Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-020-00692-x

Gaussian bandwidth selection for manifold learning and classification

Ofir Lindenbaum[1] · **Moshe Salhov**[2] · **Arie Yeredor**[1] · **Amir Averbuch**[2]

Received: 29 May 2018 / Accepted: 16 May 2020
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2020

Abstract
Kernel methods play a critical role in many machine learning algorithms. They are useful in manifold learning, classification, clustering and other data analysis tasks. Setting the kernel's scale parameter, also referred to as the kernel's bandwidth, highly affects the performance of the task in hand. We propose to set a scale parameter that is tailored to one of two types of tasks: classification and manifold learning. For manifold learning, we seek a scale which is best at capturing the manifold's intrinsic dimension. For classification, we propose three methods for estimating the scale, which optimize the classification results in different senses. The proposed frameworks are simulated on artificial and on real datasets. The results show a high correlation between optimal classification rates and the estimated scales. Finally, we demonstrate the approach on a seismic event classification task.

Keywords Dimensionality reduction · Kernel methods · Diffusion maps · Classification

1 Introduction

Dimensionality reduction is an essential step in numerous machine learning tasks. Methods such as Principal Component Analysis (PCA) (Jolliffe 2002), Multidimensional Scaling (MDS) (Kruskal 1977), Isomap (Tenenbaum et al. 2000) and Local Linear Embedding (Roweis and Saul 2000) aim to extract essential information from high-dimensional data points based on their pairwise connectivities. Graph-based kernel methods such as Laplacian Eigenmaps (Belkin and Niyogi 2001) and Diffusion

Responsible editor: Jieping Ye.

✉ Ofir Lindenbaum
ofirlin@gmail.com

1 School of Electrical Engineering, Tel Aviv University, Tel Aviv, Israel

2 School of Computer Science, Tel Aviv University, Tel Aviv, Israel

Published online: 02 July 2020

Maps (DM) (Coifman and Lafon 2006), construct a positive semi-definite kernel based on the multidimensional data points to recover the underlying structure of the data. Such methods have been proven effective for tasks such as clustering (Luo 2011), classification (Lindenbaum et al. 2015), manifold learning (Lin et al. 2006) and many more.

Kernel methods rely on computing a distance function (usually Euclidean) between all pairs of data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X} \in \mathbb{R}^{D \times N}$ and application of a data dependent kernel function. This kernel should encode the inherited relations between high-dimensional data points. An example for a kernel that encapsulates the Euclidean distance takes the form

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq \mathcal{K}\left(\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\epsilon}\right) = K_{i,j}, \tag{1.1}$$

(where $\|\cdot\|$ denotes the Euclidean norm). As shown, for example, in Roweis and Saul (2000), Coifman and Lafon (2006), spectral analysis of such a kernel provides an efficient representation of the lower (d-) dimensional data (where $d \ll D$) embedded in the ambient space. Devising the kernel to work successfully in such contexts requires expert knowledge for setting two parameters, namely the scaling ϵ (Eq. 1.1) and the inferred dimension d of the low-dimensional space. We focus in this paper on setting the scale parameter ϵ, sometimes also called the *kernel bandwidth*.

The scale parameter is related to the statistics and to the geometry of the data points. The Euclidean distance, which is often used for learning the geometry of the data, is meaningful only locally when applied to high-dimensional data points. Therefore, a proper choice of ϵ should preserve local connectivities and neglect large distances. If ϵ is too large, there is almost no preference for local connections and the kernel method is reduced essentially to PCA (Lindenbaum et al. 2015). On the other hand, if ϵ is too small, the matrix $\boldsymbol{K}$ (Eq. 1.1) has many small off-diagonal elements, which is an indication of a poor connectivity within the data.

Several studies have proposed different approaches for setting ϵ. A study by Lafon et al. (2006) suggests a method which enforces connectivity among most data points—a rather simple method, which is nonetheless sensitive to noise and to outliers. The sum of the kernel is used by Singer et al. (2009) to find a range of valid scales, this method provides a good starting value but is not fully automated. An approach by Zelnik-Manor and Perona (2004) sets an adaptive scale for each point, which is applicable for spectral clustering but might deform the geometry. As a result, there is no guarantee that the rescaled kernel has real eigenvectors and eigenvalues. Others simply use the squared standard deviation (mean squared Euclidean distances from the mean) of the data as ϵ, this again is very sensitive to noise and to outliers.

Kernel methods are also used for Support Vector Machines (SVMs, Scholkopf and Smola 2001), where the goal is to find a feature space that best separates given classes. Methods such as (Gaspar et al. 2012; Staelin 2003) use cross-validation to find the scale parameter which achieves peak classification results on a given training set. The study by Campbell et al. (1999) suggests an iterative approach that updates the scale until reaching maximal separation between classes. Chapelle et al. (2002) relate the scale parameter to the feature selection problem by using a different scale for each feature. This framework applies gradient descent to a designated error function to find

the optimal scales. These methods are good for classification, but require to actually re-classify the points for testing each scale.

In this paper we propose methods to estimate the scale which do not require the repeated application of a classifier to the data. Since the value of ϵ defines the connectivity of the resulted kernel matrix (Eq. 1.1), its value is clearly crucial for the performance of kernel based methods. Nonetheless, the performance of such methods depends on the training data and on the optimization problem in hand. Thus, in principle we cannot define an 'optimal' scaling parameter value independently of the data. We therefore focus on developing tools to estimate a scale parameter based on a given training set. We found that there are almost no simple methods focusing on finding element-wise (rather than one global) scaling parameters dedicated for manifold learning. Neither are there methods that try to maximize classification performance without directly applying a classifier. For these reasons we propose new methodologies for setting ϵ, dedicated either to manifold learning or to classification.

For the manifold learning task, we start by estimating the manifold's intrinsic dimension. Then, we introduce a vector of scaling parameters $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_D]$, such that each value $\epsilon_i, i = 1, ..., D$, rescales each feature. We propose a special greedy algorithm to find the scaling parameters which best capture the estimated intrinsic dimension. This approach is analyzed and simulated to demonstrate its advantage.

For the classification task, we propose three methods for finding a scale parameter. In the first, by extending (Lindenbaum et al. 2016) we seek a scale which provides the maximal separation between the classes in the extracted low-dimensional space. The second is based on the eigengap of the kernel. It is justified based on the analysis of a perturbed kernel. The third method sets the scale which maximizes the within-class transition probability. This approach does not require to compute an eigendecomposition.

Additionally, we provide new theoretical justifications for the eigengap-based method, as well as new simulations to support all methods. Interestingly, we also show empirically that all the three methods converge to a similar scale parameter ϵ.

The structure of the paper is as follows: Preliminaries are given in Sect. 2. Section 3 presents and analyzes two frameworks for setting the scale parameter: the first is dedicated to a manifold learning task while the second fits a classification task. Section 4 presents experimental results. Finally, in Sect. 5 we demonstrate the applicability of the proposed methods for the task of learning seismic parameters from raw seismic signals.

2 Preliminaries

We begin by providing a brief description of two methods used in this study: A kernel-based method for dimensionality reduction called *Diffusion Maps* (Coifman and Lafon 2006); and *Dimensionality from Angle and Norm Concentration* (DANCo, Ceruti et al. 2014), which estimates the intrinsic dimension of a manifold based on the ambient high-dimensional data. In the following, vectors and matrices are denoted by bold letters, and their components are denoted by the respective plain letters, indexed using subscripts or parentheses.

2.1 Diffusion maps (DM)

DM (Coifman and Lafon 2006) is a nonlinear dimensionality reduction framework that extracts the intrinsic geometry from a high-dimensional dataset. This framework is based on the construction of a stochastic matrix from the graph of the data. The eigendecomposition of the stochastic matrix provides an efficient representation of the data. Given a high-dimensional dataset $\boldsymbol{X} \in \mathbb{R}^{D\times N}$, the DM framework construction consists of the following steps:

1. A kernel function $\mathcal{K} : \boldsymbol{X} \times \boldsymbol{X} \longrightarrow \mathbb{R}$ is chosen, so as to compute a matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ with elements $K_{i,j} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$, satisfying the following properties: (i) Symmetry: $\boldsymbol{K} = \boldsymbol{K}^T$; (ii) Positive semi-definiteness: $\boldsymbol{K} \succeq \boldsymbol{0}$, namely $\boldsymbol{v}^T \boldsymbol{K} \boldsymbol{v} \geq 0$ for all $\boldsymbol{v} \in \mathbb{R}^N$; and (iii) Non-negativity: $\boldsymbol{K} \geq \boldsymbol{0}$, namely $K_{i,j} \geq 0 \ \forall i, j \in \{1 \dots N\}$. These properties guarantee that $\boldsymbol{K}$ has real-valued eigenvectors and non-negative real-valued eigenvalues.
   In this study, we focus on the common choice of a *Gaussian kernel* (see Eq. 1.1)

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq K_{i,j} = \exp\left(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\epsilon}\right), i, j \in \{1 \dots N\}, \tag{2.1}$$

   as the affinity measure between two multidimensional data vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$; Obviously, choosing the kernel function entails the selection of an appropriate scale ϵ, which determines the degrees of connectivities expressed by the kernel.
2. By normalizing the rows of $\boldsymbol{K}$, the row-stochastic matrix

$$\boldsymbol{P} \triangleq \boldsymbol{D}^{-1}\boldsymbol{K} \in \mathbb{R}^{N\times N} \tag{2.2}$$

   is computed, where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $D_{i,i} = \sum_j K_{i,j}$. $\boldsymbol{P}$ can be interpreted as the transition probabilities of a (fictitious) Markov chain on $\boldsymbol{X}$, such that $\left[(\boldsymbol{P})^t\right]_{i,j} \triangleq p_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$ (where t is an integer power) describes the implied probability of transition from point $\boldsymbol{x}_i$ to point $\boldsymbol{x}_j$ in t steps.
3. Spectral decomposition is applied to $\boldsymbol{P}$, yielding a set of N eigenvalues $\{\lambda_n\}$ (in descending order) and associated normalized eigenvectors $\{\boldsymbol{\psi}_n\}$ satisfying $\boldsymbol{P}\boldsymbol{\psi}_n = \lambda_n \boldsymbol{\psi}_n, n \in \{0 \dots N-1\}$;
4. A new representation for the dataset $\boldsymbol{X}$ is defined by

$$\boldsymbol{\Psi}_\epsilon(\boldsymbol{x}_i) : \boldsymbol{x}_i \longmapsto \left[\lambda_1\psi_1(i), \lambda_2\psi_2(i), \lambda_3\psi_3(i), ..., \lambda_{N-1}\psi_{N-1}(i)\right]^T \in \mathbb{R}^{N-1}, \tag{2.3}$$

   where ϵ is the scale parameter of the Gaussian kernel (Eq. 2.1) and $\psi_m(i)$ denotes the ith element of $\boldsymbol{\psi}_m$. Note that $\lambda_0 = 1$ and $\boldsymbol{\psi}_0 = \boldsymbol{1}$ were excluded as the constant eigenvector $\boldsymbol{\psi}_0 = \boldsymbol{1}$ doesn't carry information about the data.
   The main idea behind this representation is that the Euclidean distance between two multidimensional data points in the new representation is equal to the weighted L_2 distance between the conditional probabilities $p(\boldsymbol{x}_i, :)$ and $p(\boldsymbol{x}_j, :)$, $i, j = 1, ..., N$, where i and j are the i-th and j-th rows of $\boldsymbol{P}$. The diffusion distance is

defined by

$$\mathcal{D}_\epsilon^2(x_i, x_j) = ||\Psi_\epsilon(x_i) - \Psi_\epsilon(x_j)||^2 = \sum_{m\geq 1} \lambda_m(\psi_m(i) - \psi_m(j))^2$$
$$= ||p(x_i, :) - p(x_j, :)||^2_{W^{-1}}, \tag{2.4}$$

where W is a diagonal matrix with $W_{i,i} = \frac{D_{i,i}}{\sum_{i=1}^{M} D_{i,i}}$. This equality is proved in Coifman and Lafon (2006).

5. A low-dimensional mapping $\boldsymbol{\Psi}_\epsilon^d(\boldsymbol{x}_i), i = 1, ..., N$ is defined by

$$\boldsymbol{\Psi}_\epsilon^d(\boldsymbol{x}_i) : X \to \left[\lambda_1\psi_1(i), \lambda_2\psi_2(i), \lambda_3\psi_3(i), ..., \lambda_d\psi_d(i)\right]^T \in \mathbb{R}^d, \tag{2.5}$$

such that $d \ll D$, where $\lambda_{d+1}, ..., \lambda_{N-1} \longrightarrow 0$.

We chose to use DM in our analysis as it provides an intuitive interpretation based on its Markovian construction. Nonetheless, the methods in this manuscript could also be adapted to Laplacian Eigenmaps (Belkin and Niyogi 2001) and to other kernel methods.

2.2 Intrinsic dimension estimation

Given a high-dimensional dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \subseteq \mathbb{R}^{D\times N}$, which describes an ambient space with a manifold $\mathcal{M}$ containing the data points $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$, the *intrinsic dimension* $\bar{d}$ is the minimum number of parameters needed to represent the manifold.

Definition 2.2.1 Let $\mathcal{M}$ be a manifold. The intrinsic dimension $\bar{d}$ of the manifold is a positive integer determined by how many independent "coordinates" are needed to describe $\mathcal{M}$. By using a parametrization to describe the manifold, the intrinsic dimension is the smallest integer $\bar{d}$ such that there exists a smooth map $f(\xi)$ for all data points on the manifold $\mathcal{M} = f(\xi), \xi \subseteq \mathcal{R}^{\bar{d}}$.

