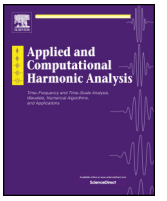
Appl. Comput. Harmon. Anal. 54 (2021) 20–43

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

www.elsevier.com/locate/acha



Geometric component analysis and its applications to data analysis

Amit Bermanis [a], Moshe Salhov [b], Amir Averbuch [b,*]

[a] *Department of Mathematics, University of Toronto, Canada*
[b] *School of Computer Science, Tel Aviv University, Israel*

ARTICLE INFO

Article history:
Received 10 July 2019
Received in revised form 17 December 2020
Accepted 26 February 2021
Available online 9 March 2021
Communicated by Radu Balan

Keywords:
Linear and non-linear dimensionality reduction
Incomplete pivoted QR
Diffusion maps
Dictionary construction
Landmark data points

ABSTRACT

Dimensionality reduction methods are designed to overcome the 'curse of dimensionality' phenomenon that makes the analysis of high dimensional big data difficult. Many of these methods are based on principal component analysis which is statistically driven and do not directly address the geometry of the data. Thus, machine learning tasks, such as classification and anomaly detection, may not benefit from a PCA-based methodology.

This work provides a dictionary-based framework for geometrically driven data analysis, for both linear and non-linear (diffusion geometries), that includes dimensionality reduction, out-of-sample extension and anomaly detection. This paper proposes the Geometric Component Analysis (GCA) methodology for dimensionality reduction of linear and non-linear data. The main algorithm greedily picks multidimensional data points that form linear subspaces in the ambient space that contain as much information as possible from the original data. For non-linear data, this greedy approach to the "diffusion kernel" is commonly used in diffusion geometry. The GCA-based diffusion maps appear to be a direct application of a greedy algorithm to the kernel matrix constructed in diffusion maps. The algorithm greedily selects data points from the data according to their distances from the subspace spanned by the previously selected data points. When the distance of all the remaining data points is smaller than a prespecified threshold, the algorithm stops.

The extracted geometry of the data is preserved up to a user-defined distortion rate. In addition, a subset of landmark data points, known as dictionary, is identified by the presented algorithm for dimensionality reduction that is geometric-based. The performance of the method is demonstrated and evaluated on both synthetic and real-world data sets. It achieves good results for unsupervised learning tasks. The proposed algorithm is attractive for its simplicity, low computational complexity and tractability.

* Corresponding author.
E-mail address: amir@math.tau.ac.il (A. Averbuch).

https://doi.org/10.1016/j.acha.2021.02.005
1063-5203/

1. Introduction

Analysis of large data silos is a fundamental task in many scientific and industrial fields, whose goal is to infer significant information from a collection of observations (measurements). The extracted information might illuminate the underlying phenomenon that generated the observed data. High-dimensional big data is of special interest since its analysis becomes harder as its dimension grows.

The challenge of processing high dimensional big data has been known for many decades [31]. The increase in dimensionality impacts many aspects of data analysis such as computational complexity, memory usage, reduction in feature predictability [29] and most importantly as the dimension grows, the proximity or the distance metrics between high dimensional data points become meaningless.

While dimensionality reduction methods are differ by the geometries they assign to the analyzed data, most of them use as a last step statistical techniques for finding significant components which compactly represent the assigned low-dimensional geometry. These techniques are based on Principal Component Analysis (PCA) [28] for the linear case and on kernel PCA for the non-linear case, when the geometry is encapsulated in a Gram matrix, referred to as *Kernel*. In both cases, PCA embeds the data in a low-dimensional subspace, whose coordinates are the directions of the high variances of the data, also known as principal components. Finally, in both linear and non-linear cases, PCA is implemented by the application of the Singular Value Decomposition (SVD) [23] to either the data matrix (in the linear case) or to the kernel matrix (in the non-linear case).

Although SVD-based approximations are optimal in the mean square error sense, which is a global measure, it is not necessarily optimal for the evaluation of local error measures. Such an error measure, which is referred to as *distortion*, is introduced in this work. As we theoretically show and experimentally demonstrate, SVD enables to provide a certain distortion, but the resulted representation may be redundant in the sense of dimensionality and, consequently, its storage and computational complexities may be high. Lastly, while the detected low-dimensional subspace provided by SVD is a mixture of the given vectors, the subspace provided by our method is spanned by a subset of the given set of vectors. Such landmarks of multidimensional data points might be very important for data analysis tasks.

This work presents a geometrically based deterministic method for linear and non-linear dimensionality reduction. In addition, we also show how to incorporate the developed method with diffusion maps (DM) [15], which is a non-linear framework for dimensionality reduction. The presented dictionary-based algorithm is designed to approximately preserve a low-dimensional Euclidean geometry of high-dimensional data, where the dictionary elements are chosen from the analyzed dataset. The dimensionality reduction algorithm identifies an embedded subspace in the ambient space, which is spanned by the dictionary, on which the orthogonal projection of the data provides a user-defined distortion of the original high-dimensional data. In that sense, our method is geometrically driven as opposed to PCA which is energy covariance driven. Moreover, it preserves global patterns of the data and trades local geometry for a low-dimensional representation. Thus, while intra-cluster geometries may be significantly affected, inter-clusters geometry is preserved. The storage and the computational complexities of the presented algorithm are lower (equal, in the worst case) to those of the SVD.

Additionally to dimensionality reduction, we present two strongly related schemes for out-of-sample extension and anomaly detection, which are naturally stem from the proposed dimensionality reduction method. Thus, the dimensionality reduction phase, followed by an out-of-sample extension and by an anomaly detection, constitutes a complete framework for unsupervised learning.

The rest of this paper is organized as follows: Work related to dimensionality reduction, greedy algorithms, dictionary construction and their relations to diffusion geometry is described in section 2. The notion of distortion, as well as a rigorously-proven algorithm for detection of low-dimensional distorted data embedding, is presented in Section 3. The connection between the proposed algorithm and the incomplete pivoted QR decomposition of a matrix is discussed in Section 4. The distortion provided by SVD is also

discussed in this section. Section 5 describes the diffusion maps, which is a non-linear method for dimensionality reduction, and shows how to implement the suggested algorithm in order to achieve a geometric rather than spectral diffusion maps. Out-of-sample-extension and anomaly detection for both linear and diffusion-maps based dimensionality reductions are described in Section 6. Experimental results for synthetic and real data analyses are shown in Section 7. Conclusions and a discussion on future works appear in Section 8.

2. Related work

The paper contains several technologies such as computational distortion bound, dimensionality reduction, greedy algorithms, dictionary construction and diffusion forecasting that are combined by the GCA framework. This section briefly describes the related work to the above technologies.

Johnson-Lindenstrauss (JL) Lemma [32] constitutes a basis for many random projection based dimensionality reduction methods [33,30,41,1,2,14,4,43] to name some. JL Lemma guarantees a relative distortion bound achieved with high probability. From data analysis perspective, an absolute bound may be more useful when the data is comprised of dense clusters separated by sparse regions. In this scenario, the absolute bound guarantees the existence of embedded space such that intra-cluster distances may be distorted but inter-cluster distances are preserved so that the global high-dimensional geometry is preserved in the embedded subspace. The relative bound, on the other hand, may produce a low-dimensional space in which clusters are becoming too close to each other.

Another significant branch of dimensionality reduction methods, which is strongly related to the presented work, deals with the column subset selection problem [21,18,11,10,9,35] to name a few. Interpolative decomposition (ID) of a matrix was introduced first for operator compression [13]. It is designed to approximate spectrally linear integral operators. A randomized version of ID is presented in [38]. The class of randomized matrix decomposition algorithms is out of the scope of this paper. CUR decomposition [12,36,44,19] of a given matrix A generalizes the ID in the sense that both subsets of columns and rows of the original matrix are selected to form the matrices C and R, respectively, such that $A \approx CUR$ for a low rank matrix U. In data analysis terms, this decomposition enables us to sample significant data points (rows) as well as significant features (columns). Similar to ID and unlike this work, CUR decompositions are designed to spectrally approximate the original matrix A.

Multidimensional Scaling (MDS) is presented in [27]. Given a matrix of pairwise distances between data points, it finds a low dimensional embedding of these data points in a Euclidean space, where the distances are preserved up to some global distortion of the pairwise distances. In a similar approach, Torgeson's multidimensional scaling (also in [27]) minimizes a global distortion function that is measured for pairwise inner products.

A clustering framework within sparse modeling and dictionary learning by optimizing a set of dictionaries, one for each cluster, are introduced in [39]. The data is a union of learned low dimensional subspaces. The data points associated to subspaces spanned by just a few atoms of the same learned dictionary are clustered together. Methods for determining the proper representation of data sets by means of reduced dimensionality subspaces, which are adaptive to both the characteristics of the signals and the processing task, are presented in [42]. These representations are based on the principle that our observations can be described by a sparse subset of atoms taken from a redundant dictionary. A large-scale matrix factorization problem, which consists of learning the basis set in order to adapt it to specific data, is described in [37]. It includes a dictionary learning of non-negative matrix factorization and sparse principal component analysis using online optimization algorithm that are based on stochastic approximations making it suitable for a wide range of learning problems. These dictionary constructions, which are sample examples from many others constructions, are different from what is proposed in this paper.

The greedy algorithms in [8,16] provide theoretical foundations to the problem of finding a good n-dimensional space which can be used to approximate the whole space including the best possible error achieved for such an approximation.

Diffusion forecasting is solved by approximating the solution of the Fokker–Planck equation with a discrete representation of the shift operator on a set of basis functions generated via the diffusion maps algorithm. While the choice of these basis functions is provably optimal under appropriate conditions, computing these basis functions is expensive since it requires an eigen-decomposition of an $N \times N$ diffusion matrix, where N denotes the data size. To overcome this computational bottleneck, a new set of basis functions constructed by orthonormalizing selected columns of the diffusion matrix and its leading eigenvectors is proposed as described in [26]. This computation can be carried out efficiently via the unpivoted Householder QR factorization. There is a minor resemblance to our use of incomplete pivoted QR decomposition and what appeared in [26].

3. Geometric component analysis (GCA)

GCA refers to the analysis of a given set of vectors in an Euclidean space which, geometrically driven, provides a subspace that well represents the given set. This subspace is spanned by a subset of representative elements from the given set. In this section, we provide a rigorous definition for that notion, as well as the GCA algorithm and a proof of its correctness. Stability of the algorithm to additive noise is briefly discussed in Appendix A.

Formally, given a finite set $\mathcal{A} \subset \mathbb{R}^m$, the objective is to provide a subset $\mathcal{D} \subset \mathcal{A}$, that spans a low-dimensional subspace $\mathcal{L} := span(\mathcal{D})$ and a function $\mathbf{f}_\mu : \mathbb{R}^m \to \mathcal{L}$, such that for every $\mathbf{u}, \mathbf{v} \in \mathcal{A}$

$$\left| \|\mathbf{u} - \mathbf{v}\| - \|\mathbf{f}_\mu(\mathbf{u}) - \mathbf{f}_\mu(\mathbf{v})\| \right| \leq \mu, \tag{3.1}$$

for a predefined $\mu > 0$, where $\|\cdot\|$ is the standard Euclidean norm in $\mathbb{R}^m$. Thus, the embedding $\mathbf{f}_\mu$ preserves the geometry of $\mathcal{A}$ up to a distortion rate μ. We refer to $\mathcal{D}$, $\mathbf{f}_\mu$ and $\mathcal{L}$ as *μ-dictionary*, *μ-embedding* (or *μ-distortion*) and *μ-embedding space* of $\mathcal{A}$, respectively.

In order to provide a μ-embedding of $\mathcal{A}$, Algorithm 1 incrementally selects dictionary elements from $\mathcal{A}$. The algorithm terminates when all the vectors of $\mathcal{A}$ are well approximated by that subset, namely - when for any $\mathbf{u} \in \mathcal{A}$ it satisfies

$$\|\mathbf{u} - \mathbf{P}_\mathcal{L}(\mathbf{u})\| \leq \frac{\mu}{2}, \tag{3.2}$$

where $\mathbf{P}_\mathcal{L} : \mathbb{R}^m \to \mathcal{L}$ is the orthogonal projection onto $\mathcal{L}$. Consequently, the μ-embedding is defined as $\mathbf{f}_\mu := \mathbf{P}_\mathcal{L}$.

