforms even PCA.

7.4 Memory Intensive Datasets. Eventually we conducted experiments with two very large datasets and performed one single projection in a space of dimensionality equal to the 10% of original one. Here we excluded algo-

rithms such as PCA, SVD, and FastMap due to their excessive memory requirements. Moreover we decided to exclude LMDS due to the excessive stress values produced in the previous experiments. We only measured classification and clustering quality in both datasets as the computation of stress was also infeasible, due to the same reasons.

Table 4 shows experimental results for two datasets: CovType and Connect-4. In the CovType dataset, due to memory limitations posed by the implementation of WEKA, we used an implementation of Gmeans. As clustering validity measure we use the fraction of the value of F-measure [7] in the projection space over the one in the original space and denote as Relative F-Measure (RFM). Ideally, a flawless projection would achieve a value of 100%. FEDRA achieved 100% clustering quality preservation, thus outperforming both Random Projection setups. In the Connect-4 dataset, we used the RCAM and RCDD values to compare the algorithms. FEDRA exhibits high quality results in both metrics and slightly outperforms Random Projections, with the exception of RP(1/2) and RCAM. Moreover, the clustering quality is significantly improved in all cases illustrating thus the merits of dimensionality reduction in this context.

8 Conclusions

In this paper, we proposed FEDRA, a fast and efficient dimensionality reduction algorithm suitable for hard dimensionality reduction problems, where existing algorithms cannot be applied. FEDRA achieves quality of results comparable to the most prominent DR algorithms, while being faster and consuming less space in memory. In addition, FEDRA is able to employ any Minkowski distance ($p > 1$) and not only the Euclidean distance. We theoretically quantitated the quality of the projection and provided bounds in the error introduced due to the projection. Moreover, we proposed two extensions of the basic algorithm, which can increase FEDRA's quality of results, for applications that can tolerate higher processing costs. Additionally we introduced a simple offline algorithm that assesses the quality of the projection and triggers the re-initialization of FEDRA if necessary. Through extensive experiments on real world datasets we demonstrated FEDRA's quality, reflected in the high quality clustering and classification results achieved. In our future work, we will focus on applying FEDRA on text collections and study its performance in such ultra-high dimensional sparse representation spaces.

References

[1] UCI Machine Learning Repository.

[2] Waikato environment for knowledge analysis (WEKA), University of Waikato, New Zealand.

[3] D. Achlioptas. Database-friendly random projections. In *Proceedings of PODS*, pages 274-281, 2001.

[4] D. Bassu and C. Behrens. Distributed LSI: Scalable concept-based information retrieval with high semantic resolution. In *Proceedings of the 3rd SDM Text Mining Workshop*, 2003.

[5] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *Proceedings of ICDT*, pages 217-235, 1999.

[6] M. Carreira. A review of dimension reduction techniques. Technical report, University of Stanford, 1997.

[7] S. Chakrabarti. *Mining the Web - Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, 2003.

[8] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical report, ICSI, tr-99-006, March 1999.

[9] V. de Silva and J. B. Tenenbaum. Sparse multidimensional scaling using landmark points. Technical report, Stanford Mathematics, 2004.

[10] I. S. Dhillon, Y. Guan, and J. Kogan. Iterative clustering of high dimensional text data augmented by local search. In *Proceedings of ICDM*, pages 131-138, 2002.

[11] D.L.Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Conference Mathematical Challenges of the 21st Century*, 2000.

[12] C. Faloutsos and D. Lin. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of SIGMOD*, pages 163-174, 1995.

[13] A. Hinneburg, C. Aggarwal, and D. Keim. What is nearest neighbor in high dimensional spaces? In *Proceedings of VLDB*, pages 506-515, 2000.

[14] W. Johnson and J. Lindenstrauss. Extension of lipshitz mapping into hilbert space. *Contemp. Math*, 26:189-206, 1984.

[15] H. Lindman. Analysis of variance in complex experimental designs. *SIAM Review*, 18:134-137, 1976.

[16] P. Magdalinos, C. Doulkeridis, and M. Vazirgiannis. K-landmarks: Distributed dimensionality reduction for clustering quality maintenance. In *Proceedings of PKDD*, pages 322-334, 2006.

[17] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[18] C. H. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *PODS*, pages 59-168, 1998.

[19] J. S. R.C.T.Lee and H.Blum. A triangulation method for the sequential method of points from n-space to two-space. *IEEE Transaction on Computers*, 26:288-292, 1977.

[20] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science Magazine*, 290:2323-2326, December 2000.

[21] L. Smith. A tutorial on principle components analysis. University of Otago, February 2002.

[22] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science Magazine*, 290:2319-2323, December 2000.

[23] W. Togerson. *Theory and methods of scaling*. Wiley, 1958.

[24] J. Tougas. A divide-and-conquer approach to the singular value decomposition. Technical report, Dalhousie University, April 2004.