Methods proposed by Fukunaga and Olsen (1971) or by Verveer and Duin (1995) use local or global PCA to estimate the intrinsic dimension $\bar{d}$. The dimension is set as the number of eigenvalues greater than some threshold. Others, such as Trunk (1976) or Pettis et al. (1979), use k-neaserst-neighbors (KNN) distances to find a subspace around each point and based on some statistical assumption estimate $\bar{d}$. A survey of different approaches is provided in Camastra (2003). In this study we use Dimensionality from Angle and Norm Concentration (DANCo, by Ceruti et al. (2014)) based algorithm (which we observed to be the most robust approach in our experiments) to estimate $\bar{d}$.

DANCo jointly uses the normalized distances and mutual angles to extract a robust estimate of $\bar{d}$. This is done by finding the dimension that minimizes the Kullback-Leibler divergence between the estimated probability distribution functions (pdf-s) of artificially-generated data and the observed data. A full description of DANCo is presented in the "Appendix" of this manuscript. In Sect. 3.2, we propose a framework which exploits the resulting estimate $\hat{d}$ of $\bar{d}$ for choosing the scale parameter ϵ.

3 Setting the scale parameter ϵ

DM as described in Sect. 2 is an efficient method for dimensionality reduction. The method is almost completely automated, and does not require tuning many hyper-parameters. Nonetheless, its performance is highly dependent on proper choice of ϵ (Eq. 1.1), which, along with the decaying property of Gaussian affinity kernel $\boldsymbol{K}$, defines the affinity between all points in $\boldsymbol{X} \in \mathbb{R}^{D\times N}$. If the ambient dimension D is high, the Euclidean distance becomes meaningless once it takes large values. Thus, a proper choice of ϵ should preserve local connectivities and neglect large distances. We argue that there is no one 'optimal' way for setting the scale parameter; rather, one should define the scale based on the data and on the task in hand.

In the following subsection we describe several existing method for setting ϵ. In Sect. 3.2, we propose a novel algorithm for setting ϵ in the context of manifold learning. Finally in Sect. 3.3, we present three methods for setting ϵ in the context of classification tasks, so as to optimize the classification performance (in certain senses) in the low-dimensional space. Our goal is to optimize the scale prior to the application of the classifier.

3.1 Existing methods

Several studies propose methods for setting the scale parameter ϵ. Some choose ϵ as the empirical squared standard deviation (mean squared Euclidean deviations from the empirical mean) of the data. This approach is reasonable when the data is sampled from a uniform distribution.

A max-min measure is suggested in Lafon et al. (2006) where the scale is set to

$$\epsilon_{\text{MaxMin}} = \mathcal{C} \cdot \max_j [\min_{i, i\neq j} (||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)], i, j = 1, ...N, \tag{3.1}$$

and $\mathcal{C} \in [2, 3]$. This approach attempts to set a small scale to maintain local connectivities.

Another scheme (Singer et al. 2009) aims to find a range of values for ϵ. The idea is to compute the kernel $\boldsymbol{K}$ from Eq. (2.1) at various values of ϵ. Then, search for the range of values which give rise to a well-pronounced Gaussian bell shape. The scheme in Singer et al. (2009) is implemented using Algorithm 3.1.

Algorithm 3.1: ϵ range selection

Input: dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_i \in \mathbb{R}^D$.
Output: Range of values for the scale ϵ, $\bar{\epsilon} = [\epsilon_0, \epsilon_1]$.

1: Compute Gaussian kernels $\boldsymbol{K}(\epsilon)$ for several values of ϵ.
2: Compute: $L(\epsilon) = \sum_i \sum_j K_{i,j}(\epsilon)$ (Eq. 2.1).
3: Plot a logarithmic plot of $L(\epsilon)$ (vs. ϵ).
4: Set $\bar{\epsilon} = [\epsilon_0, \epsilon_1]$ as the maximal linear range of $L(\epsilon)$.

Note that $L(\epsilon)$ consists of two asymptotes, $L(\epsilon) \overset{\epsilon\to 0}{\longrightarrow} \log(N)$ and $L(\epsilon) \overset{\epsilon\to\infty}{\longrightarrow} \log(N^2) = 2\log(N)$, since when $\epsilon \to 0$, $\boldsymbol{K}$ (Eq. 2.1) approaches the Identity matrix, whereas for $\epsilon \to \infty$, $\boldsymbol{K}$ approaches an all-ones matrix. We denote by ϵ_0 the minimal value within the range $\bar{\epsilon}$ (defined in Algorithm 3.1). This value is used in the simulations presented in Sect. 4.

A dynamic scale is proposed in Zelnik-Manor and Perona (2004), suggesting to calculate a local-scale σ_i for each data point $\boldsymbol{x}_i, i = 1, ..., N$. The scale is chosen using the L_1 distance from the r-th nearest neighbor of the point $\boldsymbol{x}_i$. Explicitly, the calculation for each point is

$$\sigma_i = ||\boldsymbol{x}_i - \boldsymbol{x}_r||, i = 1, ..., N, \tag{3.2}$$

where $\boldsymbol{x}_r$ is the r-th nearest (Euclidean) neighbor of the point $\boldsymbol{x}_i$. The value of the kernel for points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \triangleq K_{i,j} = \exp\left(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\sigma_i \sigma_j}\right), i, j \in \{1 \dots N\}. \tag{3.3}$$

This dynamic scale guarantees that at least half of the points are connected to r neighbors.

All the methods mentioned above treat ϵ as a scalar. Thus, when data is sampled from various types of sensors these methods may be dominated by the features (vector elements) with highest energy (or variance). In such cases, each feature $\ell = 1, .., D$, in a data vector $x_i[\ell]$ may require a different scale. In order to re-scale the vector, a diagonal $D \times D$ positive-definite (PD) scaling matrix $\boldsymbol{A} \succ \boldsymbol{0}$ is introduced. The rescaling of the feature vector $\boldsymbol{x}_i$ is set as $\widehat{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i$, $1 \le i \le N$. The kernel matrix is rewritten as

$$\begin{aligned} K_{i,j} &= \mathcal{K}\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) = \exp\left(-\frac{1}{2\epsilon}\|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|^2\right) \\ &= \exp\left(-\frac{1}{2\epsilon}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \boldsymbol{A}^T \boldsymbol{A}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right). \end{aligned} \tag{3.4}$$

A standard way to set the scaling elements $A_{\ell,\ell}$ is to use the empirical standard deviation of the respective elements and then set $\epsilon = \epsilon_{\text{std}} \triangleq 1$. More specifically,

$$A_{\ell,\ell} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i(\ell) - \mu_\ell)^2}, \quad \mu_\ell \triangleq \frac{1}{N}\sum_{i=1}^{N} x_i(\ell) \quad \ell = 1, ..., D, \quad \epsilon = \epsilon_{\text{std}} = 1. \tag{3.5}$$

3.2 Setting ϵ for manifold learning

In this subsection we propose a framework for setting the scale parameter ϵ when the dataset $\boldsymbol{X}$ has some low-dimensional manifold structure $\mathcal{M}$. We start by revisiting

an analysis from (Coifman et al. 2008; Hein and Audibert 2005), which relate the scale parameter ϵ to the intrinsic dimension $\bar{d}$ (Definition 2.2.1) of the manifold. In Coifman et al. (2008) a range of valid values is suggested for ϵ, here we expand the results from (Coifman et al. 2008; Hein and Audibert 2005) by introducing a diagonal PD scaling matrix $\boldsymbol{A}$ (as used in Eq. 3.4). This diagonal matrix enables a feature selection procedure which emphasizes the latent structure of the manifold.

Let $\boldsymbol{K}(\epsilon) \in \mathbb{R}^{N\times N}$, $\boldsymbol{A} \in \mathbb{R}^{D\times D}$ be the kernel matrix and diagonal PD matrix (resp.) from Eq. (3.4). By taking the double sum of all elements in Eq. (3.4), we get

$$S(\epsilon) \triangleq \sum_{i,j} K_{i,j}(\epsilon) = \sum_{i,j} \exp\left(-\frac{1}{2\epsilon}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \boldsymbol{A}^T \boldsymbol{A}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\right). \tag{3.6}$$

By assuming that the data points in $\boldsymbol{X}$ are independently uniformly distributed over the manifold $\mathcal{M}$, this sum can be approximated using the mean value theorem as

$$S(\epsilon) \approx \frac{N^2}{\mathrm{Vol}^2(\mathcal{M})} \int_{\mathcal{M}} \int_{\mathcal{M}} \exp\left(-\frac{1}{2\epsilon}\|\boldsymbol{y}' - \boldsymbol{y}\|^2\right) \mathrm{d}\boldsymbol{y}'\mathrm{d}\boldsymbol{y}, \tag{3.7}$$

where $\mathrm{Vol}(\mathcal{M}) \triangleq \int_{\mathcal{M}} \mathrm{d}\boldsymbol{y}$ is the (weighted) volume of the $\bar{d}$-dimensional manifold $\mathcal{M}$, with $\mathrm{d}\boldsymbol{y}$, $\mathrm{d}\boldsymbol{y}'$ being infinitesimal parallelograms on the manifold, carrying the dependence on $\boldsymbol{A}$ (see Moser 1965 for a more detailed discussion). When ϵ is sufficiently small, the integrand in the internal integral in Eq. (3.7) takes non-negligible values only when $\boldsymbol{y}'$ is close to the hyperplane tangent to the manifold at $\boldsymbol{y}$. Thus, the integration over $\boldsymbol{y}'$ within a small patch around each $\boldsymbol{y}$ can be approximated by integration in $\mathbb{R}^{\bar{d}}$, so that

$$S(\epsilon) \approx \frac{N^2}{\mathrm{Vol}^2(\mathcal{M})} \int_{\mathcal{M}} \int_{\mathbb{R}^{\bar{d}}} \exp\left(-\frac{1}{2\epsilon}\|\boldsymbol{y} - \boldsymbol{t}\|^2\right) \mathrm{d}\boldsymbol{t}\mathrm{d}\boldsymbol{y}, \tag{3.8}$$

where $\bar{d}$ is the intrinsic dimension of $\mathcal{M}$ and $\boldsymbol{t}$ is a $\bar{d}$-dimensional vector of coordinates on the tangent plane.

The integral in Eq. (3.8) has a closed-form solution (the internal integral yields $(2\pi\epsilon)^{\bar{d}/2}$, so that the outer integral yields $(2\pi\epsilon)^{\bar{d}/2}\,\mathrm{Vol}(\mathcal{M})$), and we therefore obtain the relation

$$S(\epsilon) = \sum_{i,j} \exp\left(-r_{i,j}(\boldsymbol{A})/2\epsilon\right) \approx \frac{N^2 (2\pi\epsilon)^{\bar{d}/2}}{\mathrm{Vol}(\mathcal{M})}, \tag{3.9}$$

where we have used $r_{i,j}(\boldsymbol{A}) \triangleq \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \boldsymbol{A}^T \boldsymbol{A}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)$ for shorthand.

The key observation behind our suggested selection of $\boldsymbol{A}$ and ϵ, is that due to the approximation in Eq. (3.9), an "implied" intrinsic dimension $d_\epsilon(\boldsymbol{A})$ can be obtained

for different selections of $\boldsymbol{A}$ and ϵ as follows. Taking the log of Eq. (3.9) we have

$$\log S(\epsilon) = \log\left(\sum_{i,j}\exp\left(-r_{i,j}(\boldsymbol{A})/2\epsilon\right)\right) \approx \frac{\bar{d}}{2}\log(\epsilon) + \log\left(\frac{N^2(2\pi)^{\bar{d}/2}}{\mathrm{Vol}(\mathcal{M})}\right). \tag{3.10}$$

Differentiating w.r.t. ϵ we obtain

$$\frac{\partial \log S(\epsilon)}{\partial \epsilon} = \frac{\sum_{i,j} r_{i,j}(\boldsymbol{A})\exp\left(-r_{i,j}(\boldsymbol{A})/2\epsilon\right)}{2\epsilon^2\sum_{i,j}\exp\left(-r_{i,j}(\boldsymbol{A})/\frac{1}{2\epsilon}\right)} \approx \frac{\bar{d}}{2\epsilon}, \tag{3.11}$$

leading to the "implied" dimension

$$d_\epsilon(\boldsymbol{A}) \approx \frac{\sum_{i,j} r_{i,j}(\boldsymbol{A})\exp\left(-r_{i,j}(\boldsymbol{A})/2\epsilon\right)}{\epsilon\sum_{i,j}\exp\left(-r_{i,j}(\boldsymbol{A})/2\epsilon\right)}. \tag{3.12}$$

We propose to choose the scaling so as to minimize the difference between the estimated dimension $\hat{d}$ (see Sect. 2.2) and the implied dimension $d_\epsilon(\boldsymbol{A})$. We therefore set $\boldsymbol{A}$ (and ϵ) based on solving the following optimization problem

$$\boldsymbol{A} = \arg\min_{\boldsymbol{A},\epsilon} |d_\epsilon(\boldsymbol{A}) - \hat{d}| \quad \text{s.t. } \boldsymbol{A} \text{ is diagonal and PD, } \epsilon > 0. \tag{3.13}$$

We note that this minimization problem has one degree of freedom, which can be resolved, e.g., by arbitrarily setting $A_{1,1} = 1$ (see Algorithm 3.2 below). When working in a sufficently small ambient dimension D, the minimizaion can be solved using an exhaustive search (e.g., on some pre-defined grid of scaling values). However, for large D an exhaustive search may become unfeasible in practice, so we propose a greedy algorithm, outlined below as Algorithm 3.2, for computing both the scaling matrix $\boldsymbol{A}$ and the scale parameter ϵ.