Algorithm 1 provides a linearly independent dictionary $\mathcal{D}$, as Proposition 3.1 shows.

Proposition 3.1. *The dictionary $\mathcal{D}$ produced by Algorithm 1 is linearly independent.*

Proof. Suppose that $\mathcal{D} = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\} \subset \mathbb{R}^m$ is linearly dependent, where indexation is compatible with Algorithm 1. Then, there are real scalars $a_1, \ldots, a_N$ not all zeros, for which $\sum_{i=1}^N a_i \mathbf{v}_i = \mathbf{0}$. Let J be the largest index for which $a_J \neq 0$, then $\mathbf{v}_J = -\frac{1}{a_J} \sum_{i=1}^{J-1} a_i \mathbf{v}_i$. Thus, since $L_{J-1} = span\{\mathbf{v}_1, \ldots, \mathbf{v}_{J-1}\}$, in the J-th iteration of Algorithm 1, we get $\delta = \|\mathbf{v}_J - \mathbf{P}_{\mathcal{L}_{J-1}}(\mathbf{v}_J)\| = 0$, which doesn't satisfy the 'if' statement in Algorithm 1. □

Obviously, for any $\mathbf{u} \in \mathcal{D}$, $\mathbf{f}_\mu(\mathbf{u}) = \mathbf{u}$. Proposition 3.2 shows that $\mathbf{f}_\mu$ produced by Algorithm 1 (i.e. the orthogonal projection $\mathbf{P}_\mathcal{L}$), is a μ-embedding of $\mathcal{A}$.

Algorithm 1: GCA - Geometrically-driven dimensionality reduction.

Input : Set $\mathcal{A} \subset \mathbb{R}^m$ of size n, and a nonnegative distortion parameter μ.
Output: μ-embedding and μ-dictionary of $\mathcal{A}$ denoted by $\mathbf{f}_\mu$ and $\mathcal{D}$.

1 Initialization: set $k = 0$, $\mathcal{D}_k = \emptyset$, $\mathcal{L}_k = \{\mathbf{0}\}$, $\delta = \infty$
2 **while** $\delta > \frac{\mu}{2}$ **do**
3 $\quad \mathbf{v}_{k+1} = \arg\max_{\mathbf{u} \in \mathcal{A} \setminus \mathcal{D}_k} \|\mathbf{u} - \mathbf{P}_{\mathcal{L}_k}(\mathbf{u})\|$ ($\mathcal{A}\setminus\mathcal{D}_k$ is the set of all elements from $\mathcal{A}$ which are not in $\mathcal{D}_k$)
4 $\quad \delta = \|\mathbf{v}_{k+1} - \mathbf{P}_{\mathcal{L}_k}(\mathbf{v}_{k+1})\|$
5 $\quad$ **if** $\delta > \frac{\mu}{2}$ **then**
6 $\quad\quad \mathcal{D}_{k+1} = \mathcal{D}_k \cup \{\mathbf{v}_{k+1}\}$
7 $\quad\quad \mathcal{L}_{k+1} = span(\mathcal{D}_{k+1})$
8 $\quad\quad$ set $k = k + 1$
9 $\quad$ **else**
10 $\quad\quad$ break
11 $\quad$ **end**
12 **end**
13 set $\mathcal{D} = \mathcal{D}_k$ and $\mathcal{L} = \mathcal{L}_k$
14 Define $\mathbf{f}_\mu$ to be the orthogonal projection onto $\mathcal{L}$, i.e. $\mathbf{f}_\mu := \mathbf{P}_{\mathcal{L}}$

Proposition 3.2. *For every* $\mathbf{u}, \mathbf{v} \in \mathcal{A}$, $|\|\mathbf{u} - \mathbf{v}\| - \|\mathbf{f}_\mu(\mathbf{u}) - \mathbf{f}_\mu(\mathbf{v})\|| \leq \mu$.

Proof. According to the condition of the 'while' loop in Algorithm 1, for any $\mathbf{u} \in \mathcal{A}$, Eq. (3.2) is satisfied. Therefore, for every $\mathbf{u}, \mathbf{v} \in \mathcal{A}$ we have

$$\begin{aligned} |\|\mathbf{u} - \mathbf{v}\| - \|\mathbf{f}_\mu(\mathbf{u}) - \mathbf{f}_\mu(\mathbf{v})\|| &= |\|\mathbf{u} - \mathbf{v}\| - \|\mathbf{P}_{\mathcal{L}}(\mathbf{u}) - \mathbf{P}_{\mathcal{L}}(\mathbf{v})\|| \\ &\leq \|\mathbf{u} - \mathbf{P}_{\mathcal{L}}(\mathbf{u}) + \mathbf{P}_{\mathcal{L}}(\mathbf{v}) - \mathbf{v}\| \\ &\leq \|\mathbf{u} - \mathbf{P}_{\mathcal{L}}(\mathbf{u})\| + \|\mathbf{v} - \mathbf{P}_{\mathcal{L}}(\mathbf{v})\| \\ &\leq \mu. \quad \square \end{aligned}$$

Suppose that the μ-dictionary $\mathcal{D}$, produced by Algorithm 1, contains p elements. Then, since p is also the dimension of the μ-embedding space $\mathcal{L}$, it is bounded above by m. Obviously, there is an interplay between the dimension p and the distortion rate μ: the smaller the distortion rate the higher the dimension, and vice versa. In other words, p is a decreasing (unknown) function of μ.

The computational complexity of Algorithm 1 is $\mathcal{O}(mnp)$: The 'while' loop is repeated p times. Step 3 is the most costly, as in the k-th iteration, the projection $\mathbf{P}_{\mathcal{L}_k}$ has to be recomputed according to the dictionary update (steps 6 and 7) at the previous iteration and due to the fact that the required vector provides the maximum norm among $n - k$ vectors. Since $\mathcal{L}_k = \mathcal{L}_{k-1} \oplus span\{\mathbf{v}_k\}$, the projection $\mathbf{P}_{\mathcal{L}_k}$ is merely $\mathbf{P}_{\mathcal{L}_k} = \mathbf{P}_{\mathcal{L}_{k-1}} + \mathbf{P}_{\{\mathbf{v}_k - \mathbf{P}_{\mathcal{L}_{k-1}}(\mathbf{v}_k)\}}$, where $\mathbf{P}_{\{\mathbf{v}_k - \mathbf{P}_{\mathcal{L}_{k-1}}(\mathbf{v}_k)\}}$ is the orthogonal projection onto the subspace spanned by the vector $\mathbf{v}_k - \mathbf{P}_{\mathcal{L}_{k-1}}(\mathbf{v}_k)$. Therefore, since $\mathbf{P}_{\mathcal{L}_{k-1}}(\mathbf{u})$ was already computed for any $\mathbf{u} \in \mathcal{A}\setminus\mathcal{D}_{k-1}$ in step 3 of the previous iteration, the actual computational complexity of step 3 in the k-th iteration is $\mathcal{O}((n-k)m)$.

As for the storage complexity, since at each iteration of Algorithm 1 only the coefficients of the projections are stored in complexity of $\mathcal{O}(n)$, which is accumulated to an overall storage complexity of $\mathcal{O}(np)$, the actual storage complexity is dominated by the storage of the dataset $\mathcal{A}$ which is $\mathcal{O}(mn)$.

4. Link to low rank matrix decompositions: incomplete pivoted QR and PCA

Identification of a subspace, which well approximates a set of vectors in Euclidean space, is a central task in every low rank matrix decomposition. More specifically, if $A \in \mathbb{R}^{m \times n}$ is approximated by a multiplication of two (or more) matrices say, $C \in \mathbb{R}^{m \times p}$ and $R \in \mathbb{R}^{p \times n}$ for some p, then, the subspace spanned by the columns of C must be close in some sense to the subspace spanned by the columns of A. The same holds for the relation between the rows of A and the rows of R. The challenge is to find, when possible, such matrices with small p, for which the approximation is still reasonable.

There are many deterministic and randomized low-rank matrix approximation algorithms. Randomized algorithms are usually utilized to reduce the computational complexity by dimensionality reduction of the problem. In this setup, the original data matrix is first projected into a low-dimensional space, where it is deterministically decomposed. Then the high-dimensional matrix is decomposed accordingly. A survey of randomized algorithms for low-rank matrix decompositions is given in [25].

When the columns of C constitute a subset of the columns of A, the problem is referred to as *Columns Sampling* in the randomized case, and *Columns Selection* in the deterministic case. Examples for such matrix decompositions are the incomplete pivoted QR decomposition, which is discussed in Section 4.1, interpolative decomposition [13,38], which is based on QR decomposition, and CUR decomposition (see [11] and references therein). The CUR decomposition is very interesting in the context of data analysis since C and R are matrices whose columns and rows (respectively) are subsets of the columns and rows of the original matrix. This is associated with the choice of 'good' data points (rows) and 'good' features (columns). Not all low rank matrix decompositions are of the columns sampling/selection type. An example of such decomposition is the Singular Values Decomposition (SVD), which is discussed in Section 4.2.

Since most algorithms are based on either SVD or pivoted QR, in this section we relate the GCA to these two well known matrix decompositions, which are fundamentally differ from each other. As singular values and vectors of a matrix are influenced by statistical properties of the rows and the columns of the matrix, SVD is considered to be statistically driven. On the other hand, since QR decomposition is based on Gram-Schmidt process, applied to the columns of the matrix, it is considered to be geometrically driven. Moreover, since pivoting is done by choosing significant columns (according to a predefined criterion), it is strongly related to the columns selection setup. Both algorithms have incomplete low-rank versions. Both can be utilized to achieve a low-rate distortion of data that is stored as A's columns.

In section 4.1, the elements of the dataset $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ constitute the columns of the data matrix $A \in \mathbb{R}^{m \times n}$, i.e.

$$A = \begin{bmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_n \\ | & & | \end{bmatrix}.$$

4.1. Incomplete pivoted QR decomposition

In this section, we assume that the order of the A's columns is compatible with the selection order of Algorithm 1. In the context of matrix decomposition, such ordering is called *pivoting*.

QR decomposition [22] of an $m \times n$ rank-ρ matrix A is

$$A = Q_\rho R_\rho, \tag{4.1}$$

where R_ρ is a $\rho \times n$ upper triangular matrix, and Q_ρ is an $m \times \rho$ matrix, whose first k columns $\mathbf{q}_1, \ldots, \mathbf{q}_k$ constitute an orthonormal basis for the subspace spanned by the first k columns of A. Thus, if for every $k = 1, \ldots, \rho$, the subspace $\mathcal{L}_k \subset \mathbb{R}^m$ is defined by $\mathcal{L}_k := span\{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$, then

$$\mathcal{L}_k = span\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}, \ k = 1, \ldots, \rho. \tag{4.2}$$

The diagonal elements of R_ρ, denoted by r_{kk}, satisfy

$$|r_{kk}| = \|\mathbf{a}_k - \mathbf{P}_{\mathcal{L}_{k-1}}(\mathbf{a}_k)\|, \ k = 1, \ldots, \rho. \tag{4.3}$$

Such decomposition can be achieved by the application of Gram-Schmidt process to the columns of A.

When the columns of A are ordered compatibly with Algorithm 1, the diagonal of R_ρ is ordered decreasingly by their modulus, i.e. $|r_{11}| \geq |r_{22}| \geq, \ldots, \geq |r_{\rho\rho}|$.