Algorithm 3.2: Manifold Based Vector Scaling

Input: dataset: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \in \mathbb{R}^{D\times N}$.
Intrinsic dimension estimate: $\hat{d}$ (optional).
Output: Normalized dataset $\widehat{\boldsymbol{X}}$
1: **if** isempty($\hat{d}$) **then**
2: Apply DANCo Ceruti et al. (2014) to $\boldsymbol{X}$ to estimate $\hat{d}$ (description in Appendix).
3: **end if**
4: Set $\widehat{\boldsymbol{X}}^{(\hat{d})} = \left(\boldsymbol{X}_{1:\hat{d},:} - \mathrm{mean}(\boldsymbol{X}_{1:\hat{d},:})\right)./\mathrm{std}(\boldsymbol{X}_{1:\hat{d},:})$.
5: **for** $\ell = \hat{d}+1$ **to** D **do**
6: Construct $\widehat{\boldsymbol{X}}^{(\ell)} \triangleq [\widehat{\boldsymbol{X}}^{(\ell-1)}; \boldsymbol{X}_{\ell,:}]$
7: Find $A_{\ell,\ell} > 0$ and $\epsilon > 0$ minimizing $|d_\epsilon(\boldsymbol{A}) - \hat{d}|$, where $\boldsymbol{A} \in \mathbb{R}^{\ell\times\ell}$ is an identity matrix with its (ℓ,ℓ)-th element (only) replaced by $A_{\ell,\ell}$, and where $r_{i,j}(\boldsymbol{A})$ in Eq. (3.12) operates on $\widehat{\boldsymbol{X}}^{(\ell)}$
8: Update $\widehat{\boldsymbol{X}}^{(\ell)} = [\widehat{\boldsymbol{X}}^{(\ell-1)}; \boldsymbol{X}_{\ell,:}\cdot A_{\ell,\ell}]/\sqrt{\epsilon}$
9: **end for**

The proposed algorithm (Algorithm 3.2) operates by iteratively constructing the normalized dataset $\widehat{\boldsymbol{X}} \triangleq \boldsymbol{A}\boldsymbol{X}/\sqrt{\epsilon}$ row by row. To this end, the algorithm is first initialized by normalizing the first $\hat{d}$ rows (coordinates) using their empirical standard deviations (note that the estimated (or known) intrinsic dimension $\hat{d}$ is either provided as an input or estimated using DANCo Ceruti et al. (2014)). Then, in the ℓ-th iteration ($\ell = \hat{d}+1, \ldots D$) only the scaling factor $A_{\ell,\ell}$ for the next (ℓ-th) row of $\boldsymbol{X}$ and a new overall scaling ϵ are found, using a (two-dimensional) exhaustive search. The resulting $A_{\ell,\ell}$ is then applied to the ℓ-th row, and the entire $\ell \times N$ data block is normalized by $\sqrt{\epsilon}$ before the next iteration.

The computational complexity of this algorithm with k hypotheses of ϵ and $A_{l,l}$ for each iteration is $O\left(N^2 k D\right)$, since N^2 operations are required in the computation of a single scaling hypothesis, and this is required for each coordinate $d = 1, ..., D$.

Due to the greedy nature of the algorithm, its performance depends on the order of the D features. We further propose to reorder the features using a soft feature-selection procedure. The studies in Cohen et al. (2002), Lu et al. (2007), Song et al. (2010) suggest an unsupervised feature selection procedure based on PCA. The idea is to use the features which are most correlated with the top principle components. We propose an algorithm for reordering the features based on their correlation with the leading coordinates of the DM embedding. The algorithm (Algorithm 3.3 below) is called Correlation Based Feature Permutation (CBFP), and uses the correlation between the D features and $\hat{d}$ embedding coordinates. This correlation value provides a natural measure for the influence of each feature on the extracted embedding.

Algorithm 3.3: Correlation Based Feature Permutation (CBFP)

Input: dataset: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \in \mathbb{R}^{D \times N}$.
Intrinsic dimension estimate: $\hat{d}$ (optional).
Output: Feature permutation vector $\boldsymbol{j}$, such that $\boldsymbol{j} = \sigma([1, ..., D])$ and $\sigma()$ is a permutation operation.

1: **if** isempty($\hat{d}$) **then**
2: Apply DANCo Ceruti et al. (2014) to $\boldsymbol{X}$ to estimate $\hat{d}$ (described in the Appendix).
3: **end if**
4: Compute ϵ_{MaxMin} based on Eq. (3.1).
5: Using ϵ_{MaxMin} for the kernel scaling, compute DM representation $\boldsymbol{\Psi}^{\hat{d}}$ using Eq. (2.5).
6: Compute a vector $\boldsymbol{c} \in \mathbb{R}^D$ of the feature-embedding correlation scores, defined as

$$c_i \triangleq \sum_{\ell=1}^{\hat{d}} |\text{corr}(\boldsymbol{X}_{i,:}, \boldsymbol{\Psi}^{\hat{d}}_{\ell,:})|, \; i = 1, ..., D, \tag{3.14}$$

where $\text{corr}(\cdot,\cdot)$ denotes the correlation coefficient between its two vector agruments.
7: Set $[\boldsymbol{v}, \boldsymbol{j}] = \text{sort}(\boldsymbol{c})$, where $\boldsymbol{v}$, $\boldsymbol{j}$ are the sorted values and corresponding indices of $\boldsymbol{c}$.
8: Reorder the D features by $\widetilde{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{j}, :)$.

Some remarks on the uniform distribution and finite sample assumptions: The derivation in Eq. (3.7) is based on the assumption that the distribution of $\boldsymbol{X}$ is uniform. If the density is not uniform, consider a measure μ with probability density $\mu(x)$. The integration in Eqs. (3.7) and (3.8) should be with respect to $\mu(x)$. This will change

the result in Eq. (3.9) and in the algorithm that follows. In Hein and Audibert (2005), the authors show that under certain assumptions on $\mu(x)$ the integral in Eq. (3.9) can still be used to approximate the intrinsic dimension $\bar{d}$.

The approximations in Eqs. (3.7) and (3.9) are unbiased for proper choices of N and ϵ. As shown in Hein and Audibert (2005), the bias error in Eq. (3.9) is proportional to $\epsilon^2 C(\mathcal{M})$, where $C(\mathcal{M})$ is the curvature of the manifold. This means that ϵ should be sufficiently small for this term to vanish; However, due to the finite sample approximation of the integral, ϵ should not be too small, since the variance term is proportional to $\frac{1}{N\epsilon^{\bar{d}}}$. This quantity could be used to validate the scaling parameter estimated by Algorithm 3.2. In practice, a proper range of scales ϵ, can be validated by evaluating the slope of $\log S(\epsilon)$ (see Eq. 3.9). If the slope of $\log S(\epsilon)$ as a function of ϵ appears linear, this indicates that the approximation holds, and the bias and the variance errors are negligible.

In Sect. 4.1 below, we evaluate the performance of Algorithms 3.2 and 3.3 when applied to synthetic data embedded in artificial manifolds.

3.3 Setting ϵ for classification

Classification algorithms use a metric space and an induced distance to compute the category of unlabeled data points. Dimensionality reduction is effective for capturing the essential intrinsic geometry of the data and neglecting the undesired information (such as noise). Therefore, dimensionality reduction can drastically improve classification results (Lindenbaum et al. 2015). As previously mentioned, ϵ playes a crucial role in the performance of kernel methods for dimensionality reduction. Various methods have been proposed for finding a scale ϵ that would potentially optimize classification performance.

Studies such as by Gaspar et al. (2012) and by Staelin (2003) use a cross-validation procedure and select the scale ϵ that maximizes the performance on the validation data. In Chapelle et al. (2002) apply gradient descent to a classification-error function to find an 'optimal' scale ϵ. Although these methods share our goal, they require performing classification on a validation set for selecting the scale parameter. To the best of our knowledge, the only method that estimates the scale without using a validation set was proposed by Campbell et al. (1999), where for binary classificationit was suggested to use the scale that maximizes the margin between the support vectors. The authors show empirically that their suggested value correlates with peak classification performance on a validation set.

In this subsection we focus on DM for dimensionality reduction and demonstrate the influence of the scale parameter ϵ on classification performance in the low-dimensional space. The contribution in this section is threefold:

- We use the Davis-Kahan theorem to analyze a perturbed version of ideally separated classes. This allows us to optimize the choice of ϵ merely based on the eigengap of the perturbed kernel.
- Based on our study in Lindenbaum et al. (2015), we present an intuitive geometric metric to evaluate the separation in a multi-class setting. We show empirically that

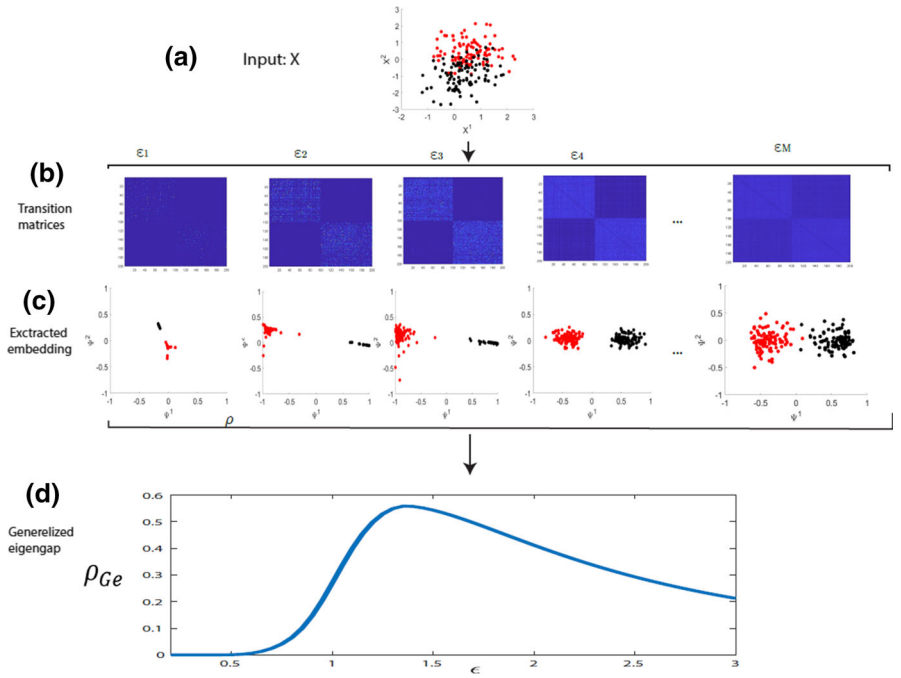


Fig. 1 A schematic of the three proposed methods for estimated ϵ dedicated for classification. Given an input $\boldsymbol{X}$ we use range of hypothesis $\bar{\boldsymbol{\epsilon}}$. **a** The first two dimensions of high-dimensional Gaussian classes. **b** The probabilistic approach- the scale is estimated based on modified transition matrices. **c** The geometric approach- the scale is estimated based on the geometry of the embedding. **d** The spectral approach- the scale is estimated based on a generalized eigengap computed for each scale within the hypothesis range

the scale which maximizes the ratio between class separation and the average class spread also optimizes classification performance.

- Finally, to reduce the computational complexity involved in the spectral decomposition of the affinity kernel, we present a heuristic that allows to estimate the scale parameter based on the stochastic version of the affinity kernel.

Next, we develop tools to estimate a scale parameter based on a given training set. The training set denoted as $\boldsymbol{T} \subset \mathbb{R}^{D\times N}$ consists of N_C classes. The classes are denoted by $\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}$. In this study we focus on the balanced setting, where the number of samples in each class is N_P, thus the total number of data points is $N = N_P N_C$. We use a scalar scaling factor ϵ. However, the analysis provided in this subsection could be expanded to a vector scaling (namely, to the use of a diagonal PD scaling matrix) in a straightforward way (Fig. 1).

3.3.1 The geometric approach

The following approach for setting ϵ is based on the geometry of the extracted embedding. The idea is to extract low-dimensional representations for a range of candidate ϵ's. Then choose ϵ which maximizes the among-classes to within-class variances ratio.

In other words, choose a scale such that the classes are dense and far apart from each other in the resulting low-dimensional embedding space. This is done by maximizing the ratio between the inter-class variance and the sum of intera-class variances. We have explored a similar approach for an audio based classification task in Lindenbaum et al. (2015). This geometric approach is implemented using the following steps:

1. Compute DM-based embeddings $\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n), n = 1, ..., N$, (Eq. 2.5) for various candidate-values of ϵ.
2. Denote by $\boldsymbol{\mu}_i$ the center of mass for class $\boldsymbol{C}_i, i = 1, ..., N_C$, and by $\boldsymbol{\mu}_a$ the center of mass for all the data points. All $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_a$ are computed in the low-dimensional DM-based embedding $\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n)$ - see step 1.
3. For each class $\boldsymbol{C}_i$, the average square distance (in the embedding space) is computed for the N_P data points from the center of mass $\boldsymbol{\mu}_i$ such that

$$D_{c_i} = \frac{1}{N_P} \sum_{\boldsymbol{x}_n \in \boldsymbol{C}_i} ||\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n) - \boldsymbol{\mu}_i||^2, i = 1, ..., N_C. \tag{3.15}$$

4. The same measure is computed for all data points such that

$$D_a = \frac{1}{N} \sum_{\boldsymbol{x}_n \in X} ||\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n) - \boldsymbol{\mu}_a||^2. \tag{3.16}$$

5. Define

$$\rho_{\Psi} \triangleq \frac{D_a}{\sum_{i=1}^{N_C} D_{c_i}}. \tag{3.17}$$

6. Choose ϵ which maximizes ρ_{Ψ}

$$\epsilon_{\rho_{\Psi}} = \arg\max_{\epsilon} \rho_{\Psi}. \tag{3.18}$$

The idea is that $\epsilon_{\rho_{\Psi}}$ (Eq. 3.18) inherits the inner structure of the classes and neglects the mutual structure. In Sect. 4.2, we describe experiments that empirically evaluate the influence of ϵ on the performance of classification algorithms. We note, however, that this approach requires an eigendecomposition computation for each ϵ, thus, its computational complexity is of order of $\mathcal{O}(N^2 d)$ (d being the number of required eigenvectors).

3.3.2 The spectral approach

In this subsection, we analyze the relation between the spectral properties of the kernel and its corresponding low-dimensional representation. We start the analysis by constructing an ideal training set, with well separated classes. Then, we add a small perturbation to the training set and compute the perturbed affinity matrix $\boldsymbol{K}$. Based on the spectral properties of the perturbed kernel we suggest a scaling ϵ to capture the essential information for class separation.

The Ideal Case: We begin the discussion by considering an ideal classification setting, in which the N_C classes are assumed to be well-separated in the ambient space $\boldsymbol{X}$ (a similar setting for spectral clustering was described in Ng et al. (2002)). The separation is formulated using the following definitions:

1. The *Euclidean gap* is defined as

$$D_{\text{Gap}}(\boldsymbol{X}) \triangleq \min_{\substack{\boldsymbol{x}_i \in \boldsymbol{C}_\ell, \boldsymbol{x}_j \in \boldsymbol{C}_m \\ \ell,m=1,...,N_C,\ \ell \neq m}} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2. \tag{3.19}$$

   This is the Euclidean distance between the two closest data points belonging to two different classes.
2. The *maximal class width* is defined as

$$D_{\text{Class}}(\boldsymbol{X}) \triangleq \max_{\substack{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{C}_\ell \\ \ell=1,...,N_C}} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2. \tag{3.20}$$

   This is the maximal Euclidean distance between two data points belonging to the same class.

We assume that $D_{\text{Class}} \ll D_{\text{Gap}}$ such that the classes are well separated. Using this assumption and the decaying property of the Gaussian kernel, the matrix $\boldsymbol{K}$ (Eq. 2.1) converges to the following block form

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}^{(1)} & 0 & ... & 0 \\ 0 & \boldsymbol{K}^{(2)} & ... & 0 \\ : & : & : & : \\ 0 & 0 & ... & \boldsymbol{K}^{(N_C)} \end{bmatrix}, \ \bar{\boldsymbol{P}} = \bar{\boldsymbol{D}}^{-1}\bar{\boldsymbol{K}}, \ \bar{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}^{(1)} & 0 & ... & 0 \\ 0 & \boldsymbol{P}^{(2)} & ... & 0 \\ : & : & : & : \\ 0 & 0 & ... & \boldsymbol{P}^{(N_C)} \end{bmatrix}, \tag{3.21}$$

where $\bar{D}_{i,i} = \sum_j \bar{K}_{i,j}$. For the ideal case, we further assume that the elements of $\boldsymbol{K}^{(i)}$, $i = 1, ..., N_P$, are non-zeros because $\epsilon \sim D_{\text{Class}}$ and the classes are connected.

Proposition 3.3.1 *Assume that $D_{\text{Class}} \ll D_{\text{Gap}}$, then, the matrix $\bar{\boldsymbol{P}}$* (Eq. 3.21) *has an eigenvalue $\lambda = 1$ with multiplicity N_C. Furthermore, the first N_C coordinates of the DM mapping* (Eq. 2.3) *are piecewise constant. The explicit form of the first nontrivial eigenvector $\boldsymbol{\psi}_1$ is given by*

$$\boldsymbol{\psi}_1 = [\underbrace{1, ..., 1}_{N_P \ 1's}, \underbrace{0, ..., 0}_{N-N_P \ 0's}]^T / \sqrt{N_P}.$$

The eigenvectors $\boldsymbol{\psi}_i$, $i = 2, ..., N_C$ have the same structure but cyclically shifted to the right by $(i-1) \cdot N_P$ bins.

Proof Recall that $\bar{\boldsymbol{P}} = \bar{\boldsymbol{D}}^{-1}\bar{\boldsymbol{K}}$ (row-stochastic). Due to the special block structure of $\bar{\boldsymbol{K}}$ (Eq. 3.21), each block $\boldsymbol{P}^{(i)}$, $i = 1, ..., N_P$, is row stochastic. Thus,

$$\boldsymbol{\psi}_i = [\ \underbrace{0, ..., 0}_{(i-1)\cdot N_P\ 0\text{'s}}\ \underbrace{1, ..., 1}_{N_P\ 1\text{'s}},\ \underbrace{0, ..., 0}_{N-i\cdot N_P\ 0\text{'s}}\ ]^T/\sqrt{N_P}.$$

Each eigenvector $\boldsymbol{\psi}_i, i = 1, .., N_p$ consists of a block of 1-s at the row indices that correspond to $\boldsymbol{P}^{(i)}$ (Eq. 3.21), padded with zeros. $\boldsymbol{\psi}_i$ is the right eigenvector ($1 \cdot \boldsymbol{\psi}_i = \bar{\boldsymbol{P}} \cdot \boldsymbol{\psi}_i$), with the eigenvalue $\lambda = 1$. We now have an eigenvalue $\lambda = 1$ with multiplicity N_P and piecewise constant eigenvectors denoted as $\boldsymbol{\psi}_i, i = 1, ..., N_P$. □

Each data point $\boldsymbol{x}_i \in \boldsymbol{C}_\ell,\ \ell = 1, ..., N_C$, corresponds to a row within the respective sub-matrix $\boldsymbol{P}^{(\ell)}$. Therefore, using $\Psi_\epsilon^{N_C}(\boldsymbol{T}) = [1 \cdot \boldsymbol{\psi}_1, ..., 1 \cdot \boldsymbol{\psi}_{N_C}]^T$ as the low-dimensional representation of $\boldsymbol{T}$, all the data points from within a class are mapped to a point in the embedding space.

Corollary 3.3.1 *Using the first N_C eigenvectors of $\bar{\boldsymbol{P}}$* (Eq. 3.21) *as a representation for $\boldsymbol{T}$ such that $\Psi_\epsilon^{N_C}(\boldsymbol{T}) = [1 \cdot \boldsymbol{\psi}_1, ..., 1 \cdot \boldsymbol{\psi}_{N_C}]^T$ yields that the distances $D_{\text{Gap}}(\boldsymbol{\Psi}_\epsilon^{N_C}) = 2$ and $D_{\text{Class}}(\boldsymbol{\Psi}_\epsilon^{N_C}) = 0$ (defined in Eqs.* (3.19) *and* (3.20)*, respectively).*

Proof Based on the representation in Proposition 3.3.1 along with Eq. (3.19), we get

$$\begin{aligned} D_{\text{Gap}}(\boldsymbol{\Psi}_\epsilon^{N_C}) &= \min_{\substack{\boldsymbol{x}_i \in \boldsymbol{C}_\ell, \boldsymbol{x}_j \in \boldsymbol{C}_m \\ \ell, m = 1, ..., N_C,\ \ell \neq m}} ||\boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_i) - \boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_j)||^2 = \sum_{r=1}^{N_C} \lambda_r \cdot (\psi_r(i) - \psi_r(j))^2 \\ &= 1 \cdot (1-0)^2 + 1 \cdot (0-1)^2 + \sum_{r=3}^{N_C} 1 \cdot (0-0)^2 = 2. \end{aligned} \tag{3.22}$$

In a similar manner, we get by Eq. (3.20)

$$\begin{aligned} D_{\text{Class}}(\boldsymbol{\Psi}_\epsilon^{N_C}) &= \max_{\substack{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{C}_\ell \\ \ell = 1, ..., N_C}} ||\boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_i) - \boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_j)||^2 = \sum_{r=1}^{N_C} \lambda_r \cdot (\psi_r(i) - \psi_r(j))^2 \\ &= 1 \cdot (0-0)^2 + 1 \cdot (1-1)^2 + \sum_{r=3}^{N_C} 1 \cdot (0-0)^2 = 0. \end{aligned} \tag{3.23}$$

□

Corollary 3.3.1 implies that we can compute an efficient representation for the N_C classes. We denote this representation by $\bar{\boldsymbol{\Psi}}_\epsilon^{N_C} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_{N_C}]$.

The Perturbed Case In real datasets, we cannot expect that the off block-diagonal elements of the affinity matrix $\boldsymbol{K}$ would be zero. The data points from different classes in real datasets are not completely disconnected, and we can assume they are weakly connected. This low connectivity implies that off-block-diagonal values

of $\boldsymbol{K}$ are non-zeros. We analyze this more realistic scenario by assuming that $\boldsymbol{K}$ is a perturbed version of the "Ideal" block form of $\bar{\boldsymbol{K}}$. Perturbation theory addresses the question of how a small change in a matrix relates to a change in its eigenvalues and eigenvectors. In the perturbed case, the off-block-diagonal terms are non-zeros and the obtained (perturbed) matrix $\widetilde{\boldsymbol{K}}$ takes the form

$$\widetilde{\boldsymbol{K}} = \bar{\boldsymbol{K}} + \widehat{\boldsymbol{W}}, \tag{3.24}$$

where $\widehat{\boldsymbol{W}}$ is assumed to be a symmetrical small perturbation of the form

$$\widehat{\boldsymbol{W}} = \begin{bmatrix} -\boldsymbol{W}^{(1,1)} & \boldsymbol{W}^{(1,2)} & \dots & \boldsymbol{W}^{(1,N_C)} \\ \boldsymbol{W}^{(2,1)} & -\boldsymbol{W}^{(2,2)} & \dots & \boldsymbol{W}^{(2,N_C)} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{W}^{(N_C,1)} & \boldsymbol{W}^{(N_C,2)} & \dots & -\boldsymbol{W}^{(N_C,N_C)} \end{bmatrix}, \boldsymbol{W}^{(\ell,m)} = \boldsymbol{W}^{(m,\ell)}, \quad \ell, m = 1, ..., N_C. \tag{3.25}$$

The analysis of the "Ideal case" has provided an efficient representation for classification tasks as described in Proposition 3.3.1. We propose to choose the scale parameter ϵ such that the extracted representation based on $\bar{\boldsymbol{K}}$ (Eq. 3.21) is similar to the extracted representation using $\tilde{\boldsymbol{K}}$ (Eq. 3.24). For this purpose we use the following theorem.

Theorem 3.1 (Davis-Kahan) Stewart (1990) *Let $\bar{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ be Hermitian matrices of the same dimensions, and let $\widetilde{\boldsymbol{A}} \triangleq \bar{\boldsymbol{A}} + \widehat{\boldsymbol{B}}$ be a perturbed version of $\bar{\boldsymbol{A}}$. Set an interval S, denote the eigenvalues within S as $\lambda_S(\bar{\boldsymbol{A}})$ and $\lambda_S(\widetilde{\boldsymbol{A}})$ with a corresponding set of eigenvectors $\bar{\boldsymbol{V}}_1$ and $\widetilde{\boldsymbol{V}}_1$ for $\bar{\boldsymbol{A}}$ and $\widetilde{\boldsymbol{A}}$, respectively. Define δ as*

$$\delta \triangleq \min\{|\lambda(\widetilde{\boldsymbol{A}}) - s|; \lambda(\widetilde{\boldsymbol{A}}) \notin S, s \in S\}. \tag{3.26}$$

Then the distance

$$d(\bar{\boldsymbol{V}}_1, \widetilde{\boldsymbol{V}}_1) \triangleq \left\|\sin\Theta\left(\bar{\boldsymbol{V}}_1, \widetilde{\boldsymbol{V}}_1\right)\right\|_F \leq \frac{1}{\delta}\left\|\widehat{\boldsymbol{B}}\right\|_F, \tag{3.27}$$

where $\Theta\left(\bar{\boldsymbol{V}}_1, \widetilde{\boldsymbol{V}}_1\right)$ is a diagonal matrix with the principal angles on the diagonal, and $\|\cdot\|_F$ denotes the Frobenius norm.

In other words, the theorem states that the eigenspace spanned by the perturbed kernel $\widetilde{\boldsymbol{K}}$ is similar, to some extent, to the eigenspace spanned by the ideal kernel $\bar{\boldsymbol{K}}$. The distance between these eigenspaces is bounded by $\frac{1}{\delta}\|\widehat{\boldsymbol{W}}\|_F$. Theorem 3.2 provides a measure which helps to minimize the distance between the ideal representation $\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}$ (proposition 3.3.1) and the realistic (perturbed) representation $\widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C}$.