If Q_ρ and R_ρ in Eq. (4.1) are replaced by the truncated versions Q_p and R_p ($p < \rho$), where Q_p is the leftmost $m \times p$ submatrix of Q_ρ, R_p is the upper $p \times n$ submatrix of R_ρ, and p is set to be the number of diagonal elements in R_ρ whose magnitudes are greater than $\mu/2$, i.e.

$$p := max_k \, |r_{kk}| > \frac{\mu}{2}, \tag{4.4}$$

then due to Eqs. (4.1)-(4.2), the operator $\mathbf{P}_{\mathcal{L}_p} : \mathbb{R}^m \to \mathcal{L}_p$ is defined by

$$\mathbf{P}_{\mathcal{L}_p}(\mathbf{v}) := Q_p Q_p^* \mathbf{v} \tag{4.5}$$

is the orthogonal projection onto $\mathcal{L}_p$, and consequently

$$\mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_k) = \mathbf{a}_k, \;\; k = 1, \ldots, p. \tag{4.6}$$

Moreover, following Eqs. (4.3)-(4.4)

$$\left\| \mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_k) - \mathbf{a}_k \right\| \leq \frac{\mu}{2}, \;\; k = p+1, \ldots, n. \tag{4.7}$$

The multiplication $Q_p R_p$ is called *incomplete pivoted QR decomposition*. It constitutes a rank-p approximation of A due to the following proposition:

Proposition 4.1. *Assume that p is defined in Eq. (4.4). Then,*[1]

$$\|A - Q_p R_p\|_\eta \leq \frac{\mu}{2}\sqrt{\rho - p}, \;\; \eta \in \{2, F\}.$$

Proof. Due to Eqs. (4.6)-(4.7), at least p columns from $A - Q_p R_p$ vanish and the ℓ_2 norms of the rest (at most) $\rho - p$ columns are bounded by $\mu/2$. This leads to $\|A - Q_p R_p\|_2 \leq \|A - Q_p R_p\|_F \leq \frac{\mu}{2}\sqrt{\rho - p}$. □

Due to the resemblance between the GCA described in Section 3 and the incomplete pivoted QR, there exist algorithms for the incomplete pivoted QR whose computational and storage costs are identical to those of the GCA, namely $\mathcal{O}(mnp)$ and $\mathcal{O}(mn)$, respectively.

4.2. PCA-based dimensionality reduction

A common practice to achieve dimensionality reduction is based on the application of PCA [28] to the (centered) $m \times n$ data matrix A. PCA is a statistical procedure that uses an orthogonal transformation to convert a dataset of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This method uses an SVD of the data matrix to detect a set of maximum variance orthogonal directions (singular vectors) in $\mathbb{R}^m$. Projection of the data onto the p most significant directions yields the best p-dimensional embedding of the data in the mean square error sense. The computational and storage complexities of SVD are $\mathcal{O}(\min\{m,n\} \cdot mn)$ and $\mathcal{O}(\max\{m,n\}^2)$, respectively.

Mathematically, suppose that the rank of A is ρ. Let $A = U\Sigma V^*$ be the (thin) SVD of A, where U and V are $m \times \rho$ and $n \times \rho$ matrices, respectively, whose columns are orthonormal, and Σ is a diagonal $\rho \times \rho$ matrix, whose diagonal elements $\sigma_1 \geq \ldots \geq \sigma_\rho > 0$ are ordered decreasingly. The columns of U and V are referred to as the left and right singular vectors of A, respectively, and the diagonal elements of Σ as its singular values. The p-SVD of A is the $m \times n$ matrix

[1] Frobenius norm of a matrix M whose entries are $M(i,j)$ is $\|M\|_F := \sqrt{\sum_{i,j} M(i,j)^2}$, its spectral norm is $\|M\|_2 := \max_{\|\mathbf{v}\|=1} \|M\mathbf{v}\|$, and the inequality $\|M\|_2 \leq \|M\|_F$ holds for any matrix M.

$$A_p := U_p \Sigma_p V_p^*, \tag{4.8}$$

where U_p is the $m \times p$ leftmost submatrix of U, Σ_p is the $p \times p$ is the upper leftmost submatrix of Σ and V_p is the $n \times p$ upper leftmost submatrix of V. For any $p \leq \rho$ we have

$$\|A - A_p\|_2 = \sigma_{p+1}, \tag{4.9}$$

which is the best rank-p matrix approximation of A in the spectral norm.

Let $\mathcal{L}_p$ be the subspace spanned by the columns of U_p, then the rank p orthogonal projection $\mathbf{P}_{\mathcal{L}_p} : \mathbb{R}^m \to \mathcal{L}_p$ is defined by

$$\mathbf{P}_{\mathcal{L}_p}(\mathbf{v}) := U_p U_p^* \mathbf{v}. \tag{4.10}$$

Lemma 4.2 quantifies the distortion rate of $\mathbf{P}_{\mathcal{L}_p}$ that is applied to the set $\mathcal{A} \subset \mathbb{R}^m$ with respect to the spectrum of the corresponding matrix A encapsulated in Σ.

Lemma 4.2. *The p-dimensional embedding $\mathbf{P}_{\mathcal{L}_p}$ is a $2\sigma_{p+1}$-distortion of $\mathcal{A}$.*

Proof. From the triangle inequality we have for any $i, j = 1, \ldots, n$

$$\|\mathbf{a}_i - \mathbf{a}_j\| \leq \|\mathbf{a}_i - \mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_i)\| + \|\mathbf{a}_j - \mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_j)\| + \|\mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_i) - \mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_j)\|.$$

Following Eqs. (4.8)-(4.10) we have $\|\mathbf{a} - \mathbf{P}_{\mathcal{L}_p}(\mathbf{a})\| < \sigma_{p+1},\ \mathbf{a} \in \mathcal{A}$. Therefore, since $\|\mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_i - \mathbf{a}_j)\| \leq \|\mathbf{a}_i - \mathbf{a}_j\|,\ i, j = 1, \ldots, n$, we get $\big|\|\mathbf{a}_i - \mathbf{a}_j\| - \|\mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_i) - \mathbf{P}_{\mathcal{L}_p}(\mathbf{a}_j)\|\big| \leq 2\sigma_{p+1}$. □

The computational and storage complexities of the p-SVD are $\mathcal{O}(pmn)$ and $\mathcal{O}(\max\{m, n\}^2)$, respectively. Thus, if the required distortion rate and the spectrum of A are known, then a $2\sigma_{p+1}$-distortion can be achieved with these complexities by using a p-SVD. However, the principal subspace $\mathcal{L}_p$, on which the data is projected, is a mixture of the entire columns set of A which, in terms of data analysis, may be less informative than a dictionary-based subspace, which is defined according to a chosen columns subset of A.

5. GCA-based nonlinear dimensionality reduction: diffusion maps

Although the GCA-based dimensionality reduction method is designed for the analysis of set of vectors in Euclidean space, this section presents its utilization in the Diffusion Maps (DM) [15], which is a non-linear spectral method for data analysis, based on exploration of a random walk process defined on the data. It is mainly utilized for clustering and manifold learning. The standard DM utilizes SVD for dimensionality reduction which as aforementioned is not directly related to the embedding's distortion.

In this section, we show how to replace the SVD-based by a GCA-based dimensionality reduction. The latter also produces a dictionary, namely a subset of the analyzed data, which is informative for the associated diffusion process. Lastly, utilization of the GCA-based method might reduce the computational complexity since the process is terminated when a required distortion by the diffusion geometry is achieved.

5.1. DM framework: overview

A diffusion geometry, assigned to a dataset, is based on the notion of similarities (or affinities) between high dimensional data points, with analysis of an associated random walk over the data. DM embeds the dataset into Euclidean space while preserving the diffusion geometry namely, the Euclidean distances in the

embedded space approximate the diffusion distances. We provide in section 5.1.1 the fundamentals of DM in the ambient space.

To incorporate the GCA into DM we have to know that details that are described in section 5.1.1.

5.1.1. Diffusion geometry

Let $\mathcal{X} = \{x_1, \ldots, x_n\}$ be a dataset and let $\mathbf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric point-wise positive kernel that defines a connected, undirected and weighted graph over $\mathcal{X}$. Then, a random walk over $\mathcal{X}$ is defined by the $n \times n$ row-stochastic transition probabilities matrix

$$P = D^{-1}K, \tag{5.1}$$

where K is an $n \times n$ matrix whose entries are $K(i,j) := \mathbf{k}(x_i, x_j),\ i,j = 1, \ldots, n$, and D is the $n \times n$ diagonal degrees matrix whose i-th element is $\mathbf{d}(i) := \sum_{j=1}^{n} \mathbf{k}(x_i, x_j),\ i = 1, \ldots, n$. The vector $\mathbf{d} \in \mathbb{R}^n$ is referred to as the *degrees vector* of the graph defined by $\mathbf{k}$.

The associated time-homogeneous random walk $X(t)$, is defined via the conditional probabilities on its state-space $\mathcal{X}$: assuming that the process starts at time $t = 0$, then for any time point $t \in \mathbb{N}$

$$\mathbb{P}(X(t) = x_j | X(0) = x_i) = P^t(i,j), \tag{5.2}$$

where $P^t(i,j)$ is the (i,j)-th entry of the t-th power of the matrix P. As long as the process is aperiodic, it has a unique stationary distribution $\hat{\mathbf{d}} \in \mathbb{R}^n$ which is the steady state of the process, i.e. $\hat{\mathbf{d}}(j) = \lim_{t\to\infty} P^t(i,j)$, regardless the initial state $X(0)$. This steady state is the probability distribution resulted from ℓ_1 normalization of the degrees vector $\mathbf{d}$, i.e.,

$$\hat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|_1} \in \mathbb{R}^n, \tag{5.3}$$

where $\|\mathbf{d}\|_1 := \sum_{i=1}^{n} \mathbf{d}(i)$. The *diffusion distances* at time t are defined by the metric $\mathbf{D}^{(t)} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

$$\begin{aligned} \mathbf{D}^{(t)}(x_i, x_j) &:= \left\| P^t(i,:) - P^t(j,:) \right\|_{\ell^2(\hat{\mathbf{d}}^{-1})} \\ &= \sqrt{\sum_{k=1}^{n} (P^t(i,k) - P^t(j,k))^2 / \hat{\mathbf{d}}(k)},\ i,j = 1, \ldots, n. \end{aligned} \tag{5.4}$$

By definition, $P^t(i,:)$, the i-th row of P^t, is the probability distribution over $\mathcal{X}$ after t time steps given that the initial state is $X(0) = x_i$. Therefore, the diffusion distance $\mathbf{D}^{(t)}(x_i, x_j)$ from Eq. (5.4) measures the difference between two propagations along t time steps: the first is originated in x_i and the second in x_j. Weighing the metric by the inverse of the steady state results in ascribing high weight for similar probabilities on rare states and vice versa. Thus, a family of diffusion geometries is defined by Eq. (5.4), each corresponds to a single time step t.

Due to the above interpretation, the diffusion distances are naturally utilized for multiscale clustering since they uncover the connectivity properties of the graph across time. In [5,15] it is proved that under some conditions, if $\mathcal{X}$ is sampled from a low intrinsic dimensional manifold then, as n tends to infinity, the defined random walk converges to a diffusion process over the manifold.

5.1.2. Spectral DM - low dimensional representation of the diffusion geometry

Spectral DM [15] represents the diffusion geometries defined by Eq. (5.4) by embedding the dataset $\mathcal{X}$ into Euclidean spaces, where the Diffusion geometries are preserved. The embedding is achieved via spectral decomposition of the transition probabilities matrix P.

Let $G^{(t)}$ be the $n \times n$ matrix defined by

$$G^{(t)} := \|\mathbf{d}\|_1^{1/2} D^{-1/2}(P^*)^t,\ t \in \mathbb{N}, \tag{5.5}$$

where P^* is the transpose of P. Then, due to Eqs. (5.1)-(5.4), the Euclidean n-dimensional geometry of the columns of

$$G^{(t)} = [\mathbf{g}_1{}^{(t)}, \ldots, \mathbf{g}_n{}^{(t)}] \tag{5.6}$$

is isomorphic to the diffusion geometry of the associated data points i.e.,

$$\mathbf{D}^{(t)}(x_i, x_j) = \left\| \mathbf{g}_i{}^{(t)} - \mathbf{g}_j{}^{(t)} \right\|,\ i, j = 1, \ldots, n. \tag{5.7}$$

Although embedding $x_i \mapsto \mathbf{g}_i$ of the dataset $\mathcal{X}$ in $\mathbb{R}^n$ preserves the diffusion geometry, it may be ineffective for large n. Therefore, a dimensionality reduction is required.

Since the transition probabilities matrix P is conjugated to the symmetric matrix $M := D^{-1/2} K D^{-1/2}$ via the relation $P = D^{-1/2} M D^{1/2}$, then P has a complete real eigen-system. Moreover, due to Gershgorin's circle theorem [23] and the fact that P is a row stochastic, all its eigenvalues lie in the interval $(-1, 1]$ (the exclusion of -1 is due to the assumption that the associated random walk is aperiodic). Let

$$M = USU^* \tag{5.8}$$

be the eigen-decomposition of M, where U is an orthogonal $n \times n$ matrix and S is a diagonal $n \times n$ matrix, whose diagonal elements s_i are ordered decreasingly due to their modulus

$$1 = s_1 > |s_2| \geq \ldots \geq |s_n|\,. \tag{5.9}$$

The first inequality is due to the assumption that the graph is connected. Thus, following Eq. (5.5)

$$G^{(t)} = \|\mathbf{d}\|_1^{1/2} M^t D^{-1/2} = \|\mathbf{d}\|_1^{1/2} U S^t U^* D^{-1/2}. \tag{5.10}$$

Therefore, due to the orthogonality of U and according to Eq. (5.7)

$$\mathbf{D}^{(t)}(x_i, x_j) = \|\mathbf{d}\|_1^{1/2} \left\| S^t U^* D^{-1/2}(\mathbf{e}_i - \mathbf{e}_j) \right\|,\ i, j = 1, \ldots, n, \tag{5.11}$$

where $\mathbf{e}_i$ denotes the i-th standard unit vector in $\mathbb{R}^n$. Thus, the diffusion map $\mathbf{\Psi}^{(t)} : \mathcal{X} \to \mathbb{R}^n$ at time t defined by

$$\mathbf{\Psi}^{(t)}(x_i) := \hat{\mathbf{d}}(i)^{-1/2} [s_1^t U(i,1), \ldots, s_n^t U(i,n)]^*,\ i = 1, \ldots, n, \tag{5.12}$$

satisfies $\mathbf{D}^{(t)}(x_i, x_j) = \left\| \mathbf{\Psi}^{(t)}(x_i) - \mathbf{\Psi}^{(t)}(x_j) \right\|$. In order to achieve a low-dimensional embedding, the diffusion map in Eq. (5.12) is projected onto its significant principal components according to the decay rate of the spectrum of M^t. Specifically, for a sufficiently small $|s_{p+1}|^t$, the p-dimensional embedding is provided by the first p coordinates, denoted by $\mathbf{\Psi}_p^{(t)} : \mathcal{X} \to \mathbb{R}^p$. Lemma 5.1 quantifies the distortion resulted by such a projection.

Lemma 5.1. *Let*

$$\gamma^{(t)} = \sqrt{2} c\, |s_{p+1}|^t\,,$$

where $c = \max_{i\in\{1,\ldots,n\}} \hat{\mathbf{d}}(i)^{-1/2}$. *Then, the* p*-dimensional embedding* $\boldsymbol{\Psi}_p^{(t)}$ *has a* $\gamma^{(t)}$*-distortion of the* n*-dimensional diffusion map* $\boldsymbol{\Psi}^{(t)}$ *from Eq. (5.12).*

Proof. Following Eq. (5.12) we get[2]

$$\begin{aligned}\left|\left\|\boldsymbol{\Psi}^{(t)}(x_i)-\boldsymbol{\Psi}^{(t)}(x_j)\right\|-\left\|\boldsymbol{\Psi}_p^{(t)}(x_i)-\boldsymbol{\Psi}_p^{(t)}(x_j)\right\|\right| &\le \left\|\boldsymbol{\Psi}^{(t)}(x_i)-\boldsymbol{\Psi}_p^{(t)}(x_i)\right\| \\ &\quad+\left\|\boldsymbol{\Psi}^{(t)}(x_j)-\boldsymbol{\Psi}_p^{(t)}(x_j)\right\| \\ &\le |s_{p+1}|^t\left(\hat{\mathbf{d}}(i)^{-1}+\hat{\mathbf{d}}(j)^{-1}\right)^{1/2} \\ &\le \sqrt{2}c\,|s_{p+1}|^t. \quad \square\end{aligned}$$

The distortion bound from Lemma 5.1 is referred to as the *analytic bound*. In some cases, upper bounds of the spectral properties of the utilized kernel can be estimated a-priori with no need for its explicit computation. The Gaussian kernel is just one example (see [6,7]). In such cases, a partial SVD of $G^{(t)}$ can be calculated to produce the relevant principal component according to the required distortion, in the computational and storage costs of $\mathcal{O}(n^2p)$ and $\mathcal{O}(n^2)$, respectively, where p is the required dimension as mentioned in Section 4.2. However, these upper bounds may be too loose and, as a result, the above mentioned complexities may be too high.

5.2. Geometric DM

In this section, we provide an algorithm for GCA-based DM. The advantages of this method over the spectral DM, which is detailed in Section 5.1.2, are two: 1. The computational process terminates when the required distortion is achieved, thus the computational complexity is $\mathcal{O}(n^2p)$, where p is the dimension of the resulted μ-embedding. 2. It provides a dictionary, i.e. a subset of significant data points.

Following the discussion in Section 5.1.2, the diffusion geometry of the dataset $\mathcal{X}$ (at time t) is identical to the Euclidean geometry of the columns of the matrix $G^{(t)}$ (see Eqs. (5.6)-(5.7)). Application of Algorithm 1 to the dataset $\mathcal{G}^{(t)} = \{\mathbf{g}_1{}^{(t)},\ldots,\mathbf{g}_n{}^{(t)}\}$ with an accuracy rate parameter μ provides a p-dimensional[3] μ-embedding $\mathbf{f}_\mu : \mathcal{G}^{(t)} \to \mathbb{R}^p$, such that

$$\left|\left\|\mathbf{f}_\mu(\mathbf{g}_i{}^{(t)})-\mathbf{f}_\mu(\mathbf{g}_j{}^{(t)})\right\|-\left\|\mathbf{g}_i{}^{(t)}-\mathbf{g}_j{}^{(t)}\right\|\right| \le \mu,\ i,j=1,\ldots,n, \tag{5.13}$$

where $p = p(\mu,t)$. Thus, by combining Eqs. (5.7) and (5.13), the p-dimensional embedding of the dataset $\mathcal{X}$, $\mathbf{f}_\mu{}^{(t)} : \mathcal{X} \to \mathbb{R}^p$ such that $\mathbf{f}_\mu{}^{(t)}(x_i) := \mathbf{f}_\mu(\mathbf{g}_i{}^{(t)}),\ i=1,\ldots,n$ yields

$$\left|\mathbf{D}^{(t)}(x_i,x_j)-\left\|\mathbf{f}_\mu{}^{(t)}(x_i)-\mathbf{f}_\mu{}^{(t)}(x_j)\right\|\right| \le \mu,\ i,j=1,\ldots,n.$$

We refer to the function $\mathbf{f}_\mu{}^{(t)}$ as *μ-DM of $\mathcal{X}$ at time t*.

Algorithm 2 concludes this section. Given a dataset, a kernel function, a distortion rate and a diffusion time-step, the algorithm produces a distorted DM of $\mathcal{X}$ for that time-step, as well as a dictionary subset.

[2] Here, for analysis purposes, $\boldsymbol{\Psi}_p^{(t)}$ is considered as a map $\mathbb{R}^n \mapsto \mathbb{R}^n$, that zeros out the last $n-p$ coordinates.

[3] As mentioned in Section 3, the dimension p is an unknown decreasing function of μ. Moreover, since the diffusion process converges (with respect to time) to a steady state, then as t grows the diffusion geometry becomes smoother and consequently the final dimension p is a decreasing function of both the distortion rate μ and the diffusion time t.

Algorithm 2: Geometric DM.

Input : Dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, nonnegative kernel function $\mathbf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, time-step $t \in \mathbb{N}$, and a nonnegative distortion parameter μ.

Output: A μ-DM of $\mathcal{X}$ at time t, ${\mathbf{f}_\mu}^{(t)}$, and a dictionary ${\mathcal{D}_\mu}^{(t)} \subset \mathcal{X}$.

form the kernel matrix $K \in \mathbb{R}^{n \times n}$, whose elements are $K(i,j) = \mathbf{k}(x_i, x_j)$, $i, j = 1, \ldots, n$

set the degrees vector $\mathbf{d} \in \mathbb{R}^n$, whose elements are $\mathbf{d}(i) = \sum_{j=1}^{n} \mathbf{k}(x_i, x_j)$, $i = 1, \ldots, n$

set the $n \times n$ diagonal matrix D, whose i-th diagonal element is $\mathbf{d}(i)$

set the $n \times n$ row stochastic transition probabilities matrix, $P = D^{-1}K$

set the $n \times n$ matrix $G^{(t)} = \|\mathbf{d}\|_1^{1/2} \, D^{-1/2}(P^*)^t$

apply Algorithm 1 to the set of the columns $\mathcal{G}^{(t)} = \{{\mathbf{g}_1}^{(t)}, \ldots, {\mathbf{g}_n}^{(t)}\}$ in $G^{(t)}$, and μ to get a μ-embedding of $\mathcal{G}^{(t)}$, $\mathbf{f}_\mu$, and a dictionary $\mathcal{D} \subset \mathcal{G}^{(t)}$

define the μ-DM of $\mathcal{X}$ at time t as ${\mathbf{f}_\mu}^{(t)}(x_i) := \mathbf{f}_\mu({\mathbf{g}_i}^{(t)})$, $i = 1, \ldots, n$

define the dictionary ${\mathcal{D}_\mu}^{(t)}$: $x_i \in {\mathcal{D}_\mu}^{(t)}$ if and only if ${\mathbf{g}_i}^{(t)} \in \mathcal{D}$

6. GCA-driven data analysis

In this section, we provide two strongly related tools for data analysis: Out-of-Sample Extension (OSE) and Anomaly Detection (AD). Based on the GCA applied to a dataset $\mathcal{A}$, our goal is to find a low-dimensional representation of any additional data point $\mathbf{a} \notin \mathcal{A}$. Thus, in machine learning term, $\mathcal{A}$ is the training set, by which we learn the low-dimensional representation of the ambient data, which is further applied to the newly-arrived data point $\mathbf{a}$. This process is referred to as OSE. Once an OSE is applied, an interesting question is whether the newly-arrived data point is normal with respect to the training set $\mathcal{A}$. This is determined by the AD phase.

Out-of-sample extension method for DM is also presented in [34], where a Nyström extension is applied to the eigenvectors of the diffusion kernel, which provide the low dimensional embedding of the data.

6.1. Out-of-sample extension and anomaly detection: the linear case

Given a finite dataset $\mathcal{A} \subset \mathbb{R}^m$ and a nonnegative distortion rate μ, Algorithm 1 provides a μ-embedding space $\mathcal{L}$ and a μ-distortion $\mathbf{f}_\mu : \mathcal{A} \to \mathcal{L}$, which is the orthogonal projection of $\mathcal{A}$ onto $\mathcal{L}$. Naturally, we define the extension $\tilde{\mathbf{f}}_\mu : \mathbb{R}^m \to \mathcal{L}$ of $\mathbf{f}_\mu$ to $\mathbb{R}^m$, to be the orthogonal projection onto $\mathcal{L}$, i.e. $\tilde{\mathbf{f}}_\mu(\mathbf{v}) := \mathbf{P}_{\mathcal{L}}(\mathbf{v})$, $\mathbf{v} \in \mathbb{R}^m$. Obviously, the reduction of $\tilde{\mathbf{f}}_\mu$ to $\mathcal{A}$ coincides with $\mathbf{f}_\mu$ and, following Eq. (3.2), for any $\mathbf{a} \in \mathcal{A}$ we have $\left\|\mathbf{a} - \tilde{\mathbf{f}}_\mu(\mathbf{a})\right\| \leq \frac{\mu}{2}$. Consequently, we define the error rate function $\mathcal{E} : \mathbb{R}^m \to \mathbb{R}$ by $\mathcal{E}(\mathbf{v}) := \left\|\mathbf{v} - \tilde{\mathbf{f}}_\mu(\mathbf{v})\right\|$, which is based on a normality of vectors in $\mathbb{R}^m$ with respect to $\mathcal{A}$. A vector $\mathbf{v} \in \mathbb{R}^m$ is defined as normal if

$$\mathcal{E}(\mathbf{v}) \leq \frac{\mu}{2}.$$

Of course, all the elements in $\mathcal{A}$ are normal with respect to $\mathcal{A}$.