Theorem 3.2 *The distance between $\bar{\boldsymbol{\Psi}}_\epsilon^{N_C} \in \mathbb{R}^{N_C}$ and $\widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C} \in \mathbb{R}^{N_C}$ in the DM representations based on the matrices $\bar{\boldsymbol{P}}$ and $\widetilde{\boldsymbol{P}}$, respectively, is bounded such that*

$$d\left(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C}\right) = \left\|\sin\Theta\left(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C}\right)\right\|_F \leq \frac{||\widehat{\boldsymbol{W}}||_F ||\bar{\boldsymbol{D}}^{-1/2}||_F^2}{\widetilde{\lambda}_{N_C} - \widetilde{\lambda}_{N_C+1}}, \tag{3.28}$$

where $\widehat{\boldsymbol{W}}$ is the perturbation matrix defined in Eq. (3.24) and $\bar{\boldsymbol{D}}$ is a diagonal matrix whose elements are the sums of rows $\bar{D}_{i,i} = \sum_j \bar{K}_{i,j}$.

Proof Define $\bar{\boldsymbol{A}} \triangleq \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{K}}\bar{\boldsymbol{D}}^{-1/2} = \bar{\boldsymbol{D}}^{1/2}\bar{\boldsymbol{P}}\bar{\boldsymbol{D}}^{-1/2}$. Based on Eq. (3.24), we have

$$\widetilde{\boldsymbol{A}} = \bar{\boldsymbol{A}} + \bar{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{W}}\bar{\boldsymbol{D}}^{-1/2}. \tag{3.29}$$

We are now ready to use Theorem 3.1. Assume that the eigenvalues $\{\bar{\lambda}_i\}$ of $\bar{\boldsymbol{A}}$ and $\{\widetilde{\lambda}_i\}$ of $\widetilde{\boldsymbol{A}}$ are ordered in descending order, and set $S = [\lambda(\widetilde{\boldsymbol{A}})_{N_C}, 1]$, denoting the first N_C eigenvectors of $\bar{\boldsymbol{A}}$ and of $\widetilde{\boldsymbol{A}}$ as $\bar{\boldsymbol{V}}_1$ and $\widetilde{\boldsymbol{V}}_1$, respectively. Obviously, by construction we have $\widetilde{\lambda}_i \in S$, $i = 1, ..., N_C$. Based on the analysis of the "ideal" matrix $\bar{\boldsymbol{P}}$, we know that its first N_C eigenvalues are equal to 1. Noting that $\bar{\boldsymbol{A}}$ is algebraically similar to $\bar{\boldsymbol{P}}$, they have the same eigenvalues, implying that $\bar{\lambda}_i \in S$, $i = 1..., N_C$, as well. Using the definition of δ from Eq. (3.26), we conclude that $\delta = \widetilde{\lambda}_{N_C} - \widetilde{\lambda}_{N_C+1}$. Setting $\bar{\boldsymbol{A}} \triangleq \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{K}}\bar{\boldsymbol{D}}^{-1/2}$ and $\widehat{\boldsymbol{B}} = \bar{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{W}}\bar{\boldsymbol{D}}^{-1/2}$, the Davis-Kahan Theorem 3.1 asserts that the distance between the eigenspaces $\bar{\boldsymbol{V}}_1$ and $\widetilde{\boldsymbol{V}}_1$ is bounded such that

$$d\left(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C}\right) = \left\|\sin\Theta\left(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C}\right)\right\|_F \leq \frac{||\widehat{\boldsymbol{W}}||_F||\bar{\boldsymbol{D}}^{-1/2}||_F^2}{\widetilde{\lambda}_{N_C} - \widetilde{\lambda}_{N_C+1}}, \tag{3.30}$$

The eigen-decomposition of $\bar{\boldsymbol{A}}$ is written as $\bar{\boldsymbol{A}} = \bar{\boldsymbol{V}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^T$. Note that $\bar{\boldsymbol{P}} = \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{A}}\bar{\boldsymbol{D}}^{1/2}$ which means that the eigen-decomposition of $\bar{\boldsymbol{P}}$ could be written as $\bar{\boldsymbol{P}} = \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{V}}\bar{\boldsymbol{\Sigma}}\bar{\boldsymbol{V}}^T\bar{\boldsymbol{D}}^{1/2}$ and the right eigenvectors of $\bar{\boldsymbol{P}}$ are $\bar{\boldsymbol{\Psi}} = \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{V}}$. Using the same argument for $\widetilde{\boldsymbol{A}}$ and choosing the eigenspaces using the first N_C eigenvectors, we get that decreasing the term $d(\bar{\boldsymbol{V}}_1, \widetilde{\boldsymbol{V}}_1)$ also decreases $d(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \widetilde{\boldsymbol{\Psi}}_\epsilon^{N_C})$. □

Assumption 1 The perturbation matrix $\widehat{\boldsymbol{W}}$ (Eq. 3.24) changes only slightly over the range of values of $\epsilon \in (D_{\text{Class}}, D_{\text{Gap}})$.

Explanation For two data points $\boldsymbol{x}_i, \boldsymbol{x}_j$ from different classes $\boldsymbol{x}_i \in C_\ell, \boldsymbol{x}_j \in C_m$, $\ell \neq m$ and $\epsilon \sim D_{\text{Class}}$, the values of $\widehat{\boldsymbol{W}}_{i,j} \ll 1$. The decaying property of the Gaussian kernel provides a range of values for $\epsilon \sim D_{\text{Class}}$ such that the perturbation matrix $\widehat{\boldsymbol{W}}$ is indeed small. In Sect. 4.2 below we evaluate this assumption using a mixture of Gaussians.

Corollary 3.3.2 *Given N_C classes under the perturbation assumption and assumption 1, the generalized eigengap is defined as $Ge = |\widetilde{\lambda}_{N_C} - \widetilde{\lambda}_{N_C+1}|$. The scale parameter ϵ, which maximizes Ge*

$$\epsilon_{Ge} = \arg\max_\epsilon(Ge) = \arg\max_\epsilon(\widetilde{\lambda}_{N_C} - \widetilde{\lambda}_{N_C+1}) \tag{3.31}$$

provides the best class separation using an N_C coordinates embedding ($\widetilde{\boldsymbol{\Psi}}_{\epsilon}^{N_C}$).

This approach also requires computing an eigendecomposition for each ϵ value, thus, its computational complexity is of the order of $\mathcal{O}(N^2 N_C)$.

3.3.3 The probabilistic approach

We introduce here notations from graph theory to compute a measure of the class separation based on the stochastic matrix $\boldsymbol{P}$ (Eq. 2.2). Based on the values of the matrix $\boldsymbol{P}$, a *Cut* (Shi and Malik 2000) is defined for any two subsets $\boldsymbol{A}, \boldsymbol{B} \subset \boldsymbol{T}$

$$\text{cut}(\boldsymbol{A}, \boldsymbol{B}) = \sum_{\boldsymbol{x}_i \in \boldsymbol{A}, \boldsymbol{x}_j \in \boldsymbol{B}} P_{i,j}. \tag{3.32}$$

Given N_C classes $\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3, ..., \boldsymbol{C}_{N_C} \subset \boldsymbol{T}$, we define the *Classification Cut* by the following measure

$$\text{Ccut}(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}) = \frac{\sum_{\ell=1}^{N_C} \text{cut}(\boldsymbol{C}_\ell, \boldsymbol{C}_\ell)}{N}. \tag{3.33}$$

In clustering problems, a partition is searched such that the normalized version of the cut is minimized (Dhillon et al. 2004; Ding et al. 2005). We use this intuition for a more relaxed classification problem.

We first define a *Generalized cut* using the following matrix

$$\widehat{P}_{i,j} \triangleq \begin{cases} P_{i,j}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}. \tag{3.34}$$

Based on $\widehat{\boldsymbol{P}}$ (which carries the dependence on ϵ), the Generalized cut is then defined as

$$\text{Gcut}(\boldsymbol{A}, \boldsymbol{B}) \triangleq \sum_{\boldsymbol{x}_i \in \boldsymbol{A}, \boldsymbol{x}_j \in \boldsymbol{B}} \widehat{P}_{i,j}. \tag{3.35}$$

The idea is to remove the probability of "staying" at a specific node from the within-class transition probability. Now let

$$\rho_P \triangleq \text{GCcut}(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}) \triangleq \frac{1}{N} \sum_{\ell=1}^{N_C} \text{Gcut}(\boldsymbol{C}_\ell, \boldsymbol{C}_\ell). \tag{3.36}$$

We search for ϵ which maximizes ρ_P, namely

$$\epsilon_{\rho_P} = \arg\max_{\epsilon}(\rho_P). \tag{3.37}$$

By the stochastic model, the implied probability of transition between point $\boldsymbol{x}_i$ and point $\boldsymbol{x}_j$ is equal to $p(\boldsymbol{x}_i, \boldsymbol{x}_j) = P_{i,j}$, therefore by maximizing ρ_P, the sum of within-class transition probabilities is maximized. Based on the definition of the diffusion

distance (Eq. 2.4), this implies that the within-class diffusion distance would be small, followed by a small Euclidean distance in the DM space. The heuristic approach entailed in Eq. (3.37) provides yet another criterion for setting a scale parameter which captures the geometry of the given classes. This approach does not require computing an eigendecomposition for each candidate ϵ value, thus, its computational complexity is of order of $\mathcal{O}(N^2)$.

4 Experimental results

In this section we provide some experimental results, showing and comparing the different approaches (outlined in the previous section) in their respective contexts. We begin by demonstrating our proposed manifold-based scaling in Sect. 4.1, and then demonstrate the classification-based scaling approaches in Sect. 4.2.

4.1 Manifold learning

In this subsection we evaluate the performance of the proposed manifold-based approach by embedding a low-dimensional manifold which lies in a high-dimensional space. We consider two datasets: A synthetic set and a set based on an image taken from the MNIST database.

4.1.1 A synthetic dataset

The first experiment is constructed by projecting a 3-dimensional synthetic manifold into a high-dimensional space, then concatenating it with Gaussian noise. Data generation is done according to the following steps:

- First, a 3-dimensional Swiss Roll is constructed based on the following function

$$\boldsymbol{y}_i = \begin{bmatrix} y_i(1) \\ y_i(2) \\ y_i(3) \end{bmatrix} = \begin{bmatrix} 6\theta_i \cos(\theta_i) \\ h_i \\ 6\theta_i \sin(\theta_i) \end{bmatrix}, i = 1, ..., N, \tag{4.1}$$

  where θ_i, h_i $(i = 1, ..., N)$, are drawn from Uniform distributions within the intervals $[\frac{3\cdot\pi}{2}, \frac{9\cdot\pi}{2}]$, $[0, 100]$, respectively. In our experiment we chose $N = 2000$.
- We project the Swiss roll into a high-dimensional space by multiplying the data by a random matrix $\boldsymbol{N}_T \in R^{D_1 \times 3}$, $D_1 > 3$. The elements of $\boldsymbol{N}_T$ are drawn from a Gaussian distribution with zero mean and variance of σ_T^2.
- Finally, we augment the projected Swiss Roll with a vector of Gaussian noise, obtaining

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{N}_T \cdot \boldsymbol{y}_i \\ \boldsymbol{n}_i^1 \end{bmatrix}, \quad i = 1, ..., N, \tag{4.2}$$

  where each component of $\boldsymbol{n}_i^1 \in \mathbb{R}^{D_2}$, $i = 1, ..., 2000$, is an independent Gaussian variable with zero mean and variance of σ_N^2.

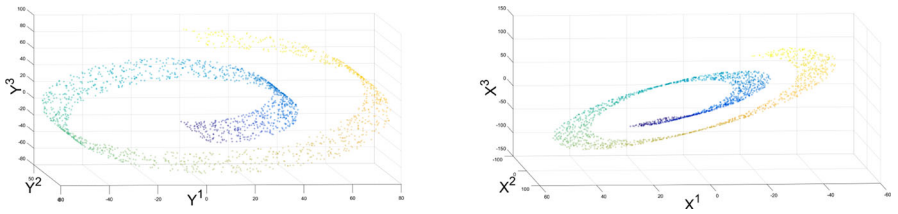


Fig. 2 Left: "clean" Swiss Roll ($\boldsymbol{Y}$ in Eq. (4.1). Right: 3 coordinates of the projected Swiss Roll ($\boldsymbol{X}$ in Eq. (4.2)). Both figures are colored by the value of the underlying parameter θ_i, $i = 1, ..., 2000$ (Eq. 4.1)

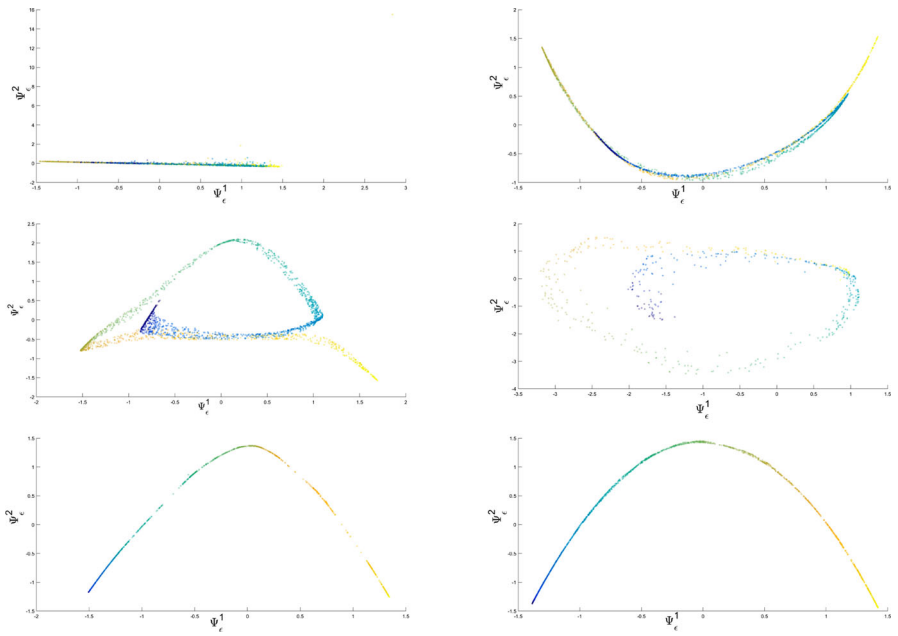


Fig. 3 Extracted DM-based embedding of the "noisy" Swiss roll using different methods for choosing the scale parameter ϵ. Top left: standard deviation scalings, the matrix A and scaling ϵ_{std} are computed by Eq. (3.5). Top right: the ϵ_0 scaling, the calculation of ϵ_0 is described in Alg. (3.1) and Singer et al. (2009). Mid left: the MaxMin scaling, the value ϵ_{MaxMin} is defined by Eq. (3.1). Mid right: KNN based scaling (Zelnik-Manor and Perona 2004). Bottom left: the proposed scaling $\hat{\epsilon}$, $\hat{A}$ which is described in Alg. (3.2). Bottom right: scaling based on ϵ_0 as described in Algorithm (3.1) and Singer et al. (2009) applied to the clean Swiss roll Y that is defined by Eq. (4.1)

We define the datasets $\boldsymbol{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_N] \in \mathbb{R}^{3\times N}$ and $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_N] \in \mathbb{R}^{(D_1+D_2)\times N}$

To evaluate the proposed framework, we apply Algorithm 3.1 followed by Algorithm 3.2, and extract a low-dimensional embedding (Fig. 2).