6.2. Out-of-sample extension and anomaly detection by DM: the nonlinear case

In this section, we define an extension of the diffusion process to handle a newly-arrived data point for the first $t = 1$ diffusion time-step. Suppose that the diffusion process is already defined for the dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$ via a kernel function $\mathbf{k}$ as described in Section 5.1. Then, as Eq. (5.2) suggests, the (i,j)-th entry of the transition probabilities matrix P is the probability to move from x_i to x_j in a single time-step. Given a data point $x \notin \mathcal{X}$, we are interested in the transition probabilities from x to $\mathcal{X}$. Based on these probabilities, an out-of-sample extension for the DM is defined for x.

For this purpose, assume that the kernel function $\mathbf{k}$ is defined for all pairs (x_i, x), $i = 1, \ldots, n$. We define the transitions probability distribution

$$\mathbb{P}(X_{t+1} = x_j | X_t = x) := \mathbf{k}(x_j, x) / \sum_{i=1}^{n} \mathbf{k}(x_i, x).$$

This definition is consistent with the definition of the transition probabilities from Eqs. (5.1)-(5.2) in the sense that if $x = x_i$ for some $1 \leq i \leq n$, then $\mathbb{P}(X_{t+1} = x_j | X_t = x) = P(i, j)$. Consequently, the diffusion distances of x from the elements in $\mathcal{X}$ are defined to be the ℓ_2 distance between the transition probabilities weighted by the stationary distribution $\hat{\mathbf{d}}$ of the original process, defined on $\mathcal{X}$ (see Eq. (5.3)).

Formally,[4] we define $x_{n+1} := x$ and $\bar{P} := \bar{D}^{-1}\bar{K}$, which is the $(n+1) \times n$ transition probabilities matrix and $\bar{K}$ is the $(n+1) \times n$ matrix whose entries are defined by the kernel function, i.e. $\bar{K}(i, j) := \mathbf{k}(x_i, x_j),\ i = 1, \ldots, n+1,\ j = 1, \ldots, n$, and $\bar{D}$ is the diagonal $(n+1) \times (n+1)$ degrees matrix defined on $\mathcal{X} \cup \{x\}$, whose entries are $\bar{D}(i, i) := \sum_{j=1}^{n} \mathbf{k}(x_i, x_j),\ i = 1, \ldots, n+1$. Then, the extended diffusion distances are defined by

$$\bar{\mathbf{D}}(x_i, x_j) := \left\| \bar{P}(i, :) - \bar{P}(j, :) \right\|_{\ell^2(\hat{\mathbf{d}}^{-1})},\ i, j = 1, \ldots, n+1,$$

where $\hat{\mathbf{d}} \in \mathbb{R}^n$ is the stationary distribution of the diffusion process defined on $\mathcal{X}$, namely $\hat{\mathbf{d}} = \mathbf{d}/\|\mathbf{d}\|_1$, where $\mathbf{d}(i) = \sum_{j=1}^{n} \mathbf{k}(x_i, x_j),\ i = 1, \ldots, n$. Restriction of these diffusion distances to $\mathcal{X}$ coincide with the diffusion distances in Eq. (5.4), i.e. $\bar{\mathbf{D}}(x_i, x_j) = \mathbf{D}(x_i, x_j),\ i, j = 1, \ldots, n$.

Similarly to $G^{(t)}$ in Eq. (5.5), we define the extended $n \times (n+1)$ matrix

$$\bar{G} := \|\mathbf{d}\|_1^{1/2} D^{-1/2} \bar{P}^*,$$

whose i-th column is denoted by $\mathbf{g}_i \in \mathbb{R}^m$ and D is the $n \times n$ diagonal degrees matrix defined for $\mathcal{X}$. Then, the n-dimensional embedding $x_i \mapsto \mathbf{g}_i$ preserves the extended diffusion geometry, i.e. $\bar{\mathbf{D}}(x_i, x_j) = \|\mathbf{g}_i - \mathbf{g}_j\|,\ i, j = 1, \ldots, n+1$.

Thus, the linear OSE and AD for the linear case suggested in Section 6.1 can be applied to the dataset $\mathcal{A} = \{\mathbf{g}_1, \ldots, \mathbf{g}_n\} \subset \mathbb{R}^n$, which is associated with $\mathcal{X}$, and to the vector $\mathbf{g}_{n+1} \in \mathbb{R}^n$ that is associated with x. More specifically, by the application of Algorithm 2 to the dataset $\mathcal{X} = \{x_1, \ldots, x_n\}$, the kernel $\mathbf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and the nonnegative distortion parameter μ in $t = 1$, provides a μ-DM for $\mathcal{X}$, $\mathbf{f}_\mu : \mathcal{X} \to \mathbb{R}^n$ that is defined by the orthogonal projection of the associated vectors onto the p-dimensional μ-embedding space $\mathcal{L}$, i.e.

$$\mathbf{f}_\mu(x_i) := \mathbf{P}_{\mathcal{L}}(\mathbf{g}_i),\ i = 1, \ldots, n. \tag{6.1}$$

The OSE of $\mathbf{f}_\mu$ related to x is defined by

$$\tilde{\mathbf{f}}_\mu(x) := \mathbf{P}_{\mathcal{L}}(\mathbf{g}_{n+1}). \tag{6.2}$$

Similarly to the definition of normality in the linear case, here we define x as normal with respect to $\mathcal{X}$ if the error

$$\mathcal{E}(x) := \|\mathbf{g}_{n+1} - \mathbf{P}_{\mathcal{L}}(\mathbf{g}_{n+1})\| \tag{6.3}$$

satisfies

$$\mathcal{E}(x) \leq \frac{\mu}{2}. \tag{6.4}$$

[4] Since the current discussion is for $t = 1$, all the upper scripts, which indicate the time-step, are omitted from the mathematical notation.

Algorithm 3 concludes the above. Its application follows the application of Algorithm 2 to the reference dataset $\mathcal{X}$. An alternative to this algorithm would be a full computation of the geometric DM for $\mathcal{X} \cup \{x\}$. A significant advantage of Algorithm 3 over this alternative is in its computational complexity, which is $\mathcal{O}(n)$, compared to the $\mathcal{O}((n+1)^2 p)$ complexity of the GCA-based algorithm.

Algorithm 3: Geometric OSE for DM.

Input : Degrees vector $\mathbf{d} \in \mathbb{R}^n$ and the orthogonal projection $\mathbf{P}_{\mathcal{L}} : \mathbb{R}^n \to \mathbb{R}^n$, computed in Algorithm 2, and nonnegative similarities vector $\mathbf{s} = [\mathbf{k}(x_1, x), \ldots, \mathbf{k}(x_n, x)]^* \in \mathbb{R}^n$.

Output: Extension $\tilde{\mathbf{f}}_\mu(x)$ of $\mathbf{f}_\mu : \mathcal{X} \to \mathcal{L}$ by the $\mu-$DM to the new data point x.

1 set the diagonal $n \times n$ matrix D, whose i-th diagonal element is $\mathbf{d}(i)$, $i = 1, \ldots, n$

2 set $\mathbf{p} = \mathbf{s} / \|\mathbf{s}\|_1 \in \mathbb{R}^n$

3 set $\mathbf{g}_{n+1} = \|\mathbf{d}\|_1^{1/2} D^{-1/2} \mathbf{p}$

4 define the extension $\mathbf{f}_\mu$ by $\tilde{\mathbf{f}}_\mu(x) := \mathbf{P}_{\mathcal{L}}(\mathbf{g}_{n+1})$ to x

7. Experimental results

Geometric analyses of three different datasets are presented in this section. Section 7.1 exemplifies the basic notion of geometry preservation, anomaly detection and out-of-sample extension through the application of the QR-based DM to a synthetic dataset as described in Section 5. A comparison with the method proposed in [40] for diffusion geometry preservation is presented in this section as well. A QR-based DM analysis of real data is demonstrated in Section 7.2. The analysis in both of the above examples is based on the corresponding first time step in DM. Finally, Section 7.3 presents a multiclass classification of parametric data by using generalizations of the out-of-sample extension and anomaly detection methods presented in Section 6.1.

7.1. Geometric DM analysis: toy example

A GCA-based DM, which is applied to a two dimensional synthetic manifold immersed in a three dimensional Euclidean space, is presented in this section. The analyzed dataset $\mathcal{X} \subset \mathbb{R}^3$ consists of $n = 3,000$ data points, uniformly sampled from the manifold shown in Fig. 7.1. The utilized kernel function is the commonly-used Gaussian kernel $\mathbf{k}_\epsilon : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$,

$$\mathbf{k}_\epsilon(x, y) := e^{-\|x-y\|^2/\epsilon}, \quad \epsilon > 0, \tag{7.1}$$

where the norm in the exponent is the standard Euclidean norm in $\mathbb{R}^3$. Thus, the associated transition probabilities between close data points are high and low for far data points.

7.1.1. Comparison between geometric DM and spectral DM

The numerical rank (the number significant eigenvalues) of the associated transition probabilities matrix P_ϵ is a decreasing function of ϵ (see [6]). Mathematically, let

$$1 = s_1^{(\epsilon)} \geq s_2^{(\epsilon)} \geq \ldots \geq s_n^{(\epsilon)} \geq 0$$

be the eigenvalues[5] of P_ϵ according to which the truncation of the ordinary (spectral) DM is determined (see Section 5.1.2). Let

[5] The eigenvalues of P_ϵ are nonnegative since the Gaussian kernel function k_ϵ from Eq. (7.1) is positive definite due to Bochner's theorem [45].

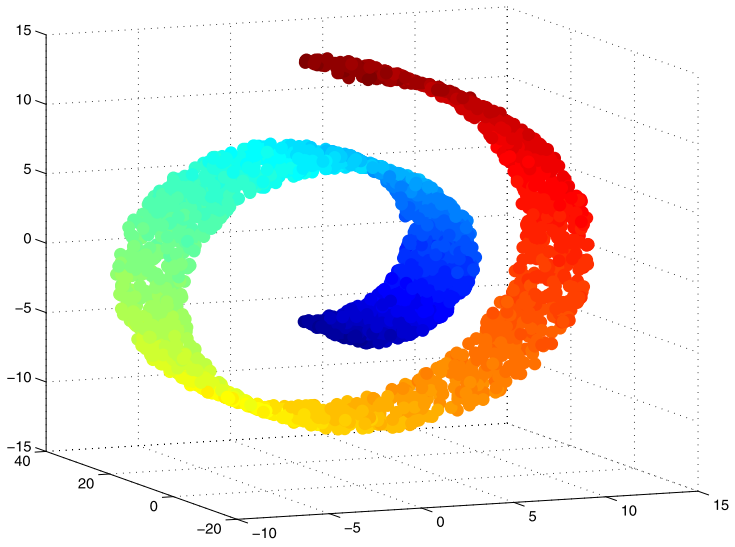


Fig. 7.1. The analyzed dataset that is uniformly sampled from this Swiss roll manifold.

$$\mathbf{E}_\epsilon(k/n) := \left(\sum_{i=1}^{k} (s_i^{(\epsilon)})^2 / \sum_{i=1}^{n} (s_i^{(\epsilon)})^2 \right)^{1/2}, \quad k = 1, \ldots, n$$

be the energy's portion of P_ϵ, which is captured by the first k eigenvalues of P_ϵ. Then, as ϵ increases, the number of significant eigen-component decreases and the diffusion geometry becomes plain in the sense that most distances are short as demonstrated in Fig. 7.2. This fact, combined with Lemma 5.1 suggests that when ϵ decreases the number of principal components required to achieve a certain distortion increases as seen in Fig. 7.3.

The analytic bound from Lemma 5.1 is an upper bound for the dimension of the ordinary (spectral) DM that guarantees a certain distortion. Fig. 7.3 compares between the analytic bound, the actual spectral components and the GCA dimensions required to achieve a certain μ-distortion of the DM geometry.