Different high-dimensional datasets $\boldsymbol{X}$ were generated using various values of σ_N, σ_{N_T}, D_1 and D_2. DM is applied to each $\boldsymbol{X}$ using:

- The standard deviation normalization as defined in Eq. (3.5).
- The ϵ_0 scale, which is described in 3.1 and in Singer et al. (2009).
- The MaxMin scale, as defined in Eq. (3.1) and in Lafon et al. (2006).

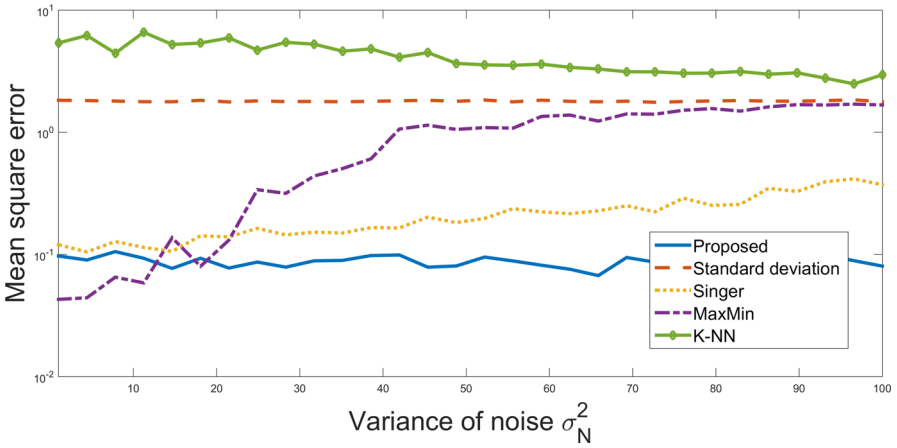


Fig. 4 The mean square error of the extracted embedding. A comparison between the proposed normalization and alternative methods which are detailed in Sect. 3

- The KNN based scaling (Zelnik-Manor and Perona 2004).
- The proposed scale parameters $\boldsymbol{A}, \epsilon$, obtained using Algorithm 3.2.

The extracted embedding is compared to the embedding extracted from the clean Swiss roll $\boldsymbol{Y}$ defined in Eq. (4.1).

Each embedding is computed using an eigendecomposition, therefore, the embedding's coordinates could be the same up to scaling and rotation. To overcome this ambiguity, we search for an optimal translation and rotation matrix of the following form

$$\bar{\boldsymbol{\Psi}}_\epsilon(\boldsymbol{X}) = \boldsymbol{R} \cdot \boldsymbol{\Psi}_\epsilon(\boldsymbol{X}) + \boldsymbol{T}, \tag{4.3}$$

where $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{T}$ is the translation matrix, which minimizes the mis-match error

$$\text{err} = ||\bar{\boldsymbol{\Psi}}_\epsilon(\boldsymbol{X}) - \boldsymbol{\Psi}_\epsilon(\boldsymbol{Y})||_F^2, \tag{4.4}$$

defined as the sum of square distances between values of the clean mapping $\boldsymbol{\Psi}_\epsilon(\boldsymbol{Y})$ and the "aligned" mapping $\bar{\boldsymbol{\Psi}}_\epsilon(\boldsymbol{X})$. We repeat the experiment 40 times and compute the empirical Mean Square error in the embedding space defined as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\Psi}_\epsilon(\boldsymbol{y}_i) - \bar{\boldsymbol{\Psi}}_\epsilon(\boldsymbol{x}_i))^2. \tag{4.5}$$

An example of the extracted embedding based on all the different methods is presented in Fig. 3, followed by the MSE in Fig. 4. It is evident that Algorithm 3.2 is able to extract a more precise embedding than the alternative scaling schemes. The strength of Algorithm 3.2 is that it emphasizes the coordinates which are essential for the embedding and neglects the coordinates which were contaminated by noise.

4.1.2 MNIST manifold

In the following experiment, we create an artificial low-dimensional manifold by rotating a handwritten image of a digit. First, we rotate the handwritten digit '6' from MNIST dataset by $N = 320$ angles that are uniformly sampled over $[0, 2\pi]$. Next, we add random zero-mean Gaussian noise with a variance of σ_N^2 independently to each pixel. An example of the original and noisy version of the handwritten '6' are shown in Fig. 5. Note that the values of the original image are in the range $[0, 1]$. In order to capture the circular structure of the manifold we apply DM to the rotated images. An example of the expected circular structure extracted by DM is depicted in Fig. 5.

We apply the different scaling schemes to the noisy images and extract a 2-dimensional DM-based embedding. For the vectorized scaling schemes (proposed and standard deviation approach) we apply the scalings to the top 50 principle components. This allows us to reduce the computational complexity, as the dimension of the feature space is reduced from 784 to 50. In this experiment we further compare to two additional local scaling schemes. The first (Vasiloglou et al. 2006), which we refer to as Harmonic, and the second, presented in Taseska et al. (2019), which we refer to as LocalDM.

To evaluate the performance of the different scaling schemes, we propose the following metric to compare the extracted embedding to a perfect circle. Given a 2-dimensional representation $\boldsymbol{\Psi}_\epsilon(\boldsymbol{X})$, we use a polar transformation to evaluate the implied radius at each point. The squared radius is defined by $r_i^2 = (\boldsymbol{\Psi}_\epsilon^1(\boldsymbol{x}_i))^2 + (\boldsymbol{\Psi}_\epsilon^2(\boldsymbol{x}_i))^2$. Next, we normalize the radius values by their empirical mean, such that $\hat{r}_i = \frac{r_i}{\mu_r}$, and μ_r is the empirical mean of the radii $\mu_r = \sum_i \frac{r_i}{N}$. Finally, we compute the empirical variance of the normalized radius $\hat{r}$. Explicitly, this value is computed by $\sigma_{\hat{r}}^2 = \frac{\sum_i (\hat{r}_i - 1)^2}{N}$. A scatter plot of $\sigma_{\hat{r}}^2$ vs. the variance of the additive noise σ_N^2 is presented in Fig. 6. As evident in this figure, up to a certain variance of the noise, the proposed scaling scheme suppresses the noise and captures the correct circular structure of the data. At some level of noise our method breaks. It seems that the standard deviation and Singer's approach also break at a similar noise level. An explanation for this phenomenon could be that at the lower SNRs all these methods start to "amplify" the noise, rather than the signal.

4.2 Classification

In this subsection we provide empirical support for the theoretical analysis from Sect. (3.3). We evaluate the influence of ϵ on the classification results using four datasets: a mixture of Gaussians, artificial classes lying on a manifold, handwritten digits and seismic recordings. We focus on evaluating how the proposed measures ρ_P, $\rho_{\boldsymbol{\Psi}}$, Ge (Eqs. (3.37), (3.18), (3.31), resp.) are correlated with the quality of the classification.

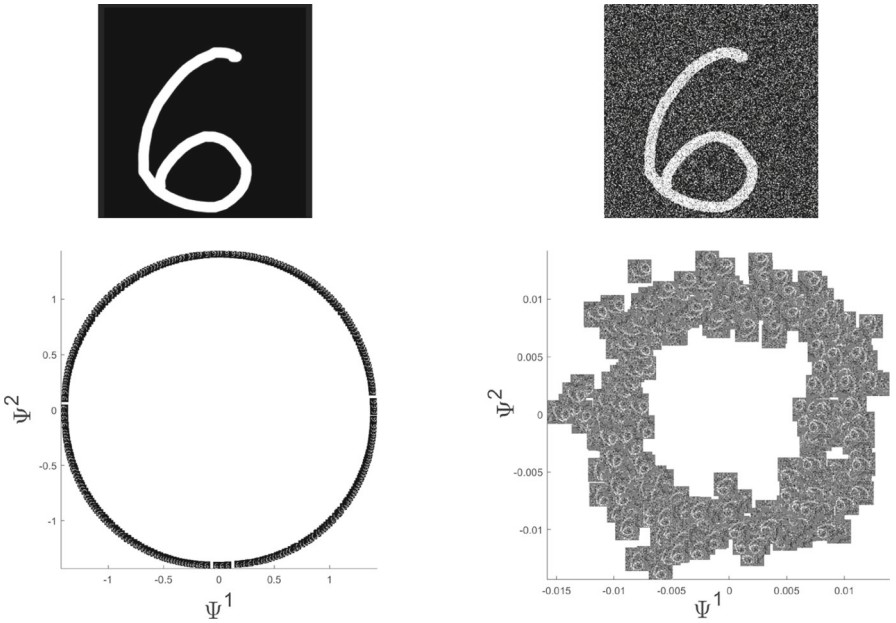


Fig. 5 Top left: an example of a clean handwritten digit of '6'. Top right: a noisy example of the digit '6'. Each pixel is added by a Gaussian noise. The noise is i.i.d. drawn from $N(0, 0.5)$. Bottom left: extracted DM-based embedding from 320 rotated images of the "clean" handwritten digit. Bottom right: extracted DM-based embedding from 320 rotated images of the "noisy" handwritten digit

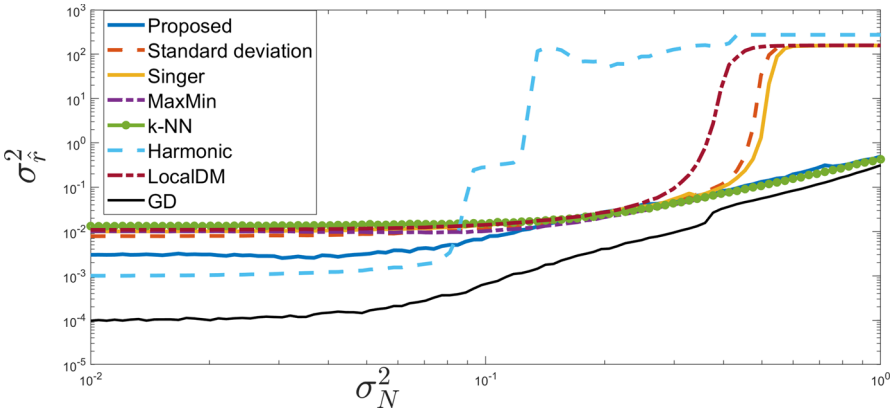


Fig. 6 The normalized radius variance (NRV) of the extracted embedding from the noisy rotated digit manifold. A comparison between the proposed normalization and alternative methods which are detailed in Sect. 3

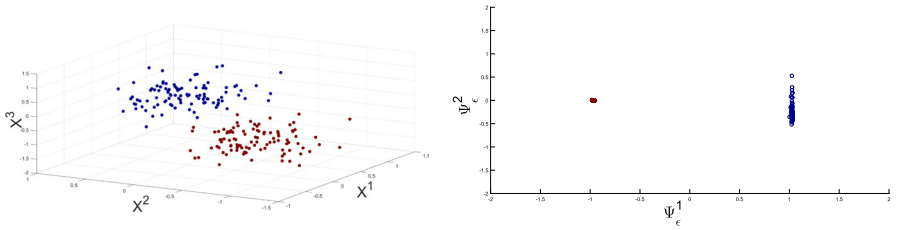


Fig. 7 Left: an example of the Gaussian distributed data points. Right: a 2-dimensional mapping of the data points

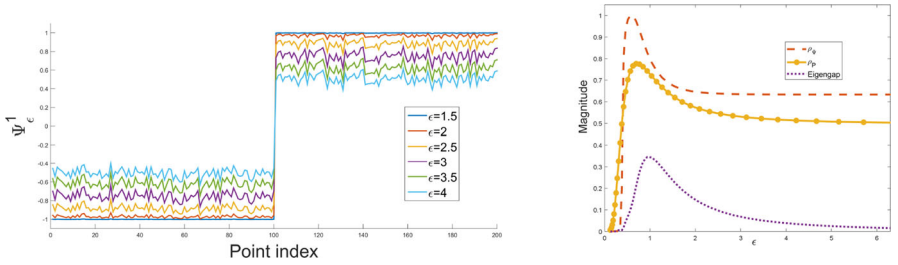


Fig. 8 Left: the first eigenvector $\boldsymbol{\Psi}^1$ computed for various values of ϵ. Right: a comparison between ρ_P, ρ_Ψ and Ge

4.2.1 Classification of a Gaussian mixture

In the following experiment we focus on a simple classification test using a mixture of Gaussians. We generate two classes using two Gaussians, based on the following steps:

1. Two vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2 \in \mathbb{R}^6$ were drawn from a Gaussian distribution $N(0, \sigma_M \cdot \boldsymbol{I}_{6\times 6})$. These vectors are the centers of masses for the generated classes $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. (resp.).
2. $N = 100$ data points were drawn for each class $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ with a Gaussian distribution $N(\boldsymbol{\mu}_1, \sigma_V \cdot \boldsymbol{I}_{6\times 6})$ and $N(\boldsymbol{\mu}_2, \sigma_V \cdot \boldsymbol{I}_{6\times 6})$, respectively. Denote these $2N$ data points by $\boldsymbol{C}_1 \cup \boldsymbol{C}_2 = \boldsymbol{T} \subset \mathbb{R}^{6\times 200}$.

The first experiment evaluates the Spectral Approach (Sect. 3.3.2). Therefore, we set $\sigma_v < \sigma_M$ such that the class variance is smaller than the variance of the center of mass. Then, we apply DM using a scale parameter ϵ such that $\epsilon \sim \sigma_v^2 < \sigma_M^2$. In Fig. 8 (left), we present the first extracted diffusion coordinate using various values of ϵ. It is evident that the separation between classes is highly influenced by ϵ. A comparison between ρ_P,ρ_Ψ and Ge is presented in Fig. 8 (right). This comparison provides evidence of the high correlation between ρ_P (Eq. 3.37), ρ_Ψ (Eq. 3.18) and the generalized eigengap (Eq. 3.31) (Fig. 7).

To evaluate the validity of Assumption 1, we calculate the Frobenius norm of the perturbation matrix $\widehat{W}$ for various values of ϵ. The results with the approximated ϵ_{Ge} are presented in Fig. 9. Indeed, as evident from Fig. 9, the value of $||\widehat{W}||_F$ is nearly constant for a small range of values around ϵ_{Ge}.

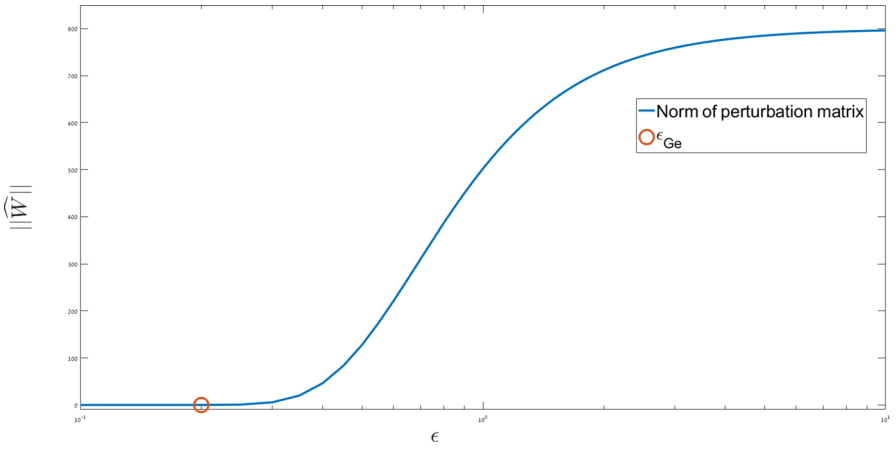


Fig. 9 The Frobenius norm of the perturbation matrix $\widehat{W}$. The annotated point is the approximated scale ϵ_{Ge}

4.2.2 Classes based on an artificial physical process

For the non-ideal case, we generate classes using a non-linear function. This non-linear function is designed to model an unknown underlying nonlinear physical process governed by a small number of parameters. Consequently, the classification task is essentially expected to provide an estimate of these hidden parameters. An example for such a problem is studied, e.g., in Lindenbaum et al. (2015), where a musical key is estimated by applying a classifier to a low-dimensional representation extracted from the raw audio signals. In the following steps we describe how we generate classes from a Spiral structure:

1. Set the number of classes N_C and a gap parameter G. Each class $\boldsymbol{C}_\ell, \ell = 1, ..., N_C$, consists of N_P data points drawn from a uniformly dense distribution within the line $[(\ell - 1) \cdot L_C, \ell \cdot L_C - G], \ell = 1, ..., N_C$. L_C is the class-length, set as $L_C = \frac{1}{N_C}$. Let $N = N_C N_P$ denote the total number of points.
2. Denote $\{r_i\}_{i=1}^N$ as the set of all points from all classes.
3. Project each r_i into the ambient space using the following spiral-like function

$$\bar{\boldsymbol{x}}_i = \begin{bmatrix} \bar{x}_i(1) \\ \bar{x}_i(2) \\ \bar{x}_i(3) \end{bmatrix} = \begin{bmatrix} (6\pi r_i)\cos(6\pi r_i) \\ (6\pi r_i)\sin(6\pi r_i) \\ r_i^3 - r_i^2 \end{bmatrix} + \boldsymbol{n}_i^2, \tag{4.6}$$

where $\boldsymbol{n}_i^2 \in \mathbb{R}^3$ are drawn independently from a zero-mean Gaussian distribution with covariance $\boldsymbol{\Lambda} = \sigma_S \cdot \boldsymbol{I}_3$. Two examples of the spiral-based classes are shown in Fig. 10. For both examples, we use $N_C = 4, N_P = 100, \sigma_S = 0.4$ with different values for the gap parameter G.

To evaluate the advantage of the proposed scale parameters ϵ_Ψ and ϵ_P (Eqs. (3.18) and (3.37), resp.) for classification tasks, we calculate the ratios ρ_P and ρ_Ψ for various

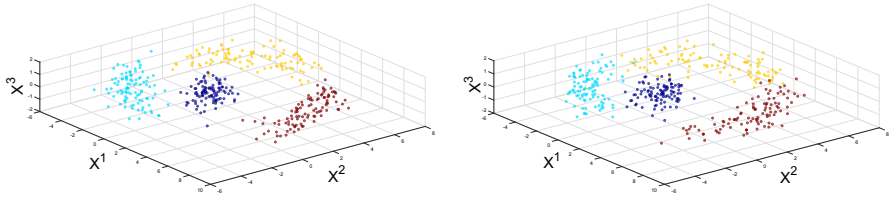


Fig. 10 Two examples of the generated three-dimensional spiral that are based on Eq. (4.6) using $N_C = 4$ classes with $N_P = 100$ data points within each class. The gaps are set to be $G = 0.02, 0.04$ left and right, respectively

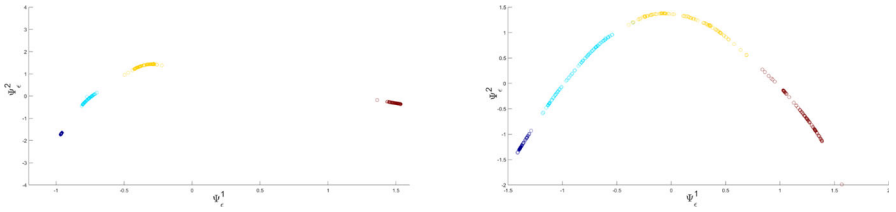


Fig. 11 A 2-dimensional mapping extracted from both spirals presented in Fig. 10

values of ϵ, and then we evaluate the resulting classification (which is based on the low-dimensional embedding). Examples of embeddings of the two spirals from Fig. 10 are shown in Fig. 11. This merely demonstrates the effect of ϵ on the quality of separation.

We apply classification in the low-dimensional space using KNN ($k = 1$). The KNN classifier is evaluated based on Leave-One-Out cross validation. The results are shown in Fig. 12, where it is evident that the classification results in the ambient space are highly influenced by the scale parameter ϵ. Furthermore, peak classification results occur at a value of ϵ corresponding to the maximal values of ρ_{Ge} and ρ_{Ψ}. The value of ρ_P did not indicate the peak classification scale, however, its computation complexity is lighter compared to ρ_{Ge} and ρ_{Ψ}.

4.2.3 Classification of handwritten digits

In the following experiment, we use the dataset from the UCI machine learning repository (Lichman 2013). The dataset consists of 2000 data points describing 200 instances of each digit from 0 to 9, extracted from a collection of Dutch utility maps. The dataset consists of multiple features of different dimensions. We use a concatenation of the Zerkine moment (ZER), morphological (MOR), profile correlations (FAC) and the Karhunen-loéve coefficients (KAR) as our features space.

We compute the proposed ratios ρ_P and ρ_{Ψ} for various values of ϵ, and estimate the optimal scale based on Eqs. (3.18), (3.37). We evaluate the extracted embedding using 20-fold cross validation (5% left out as a test set). The classification is done by applying KNN (with $k = 1$) in the d-dimensional embedding. In Fig. 13, we present the classification results and the proposed optimal scales ϵ for classification. Our proposed scale concurs with the scale that provides maximal classification rate.

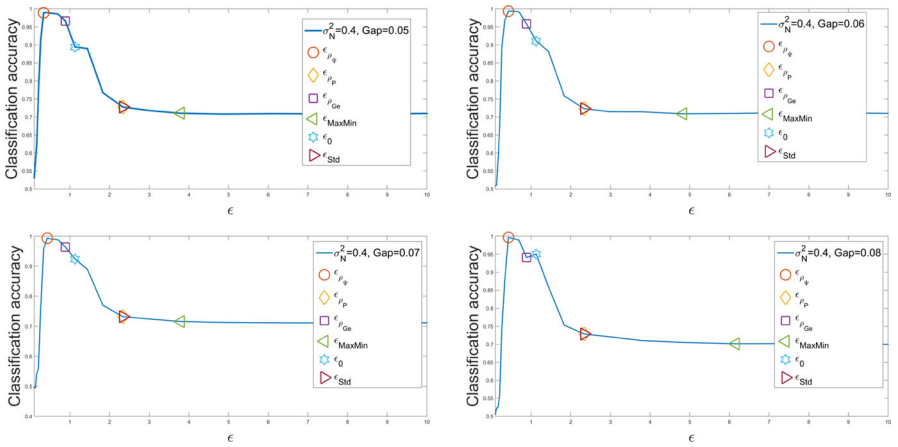


Fig. 12 Accuracy of classification in the spiral artificial dataset for different values of the gap parameter G. The data is generated based on Eq. (4.6). The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots

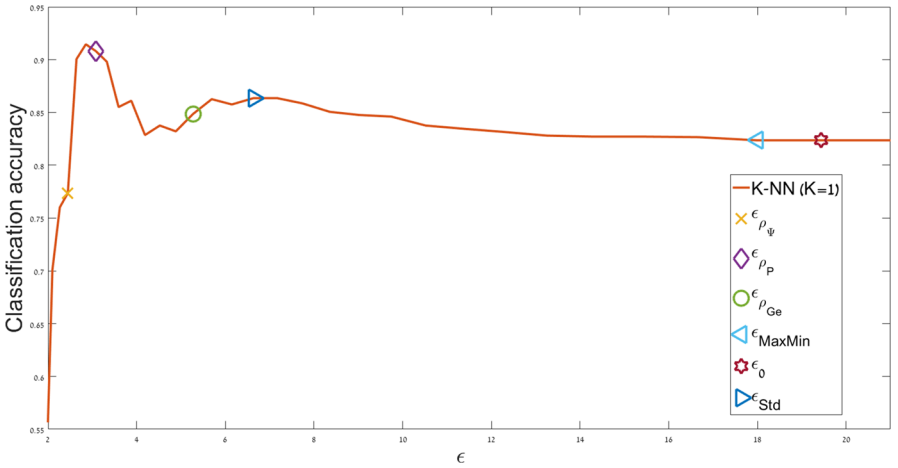


Fig. 13 Accuracy of classification in the multiple features dataset. KNN ($k = 1$) is applied in a $d = 4$ dimensional diffusion based representation. The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots

4.2.4 Classification of COVID-19 using chest X-ray images

In the next evaluation, we focus on classifying individuals that were infected by COVID-19. In certain individuals, the COVID-19 disease may cause symptoms of pneumonia; these individuals could be further diagnosed using chest X-ray images (Abbas et al. 2020). Convolutional neural networks (CNNs) have demonstrated promising results in classification of COVID-19 based on X-ray (Sethy and Behera 2020; Wang and Wong 2020) or CT (Shuai et al. 2020; Song et al. 2020) of chest images. Here, we focus on the task of classifying COVID-19 patients based on the

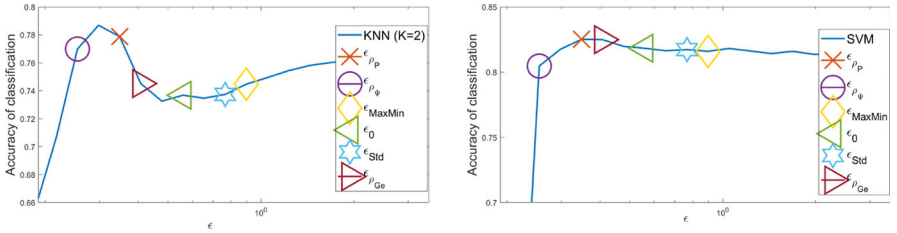


Fig. 14 Classification accuracy vs. value of ϵ on the COVID-19 chest X-ray dataset. The proposed scales ($\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P$) and existing methods ($\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std}$) are annotated on the plots. Left panel: Classification using KNN (K=2). Right panel: Classification using Support Vector Machine (SVM)

front view chest X-ray images. The data is collected from the Kaggle database,[1] from which we have used 112 images that are split equally between two classes: healthy and COVID-19 infected.