7.1.2. Out of sample extension and anomaly detection

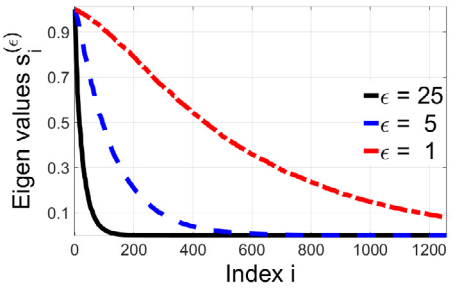
Application of Algorithm 2 to $\mathcal{X}$ with the Gaussian kernel $\mathbf{k}_3$ and $\mu = 0.2$ (and $t = 1$) was resulted in a p-dimensional μ-DM $\mathbf{f}_\mu : \mathcal{X} \to \mathbb{R}^p$ with $p = 1,246$. At the next step, an OSE of $\mathbf{f}_\mu$ is applied to each of the elements in $\bar{\mathcal{X}}$, where $\bar{\mathcal{X}} \subset \mathbb{R}^3$ is a random subset of $10,000$ data points uniformly sampled from a three dimensional bounding box of $\mathcal{X}$. For that purpose, Algorithm 3 was applied to each $\bar{x} \in \bar{\mathcal{X}}$ with the similarities vector $\mathbf{s} \in \mathbb{R}^n$, whose entries are $\mathbf{s}(i) = \mathbf{k}_3(\bar{x}, x_i),\ x_i \in \mathcal{X}$.

Finally, the elements in $\bar{\mathcal{X}}$ were classified as normal and abnormal using Eqs. (6.1)-(6.4). The results are shown in Fig. 7.4.

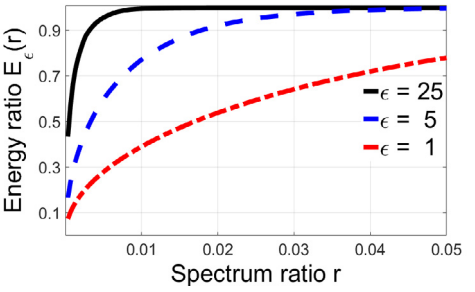
7.1.3. Comparison with μIDM

The μIDM algorithm in [40] is a dictionary-based method that provides a low dimension μ-distortion for the first time step of the diffusion geometry. The algorithm incrementally constructs an approximated map by using a single scan of the data. The μIDM algorithm is greedy and sensitive to the scan order. Typically, the growth rate of the dictionary is very high at the beginning and decays afterwards, and may be redundant. As a result, the embedding's dimension may be redundant as well.

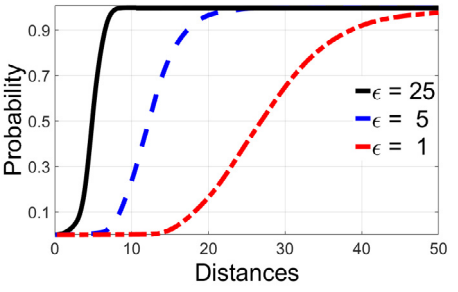
The presently proposed geometric DM method considers the entire dataset in each iteration and it is not sensitive to the order of the dataset. Therefore, the resulted dictionary is more sparse as demonstrated in Table 7.1 and in Fig. 7.5.



(a) Spectra of P_ϵ



(b) Spectra ratios of P_ϵ



(c) Diffusion distances distributions

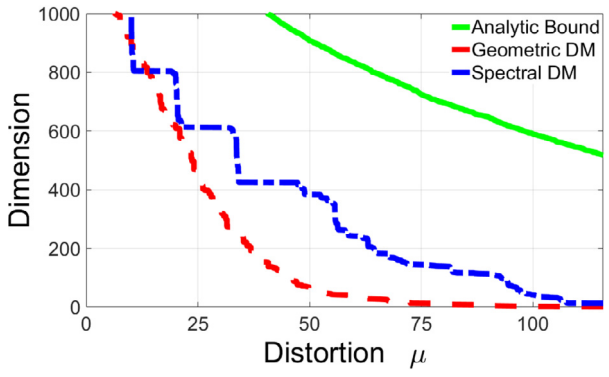
Fig. 7.2. Spectral and geometrical views of three diffusion geometries that correspond to $\epsilon = 1, 5$ and 25. (a) shows that as ϵ becomes larger the spectrum decays faster. An immediate consequence is shown in (b) that shows the relation between the number of significant component and ϵ. (c) shows the probability distribution of the diffusion distances. It is clear that the use of large ϵ results in many short diffusion distances and vice versa. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 7.1
Comparison between geometric DM and μIDM algorithms for various distortion parameters such as dictionary sizes, execution times and actual distortion.[6] Execution times are averaged over 10 runs of the algorithms. The algorithms ran on Matlab2014.

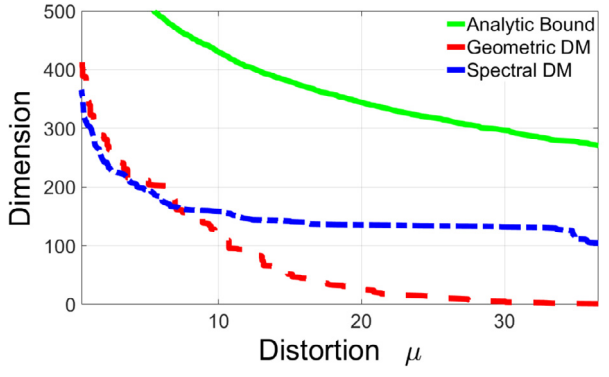
μ		Geometric DM	μIDM [40]
0.2	Dictionary size	1,246	2,305
	Execution time	43 sec.	7 hours
	Actual distortion	0.01	0.03
2	Dictionary size	752	1,293
	Execution time	27 sec.	71 minutes
	Actual distortion	0.23	0.61
10	Dictionary size	382	630
	Execution time	17 sec.	15 minutes
	Actual distortion	3.91	4.46
20	Dictionary size	190	284
	Execution time	9 sec.	4 minutes
	Actual distortion	12.81	13.24

The method in this paper generalizes the original μIDM algorithm in [40] by being more general and less specific such as fitting only to DM.

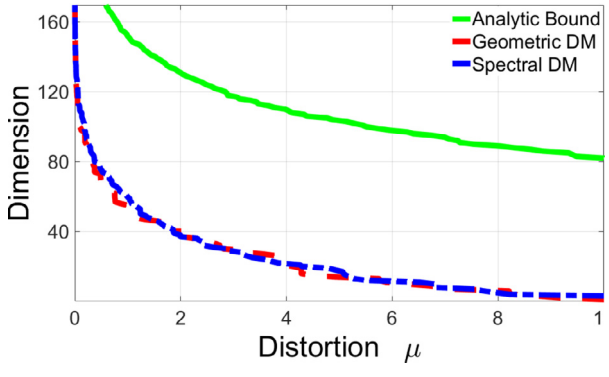
[6] An actual distortion of a function f w.r.t. g is $\sup_{x,y\in\mathcal{X}} |\|f(x) - f(y)\| - \|g(x) - g(y)\||$.



(a) $\epsilon = 1$



(b) $\epsilon = 5$

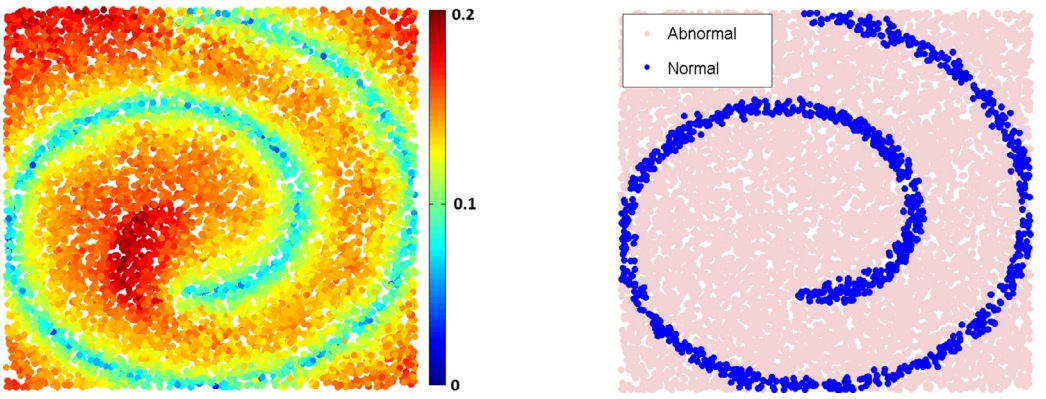


(c) $\epsilon = 25$

Fig. 7.3. The required dimensions for μ-distortion of the diffusion geometry. The continuous (green) graph denotes the analytic bound provided by Lemma 5.1, the dashed (red) graph is the μ-DM dimension produced by Algorithm 2 and the dash-dotted (blue) graph is the required minimal dimension required to achieve a μ-distortion by the spectral DM presented in Section 5.1.2. The smaller ϵ the diffusion geometry is richer and higher dimension is required to achieve a μ-distortion.

7.2. Geometric DM analysis: real-world data

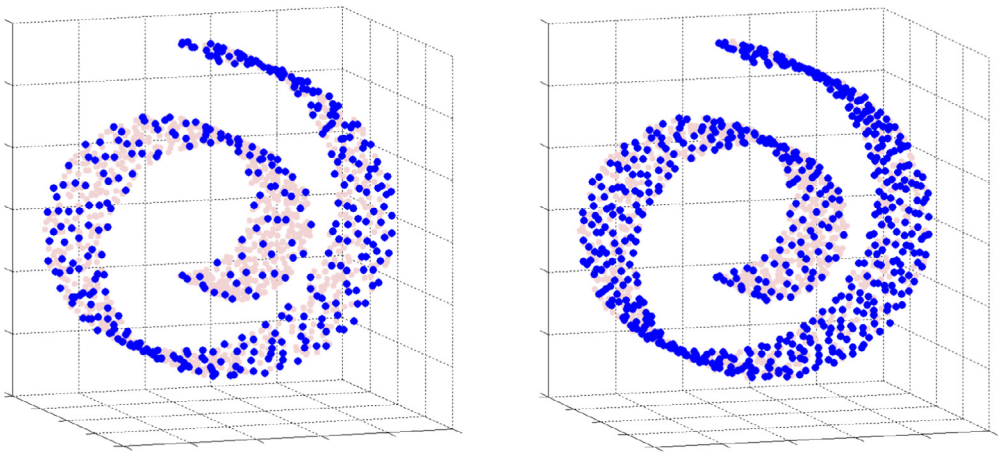
An unsupervised anomaly detection process applied to a real-world dataset is exemplified in this section. The examined dataset $\bar{\mathcal{X}} \subset \mathbb{R}^{14}$ is the DARPA dataset [24] consists of $\bar{n} = 12,617$ data points. Each



(a) Distortion rate function $\mathcal{E} : \bar{\mathcal{X}} \to \mathbb{R}$.

(b) Classification of $\bar{\mathcal{X}}$ to normal ($\mathcal{E}(\bar{x}) \leq \mu/2$) and abnormal (otherwise) classes.

Fig. 7.4. A side view of the dataset $\bar{\mathcal{X}}$. (a) Each data point $x \in \bar{\mathcal{X}}$ is colored proportionally to its out-of-sample extension distortion rate $\mathcal{E}(x)$. (b) Classification of $\bar{\mathcal{X}}$ to either normal (darkly colored) or abnormal (brightly colored).



(a) Geometric DM Dictionary, 382 data points

(b) μIDM Dictionary, 630 data points

Fig. 7.5. Dictionary points (dark points) of (a) geometric DM and (b) μIDM dictionaries with distortion parameter $\mu = 10$. Running time differences are due to different computational costs: $\mathcal{O}(n^2 p)$ for GCA versus $\mathcal{O}(n^2 p^2 + np^3)$ for μ-IDM, where n and p are the sizes of the dataset and the dictionary, respectively.

data point is a vector of 14 features that describes a computer network traffic, labeled as either normal or abnormal that pertains to be an attack (intrusion) on the network. The dataset is divided into a training set $\mathcal{X} \subset \bar{\mathcal{X}}$, which contains $n = 6,195$ normally behaving samples, and 5 testing subsets $\mathcal{T}_{mon}, \ldots, \mathcal{T}_{fri}$, collected during different days of the week, each of which contains normal and abnormal data points as described in Table 7.2.

First, the training dataset was scaled[7] to the 14-dimensional unit box $[0,1]^{14}$. Then, the same scaling was applied to the testing datasets $\mathcal{T} = \mathcal{T}_{mon} \cup \ldots \cup \mathcal{T}_{fri}$. Application of Algorithm 2 to the training set $\mathcal{X}$, using the Gaussian kernel $\mathbf{k}_\epsilon$ from Eq. (7.1) with $\epsilon = 0.6$, $t = 1$ and $\mu = 0.5 \times 10^{-8}$, produces a p-dimensional μ-DM of $\mathcal{X}$, $\mathbf{f}_\mu{}^{(t)} : \mathcal{X} \to \mathbb{R}^p$ with $p = 138$. The parameter ϵ was chosen to be twice the median of all the mutual distances between the data points in $\mathbb{R}^{14}$. Such a selection is a common heuristic for determining this parameter in DM context. This concludes the training phase.

[7] Each one of the 14-features vectors $\mathbf{f}$ was independently shifted $\mathbf{f} = \mathbf{f} - \min_{i=1,\ldots,n} \mathbf{f}(i)$ and stretched $\mathbf{f} = \mathbf{f} / \max_{i=1,\ldots,n} \mathbf{f}(i)$.

Table 7.2
Anomaly detection performances. Accuracy stands for the portion of detected anomalies out of the labeled anomalies where False Alarms are the portion of falsely detected anomalies out of the whole data.

Set	Size	# of labeled anomalies	Accuracy [%]	False Alarms [%]
$\mathcal{T}_{mon}$	$1,321$	1	100	0.68
$\mathcal{T}_{tue}$	$1,140$	53	100	0.53
$\mathcal{T}_{wed}$	$1,321$	16	100	0.08
$\mathcal{T}_{thu}$	$1,320$	24	96	1.74
$\mathcal{T}_{fri}$	$1,320$	18	100	0.15

As a second step, an OSE is applied to each testing point $\bar{x} \in \mathcal{T}$ by using Algorithm 3 with similarities vector $\mathbf{s} \in \mathbb{R}^n$, whose entries are $\mathbf{s}(i) = \mathbf{k}_\epsilon(\bar{x}, x_i),\ x_i \in \mathcal{X}$. Any testing data point, which is distant (relatively to ϵ) from the training set $\mathcal{X}$, yields $\mathbf{k}_\epsilon(\bar{x}, x) \approx 0,\ x \in \mathcal{X}$, and was a-priori classified as abnormal. This subset is denoted by $\mathcal{F}$. The rest of the data points in $\mathcal{T}$ were classified using the procedure described by Eqs. (6.1)-(6.4).

The anomaly detection results are summarized in Table 7.2. Some of them are demonstrated in Fig. 7.6. The rates in the accuracy percentage and in the false alarms columns are related to the true known labeling of $\bar{\mathcal{X}}$.

Fig. 7.6 presents three-dimensional views[8] of the μ-DM, as well as its extension to $\mathcal{T}_{thu} \backslash \mathcal{F}$.

7.3. Unsupervised multi-class classification of high dimensional data

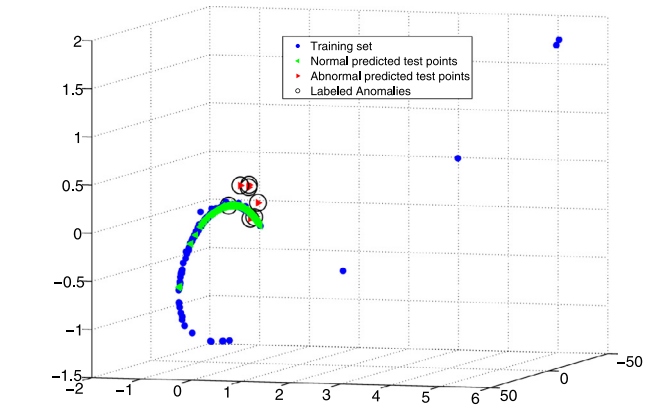
A multi-classification process, which is based on Algorithm 1 and the OSE procedure from Section 6.1, is presented in this section. The analyzed data is the m-dimensional ISOLET dataset [3] $\bar{\mathcal{X}} \subset \mathbb{R}^m$ where $m = 617$. It contains $7,797$ data points in $[-1, 1]^m$. Each data point is a list of m features extracted from a single human pronunciation of ISOlated LETters. The features are described in [20], and they include spectral coefficients of the speech signals, contour features, sonorant features, pre-sonorant features, and post-sonorant features. The goal is to classify a testing subset $\mathcal{T} \subset \bar{\mathcal{X}}$ of $1,559$ letter-pronunciation samples spoken by 30 people, to 26 classes. These are based on a training set $\mathcal{X} \subset \bar{\mathcal{X}}$ of $n = 6,238$ samples that were already classified to $\mathcal{X}_\zeta$, $\zeta \in \mathcal{I} := \{A, B, \ldots, Z\}$ spoken by 120 different people.

For this purpose, Algorithm 1 was independently applied to each training set $\mathcal{X}_\zeta$ with a distortion parameter[9] $\mu = 9.4$ to produce 26 μ-embeddings $\mathbf{f}_\mu{}^{(\zeta)}$. Consequently, the OSE (described in Section 6.1) of each such embedding, denoted by $\tilde{\mathbf{f}}_\mu^{(\zeta)}$, was applied to each testing data point $\mathbf{x} \in \mathcal{T}$ and the corresponding error $\mathcal{E}^{(\zeta)}(\mathbf{x}) := \left\| \mathbf{x} - \tilde{\mathbf{f}}_\mu^{(\zeta)}(\mathbf{x}) \right\|$ was computed. Finally, each test point $\mathbf{x} \in \mathcal{T}$ was classified to the closest class ζ_0, i.e. $\zeta_0 := \arg\min_{\zeta \in \mathcal{I}} \mathcal{E}^{(\zeta)} - (\mathbf{x})$.

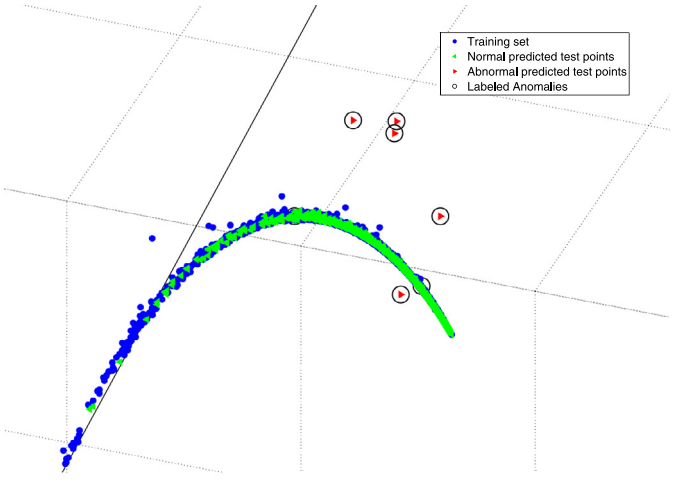
Out of $1,559$ test samples, 92% were classified correctly. The classification of the test data is presented in a confusion matrix in Fig. 7.7. By looking at the shades of the diagonal it can be seen that the majority of the test samples of each class were classified correctly. The sets $\{B, C, D, E, G, P, T, V, Z\}$ and $\{M, N\}$ are the most difficult letters to classify due to high similarity in pronunciation of these letters within each set. The state-of-the-art classification accuracy of this dataset is 96.73%. It was achieved in [17] by using 30-bit error correcting output codes that is based on neural networks. This method is far more complex than the solution proposed in this work.

[8] The dimension of the μ-DM in this example is $p = 138$. However, for visualization purposes, we present its first three principal components.

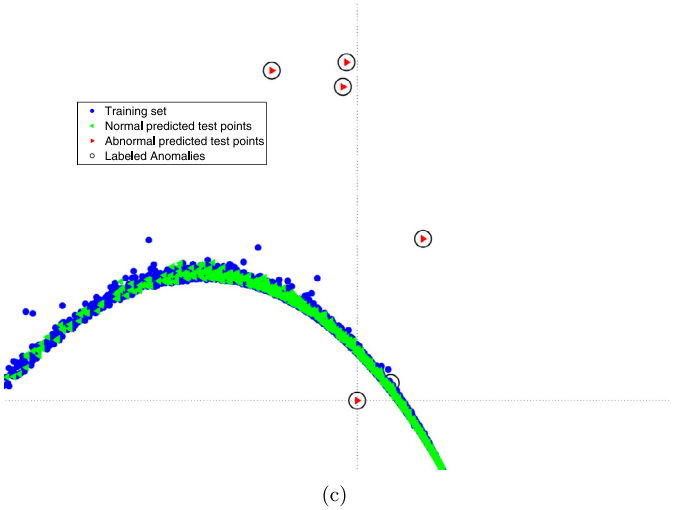
[9] The parameter μ was determined by taking part of the training set to serve as a validation set. Then, several values of μ were applied to the (reduced) training set and the validation set was classified based upon these values. The chosen μ was the one that was optimal on the validation set.



(a)



(b)



(c)

Fig. 7.6. Three different views and scales of the μ-DM of $\mathcal{X}$ (blue points) and its OSE to $\mathcal{T}_{thu}\backslash\mathcal{F}$ (normal data points are green, abnormal data points are red) and labeled anomalies (black circles). (a) General view. (b) Normal OSE data points are mapped closely to the training set. (c) Abnormal data points.

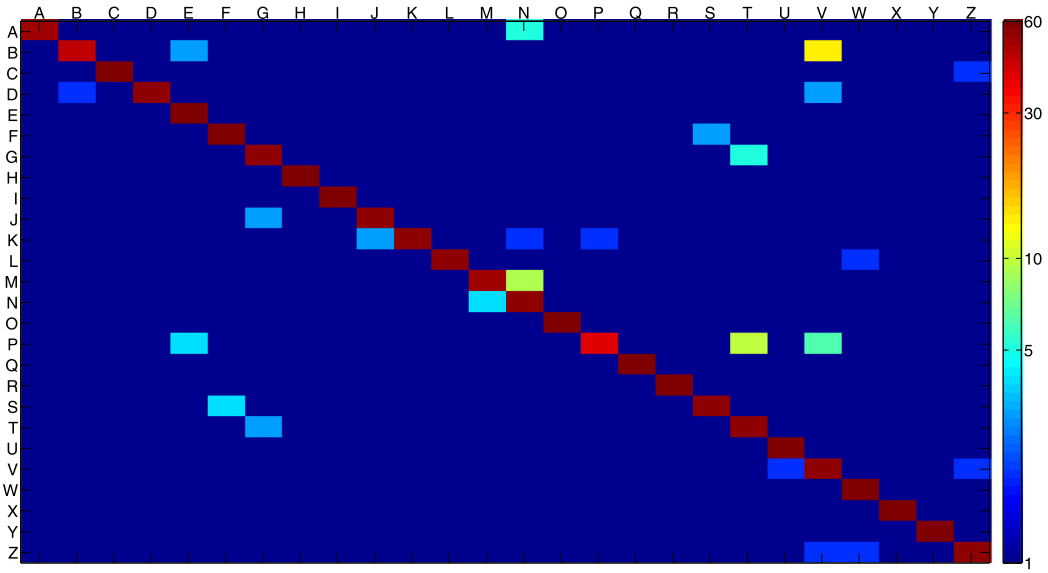


Fig. 7.7. Classification of the test samples in the ISOLET dataset. For each ordered couple (i, j), the cell at row i and column j is colored according to the number of test samples belonging to class i that were classified by the algorithm to class j. The cells on the diagonal denote correct classification. For each letter, a total of 60 test samples are provided (except for 'M' which is missing one sample due to recording difficulties).

Table 7.3
Multi-Class classification performances. Accuracy stands for the portion of true detected out of the total class size.

Class	Size	GCA-DM Accuracy [%]	NN Accuracy [%]
1	11480	**99.983**	99.956
2	12	**100.000**	91.667
3	38	**100.000**	97.368
4	2155	100.000	100.000
5	800	100.000	100.000
6	5	**80.000**	60.000
7	2	100.000	100.000

7.4. Supervised multi-class classification of the Shuttle dataset

A multi-classification process, which is based on Algorithm 2 and on the OSE procedure from Section 6.1, is presented in this section. The analyzed data $\bar{\mathcal{X}} \subset \mathbb{R}^9$ is the 9-dimensional Shuttle dataset [3]. It contains a dataset $\mathcal{X}$ that has $43,500$ data points in $\mathbb{R}^9$ for training and a dataset $\mathcal{T}$ that contains $14,500$ data points for testing. This dataset contains 7 highly imbalanced classes. The goal is to classify the testing dataset into one from the 7 classes $\zeta \in \mathcal{I} := \{1, 2, \ldots, 7\}$. For this purpose, Algorithm 2 was applied to the entire training data set $\mathcal{X}$ with a distortion parameter $\mu = 10^{-4}$ that was learned on a subset of the training data set. The application of Algorithm 2 produces a 14 μ-embeddings $\mathbf{f}_\mu{}^{(\zeta)}$. Consequently, the OSE (described in Section 6.1) of the computed embedding, denoted by $\tilde{\mathbf{f}}_\mu^{(\zeta)}$, was applied to each testing data point $\mathbf{x} \in \mathcal{T}$. Finally, each test point $\mathbf{x} \in \mathcal{T}$ was classified according to the label of the nearest neighbor in the training set $\mathcal{X}$.

Out of $14,500$ test samples, $14,497$, which are 99.97% of the total data points, were classified correctly. For completeness of the discussion we compared the GCA-DM results to a simple nearest-neighbor that is based on the given dataset $\mathcal{X}$. The classification results of the test set $\mathcal{T}$ that includes both methods are presented in Table 7.3.

The results suggest that for all classes sizes, the GCA-DM succeeded to distill better representation in terms of distance to the true label class. From 14,500 data points, the number of false classifications generated by GCA-DM is 3 (99.998% overall accuracy) while NN produces 9 (99.994% overall accuracy).

8. Conclusions and future works

Geometric Component Analysis, which is a deterministic rigorously-proven method for linear and non-linear dimensionality reduction, is presented in this paper. Unlike the well-known PCA-based dimensionality reduction method, which is statistically-driven, this method is geometrically-driven in the sense that it is designed to achieve a predefined distortion for a given geometry. Moreover, as opposed to PCA, the GCA provides a dictionary, namely a subset of significant data points that well represent the dataset and its storage and computational complexities are lower than the PCA's. The method was shown to be strongly related to the incomplete pivoted QR matrix factorization.

Experimental results, which include both linear and diffusion geometries, show that our method achieves a lower dimensional embedding than what PCA achieves for a certain distortion. It has also been shown that the GCA is more efficient than the μIDM [40], which is also a deterministic method for dimensionality reduction achieved through the application of diffusion maps. Moreover, analysis of both synthetic and real-world datasets achieved good performance for dimensionality reduction, out-of-sample extension, anomaly detection and multi-class classification.

Future work includes the construction of randomized version of the presented framework to provide a more computationally efficient method for a dictionary-based dimensionality reduction method. In addition, a non-greedy algorithm would be considered as it might be resulted in sparser dictionary, which is equivalent to a lower-dimensional distortion of the dataset.

Acknowledgments

This research was partially supported by the Israeli Ministry of Science & Technology INdo-ISrael CollAborative Grant No. 3-14481, Science Foundation (ISF 1556/17) and Blavatnik Computer Science Research & Blavatnik ICRC Funds. The authors thank to Aviv Rotbart, who helped in producing the experimental results presented in Section 7.

Appendix A. Stability of the GCA to additive noise

In real-life, data may be noisy and we have an access only for a noisy version of the dataset $\mathcal{A}$. Proposition A.1 shows that a distorted embedding of a noisy data is also a distorted embedding of the original clean data where the error originated by noise is additive.

Proposition A.1. *Let $\tilde{\mathcal{A}} = \{\tilde{\mathbf{v}} := \mathbf{v} + \nu(\mathbf{v}) \mid \mathbf{v} \in \mathcal{A}\}$, such that for any $\mathbf{v} \in \mathcal{A}$, $\|\nu(\mathbf{v})\| \leq \eta/2$ for some $\eta \geq 0$. If $\mathbf{f}_\mu$, produced by Algorithm 1, is a μ-embedding of $\tilde{\mathcal{A}}$, then it is a $(\mu + \eta)$-embedding of $\mathcal{A}$.*

Proof. Let $\mathbf{f}_\mu$ and $\mathcal{L}_\mu$ be μ-embedding and μ-space, respectively, of $\tilde{\mathcal{A}}$. Then, for each $\mathbf{u}, \mathbf{v} \in \mathcal{A}$

$$\begin{aligned}
\left| \|\mathbf{u} - \mathbf{v}\| - \|\mathbf{f}_\mu(\tilde{\mathbf{u}}) - \mathbf{f}_\mu(\tilde{\mathbf{v}})\| \right| &= \left| \|\mathbf{u} - \mathbf{v}\| - \left\|\mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{u}}) - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{v}})\right\| \right| \\
&\leq \left\|\mathbf{u} - \mathbf{v} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{u}}) + \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{v}})\right\| \\
&\leq \left\|\mathbf{u} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{u}})\right\| + \left\|\mathbf{v} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{v}})\right\| \\
&\leq \|\mathbf{u} - \tilde{\mathbf{u}}\| + \left\|\tilde{\mathbf{u}} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{u}})\right\| + \|\mathbf{v} - \tilde{\mathbf{v}}\| + \left\|\tilde{\mathbf{v}} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{v}})\right\|
\end{aligned}$$

$$= \|\nu(\mathbf{u})\| + \left\|\tilde{\mathbf{u}} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{u}})\right\| + \|\nu(\mathbf{v})\| + \left\|\tilde{\mathbf{v}} - \mathbf{P}_{\mathcal{L}_\mu}(\tilde{\mathbf{v}})\right\|$$
$$\leq \mu + \eta. \quad \square$$

References

[1] D. Achlioptas, Database-friendly random projections: Johnson-Lindenstrauss with binary coins, J. Comput. Syst. Sci. 66 (4) (2003) 671–687.
[2] N. Ailon, B. Chazelle, Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform, in: Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing, ACM, 2006, pp. 557–563.
[3] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013.
[4] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin, A simple proof of the restricted isometry property for random matrices, Constr. Approx. 28 (3) (2008) 253–263.
[5] P. Berard, G. Besson, S. Gallot, Embedding Riemannian manifolds by their heat kernel, Geom. Funct. Anal. 4 (4) (1994) 373–398.
[6] A. Bermanis, A. Averbuch, R.R. Coifman, Multiscale data sampling and function extension, Appl. Comput. Harmon. Anal. 34 (1) (2013) 15–29.
[7] A. Bermanis, G. Wolf, A. Averbuch, Cover-based bounds on the numerical rank of Gaussian kernels, Appl. Comput. Harmon. Anal. 36 (2) (2014) 302–315.
[8] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald DeVore, Guergana Petrova, Przemyslaw Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods, SIAM J. Math. Anal. 43 (3) (2011) 1457–1472.
[9] C. Boutsidis, M.W. Mahoney, P. Drineas, Unsupervised feature selection for principal components analysis, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 61–69.
[10] C. Boutsidis, M.W. Mahoney, P. Drineas, An improved approximation algorithm for the column subset selection problem, in: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '09, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009, pp. 968–977.
[11] C. Boutsidis, J. Sun, N. Anerousis, Clustered subset selection and its applications on it service metrics, in: Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, ACM, New York, NY, USA, 2008, pp. 599–608.
[12] C. Boutsidis, D.P. Woodruff, Optimal cur matrix decompositions, in: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, ACM, 2014, pp. 353–362.
[13] H. Cheng, Z. Gimbutas, P.G. Martinsson, V. Rokhlin, On the compression of low rank matrices, SIAM J. Sci. Comput. 26 (4) (April 2005) 1389–1404.
[14] K.L. Clarkson, Tighter bounds for random projections of manifolds, in: Proceedings of the Twenty-Fourth Annual Symposium on Computational Geometry, ACM, 2008, pp. 39–48.
[15] R.R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.
[16] Ronald DeVore, Guergana Petrova, Przemyslaw Wojtaszczyk, Greedy algorithms for reduced bases in Banach spaces, Constr. Approx. 37 (3) (2013) 455–466.
[17] T.G. Dietterich, G. Bakiri, Error-correcting output codes: a general method for improving multiclass inductive learning programs, in: Proceedings of AAAI-91, AAAI Press, 1991, pp. 572–577.
[18] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Subspace sampling and relative-error matrix approximation: column-based methods, in: Proc. of the 10th RANDOM, 2006, pp. 316–326.
[19] P. Drineas, M.W. Mahoney, S. Muthukrishnan, Relative-error cur matrix decompositions, SIAM J. Matrix Anal. Appl. 30 (2) (2008) 844–881.
[20] M.A. Fanty, R. Cole, Spoken letter recognition, in: NIPS, 1990, p. 220.
[21] A. Frieze, R. Kannan, S. Vempala, Fast Monte-Carlo algorithms for finding low-rank approximations, J. ACM 51 (6) (2004) 1025–1041.
[22] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins University Press, Baltimore, MD, USA, 1996.
[23] G.H. Golub, C.F. Van Loan, Matrix Computations, fourth edition, Johns Hopkins University Press, 2013.
[24] I. Graf, R. Lippmann, R. Cunningham, D. Fried, K. Kendall, S. Webster, M. Zissman, Results of darpa 1998 offline intrusion detection evaluation, in: DARPA PI Meeting, vol. 15, 1998.
[25] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions, SIAM Rev. 53 (2) (2011) 217–288.
[26] John Harlim, Haizhao Yang, Diffusion forecasting model with basis functions from qr-decomposition, J. Nonlinear Sci. (2017) 1–26.
[27] R. Tibshirani, T. Hastie, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer Science & Business Media, 2009.
[28] H. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educ. Psychol. 24 (1933).
[29] G. Hughes, On the mean accuracy of statistical pattern recognizers, IEEE Trans. Inf. Theory 14 (1) (January 1968) 55–63.
[30] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, ACM, 1998, pp. 604–613.
[31] L.O. Jimenez, D.A. Landgrebe, Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data, IEEE Trans. Syst. Man Cybern., Part C, Appl. Rev. 28 (1) (Feb 1998) 39–54.
[32] W.B. Johnson, J. Lindenstrauss, Extensions of Lipschitz mappings into a Hilbert space, Contemp. Math. 26 (189–206) (1984) 1.

[33] N. Linial, E. London, Y. Rabinovich, The geometry of graphs and some of its algorithmic applications, Combinatorica 15 (2) (1995) 215–245.
[34] Andrew W. Long, Andrew L. Ferguson, Landmark diffusion maps (l-dmaps): accelerated manifold learning out-of-sample extension, Appl. Comput. Harmon. Anal. 47 (1) (2019) 190–211.
[35] M.W. Mahoney, Randomized algorithms for matrices and data, Found. Trends Mach. Learn. 3 (2) (2011) 123–224.
[36] M.W. Mahoney, P. Drineas, Cur matrix decompositions for improved data analysis, Proc. Natl. Acad. Sci. 106 (3) (2009) 697–702.
[37] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, Online learning for matrix factorization and sparse coding, J. Mach. Learn. Res. 11 (Jan 2010) 19–60.
[38] P.G. Martinsson, V. Rokhlin, M. Tygert, A randomized algorithm for the decomposition of matrices, Appl. Comput. Harmon. Anal. 30 (1) (2011) 47–68.
[39] Ignacio Ramirez, Pablo Sprechmann, Guillermo Sapiro, Classification and Clustering via Dictionary Learning with Structured Incoherence and Shared Features, 2010.
[40] M. Salhov, A. Bermanis, G. Wolf, A. Averbuch, Approximately-isometric diffusion maps, Appl. Comput. Harmon. Anal. 38 (2015) 399–415.
[41] L.J. Schulman, Clustering for edge-cost minimization, in: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, ACM, 2000, pp. 547–555.
[42] I. Toic, P. Frossard, Dictionary learning: what is the right representation for my signal, IEEE Signal Process. Mag. 28 (2) (2011) 27–38.
[43] S. Vempala, The Random Projection Method, vol. 65, American Mathematical Soc., 2005.
[44] S. Wang, Z. Zhang, Improving cur matrix decomposition and the Nyström approximation via adaptive sampling, J. Mach. Learn. Res. 14 (1) (2013) 2729–2769.
[45] H. Wendland, Scattered Data Approximation, Cambridge University Press, 2005.