The feature space is defined by resizing all the chest X-rays to 200×200 pixel images. This feature space is still sensitive to translations and scales, which are inherent to the X-ray modality. This sensitivity could clearly be partly mitigated by defining a translation-invariant feature space, e.g. by using CNNs, however, this is beyond the scope of this study. To evaluate the proposed schemes, we compute the ratios ρ_P and ρ_Ψ for various values of ϵ, and estimate optimal scales based on Eqs. (3.18), (3.37). For each scale value, we extract a 2-dimensional embedding and perform classification using KNN (with $k = 2$) and SVM classifiers. Accuracy is averaged using a 10-folds cross-validation. Fig. 14 demonstrates that the proposed scales are good candidates for selecting a scale that yields high classification performance.

4.3 Practical guidelines

Our numerical simulations demonstrate that none of the proposed scaling schemes consistently outperforms the others. Nonetheless, across all of the evaluated datasets, the best performance was obtained within the range where the ϵ values were bounded by $\epsilon_\psi, \epsilon_{\rho_P}$ and ϵ_{Ge}. The probabilistic approach that estimates ϵ_{ρ_P} is based on the transition matrix $\boldsymbol{P}$ and requires $\mathcal{O}(N^2)$ computations. This complexity could be further reduced by computing a k-sparse approximation for the matrix $\boldsymbol{P}$. Specifically, methods such as k-sparse graph (Wang et al. 2013) can be computed with complexity $\mathcal{O}(NlogN + Nk)$. The probabilistic approach is based on a heuristic, and we consider it to be the least accurate of all of the proposed schemes.

Both the Spectral approach and the Geometric approach require a spectral decomposition. Assuming that both methods use the same number of coordinates (i.e., the embedding dimension is the same as the number of classes, $d = N_C$), they both require $\mathcal{O}(N^2 N_C)$ computations. The spectral approach is derived by analyzing a perturbed version of a kernel $\boldsymbol{K}$, constructed based on well-separated classes. Empirically, this analysis seems to hold for certain cases; however, if there is no spectral gap (i.e., $\lambda_{N_C} - \lambda_{N_C+1} \ll 1$) we do not recommend to use this method. Finally, the Geometric

[1] https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets.

approach is purely based on the extracted representation. Therefore, we consider it as the most reliable method for estimating a scale parameter ϵ that is most appropriate for classification.

5 Application: learning seismic parameters

In this section we demonstrate the capabilities of the proposed approach for extracting meaningful parameters from raw seismic recordings. Extracting reliable seismic parameters is a challenging task. Such parameters could help discriminate earthquakes from explosions, moreover, they can enable automatic monitoring of nuclear experiments. Traditional methods such as Rodgers et al. (1997); Blandford (1982) use signal processing to try to analyze seismic recordings. More recent methods, such as Kortström et al. (2016); Ruano et al. (2014); Ohrnberger (2001); Beyreuther et al. (2012); Hammer et al. (2013); Del Pezzo et al. (2003); Tiira (1996) use machine learning to construct a classifier for a variety of seismic events. Here, we extend our result from Rabin et al. (2016a); Lindenbaum et al. (2018), in which we have demonstrated the strength of DM for extracting seismic parameters. Our proposed method performs a vector scaling for manifold learning. Thus, essentially if the data lies on a manifold, our scaling combined with DM will extract the manifold from high-dimensional seismic recordings. Moreover, it will provide a natural feature selection procedure, thus if some features are corrupt, the proposed scaling may be able to reduce their influence.

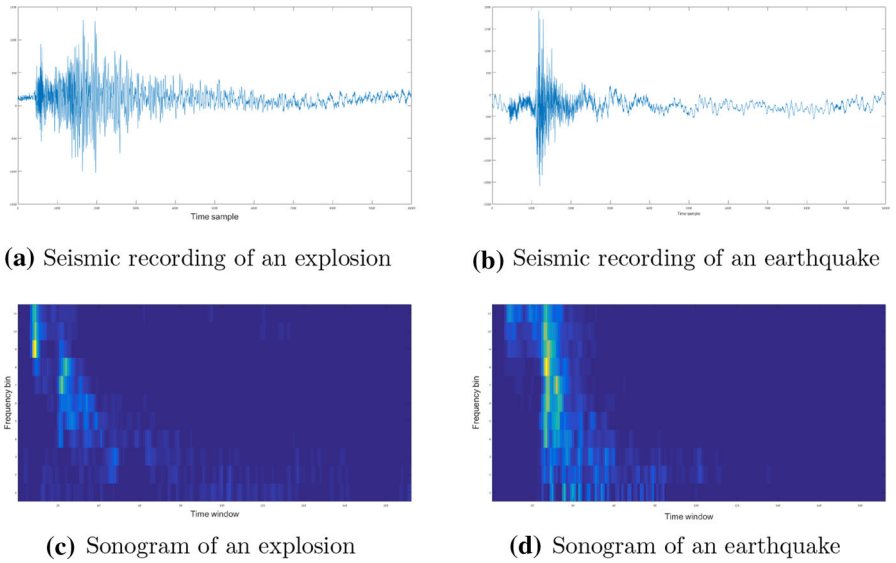
As a test case, we use a dataset from (Rabin et al. 2016a, b; Lindenbaum et al. 2018), which was recorded in Israel and Jordan between 2005–2015. All recordings were collected in HRFI (Harif) station located in the south of Israel. The station collects three signals from north (N), east (E) and vertical (Z). Each signal is sampled using a broadband seismometer at 40[Hz] and consists of 10,000 samples.

5.1 Feature extraction

Seismic events usually generate two waves, primary-waves (P) and secondary waves (S). The primary wave arrives directly from the source of the event to the recorder, while the secondary wave is a shear wave and thus arrives at some time delay. Both waves pass through different material thus have different spectral properties. This motivates the use of a time-frequency representation as the feature vector for each seismic event. The time-frequency representation used in this study is a Sonogram (Joswig 1990), which offers computation simplicity while retaining the sufficient spectral resolution for the task in hand. The Sonogram is basically a spectrogram, renormalized and rearranged in a logarithmic manner. Given a seismic signal $z(n)$, the Sonogram is extracted using the following steps:

1. Compute a discrete-time Short Term Fourier Transform:

$$\widehat{\boldsymbol{Z}}(f,t) = \sum_{n=1}^{N} w(n-\ell(t)) \cdot z(n) \cdot e^{-j2\pi fn}, \tag{5.1}$$



(a) Seismic recording of an explosion

(b) Seismic recording of an earthquake

(c) Sonogram of an explosion

(d) Sonogram of an earthquake

Fig. 15 Top: Example of a raw signal recorded from **a** an explosion and **b** earthquake. Bottom: The Sonogram matrix extracted from **c** an explosion and **d** earthquake

where $w(n - \ell(t))$ is a window function of length $N_0 = 512$, with a shift value of $\ell(t) = \lfloor (1 - s) \cdot N_0 \cdot t \rfloor$ time steps. We use overlap of $s = 0.85$ and compute for N_0 frequency bins, such as the values of f are spread uniformly on a logaritmic scale.

2. Normalize the energy by the number of frequency bins

$$\widetilde{\boldsymbol{Z}}(f, t) = \frac{|\widehat{\boldsymbol{Z}}(f, t)|^2}{N_0}. \tag{5.2}$$

3. Reshape the time frequency representation into a vector, by concatenating columns. The resulted vector representation for a signal $\boldsymbol{z}(n)$ is denoted by $\boldsymbol{x}$.

These steps are applied to each of the channels separately. This results in three sets $\boldsymbol{X}_E$ for the east channel, $\boldsymbol{X}_N$ for the north channel and $\boldsymbol{X}_Z$ for the horizontal. Examples for seismic recording of an explosion and of an earthquake are presented on Fig. 15a, b. Examples of corresponding Sonograms are presented in Fig. 15a, b .

5.2 Seismic manifold learning

To evaluate the proposed scaling for manifold learning, we use a subset of the seismic recording with 352 quarry blasts. The explosions have occurred at 4 known quarries surrounding the recording station HRFI. Our study in Rabin et al. (2016a), Lindenbaum et al. (2018), has demonstrated that most of the variability of quarry blasts stems from the source location of each quarry, therefor, we assume that the 352 blasts lie on some low-dimensional manifold. Where the parameters of the manifold should correlate

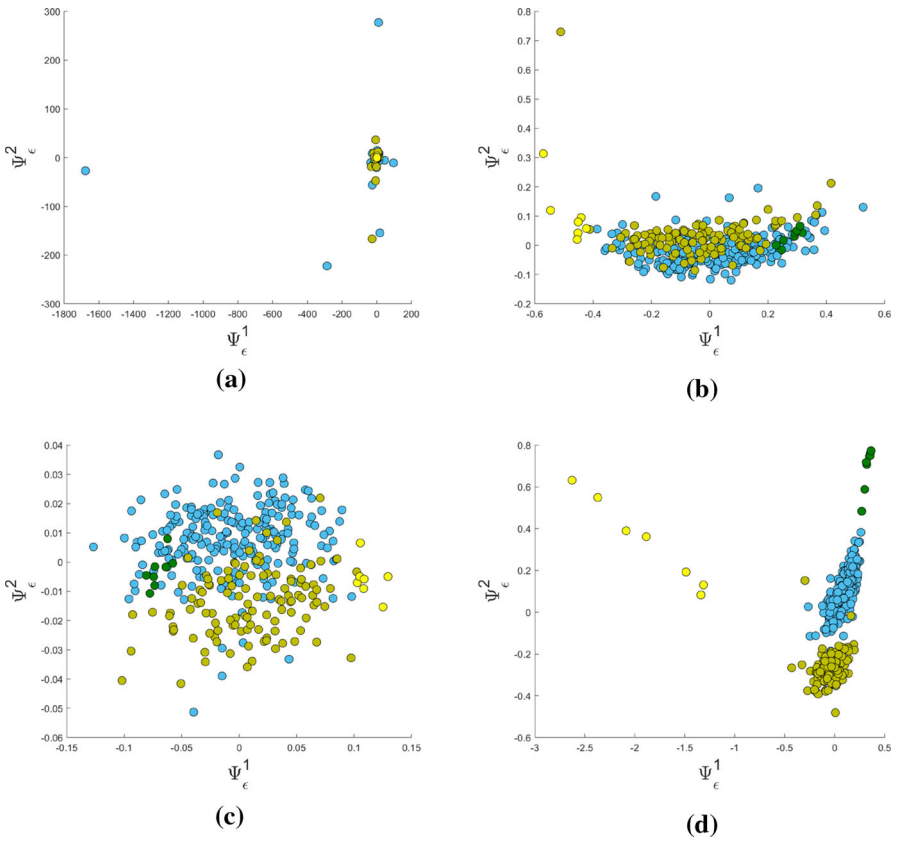


Fig. 16 The two leading DM coordinates of the 352 quarry blasts, colored by source quarry cluster. Scaling method based on: **a** The standard deviation of the data. **b** Singer's (2009) approach ϵ_0 (detailed in Algorithm 3.1. **c** The max-min methods ϵ_{MaxMin} (Eq. 3.1). **d** Proposed scaling for manifold learning (detailed in Algorithm 3.2)

with location parameters. Our approach for setting the scale parameter provides a natural feature selection procedure. To evaluate the capabilities of this procedure we "destroy" the information in some of the features. We do this by applying a deformation function to one channel out of the three seismometer recordings. We define the input for Algorithm 3.2 as

$$\boldsymbol{X} = [\boldsymbol{X}_N, \boldsymbol{X}_E, g(\boldsymbol{X}_Z)], \tag{5.3}$$

where $g(\cdot)$ is an element-wise deformation function. In the first test case the deformation function is defined by $g(y) = y^{0.1}$. Then, we apply DM with various scaling schemes and examine the extracted representation. In Fig.16 the two leading DM coordinates of different scaling methods are presented.

The quarry cluster separation is clearly evident in Fig. 16d. To further evaluate how well the low-dimensional representation correlates with the source location we use a list of source locations. A list with the explosions locations is provided to us based on manual calculations, performed by an analyst by considering the phase difference

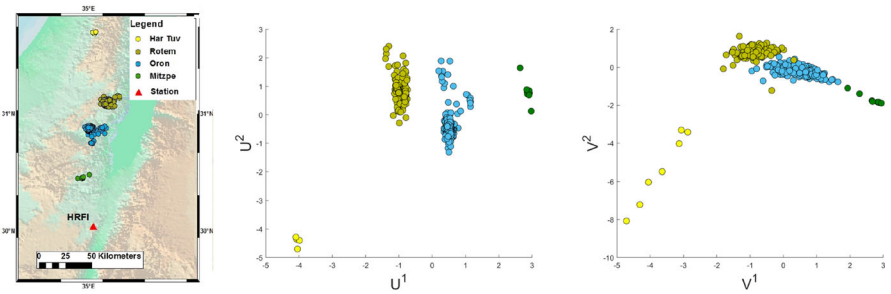


Fig. 17 **a** A map with source locations of 352 explosions. Points are colored by quarry cluster. **b** A CCA based representation of the latitude and longitude of the explosions. **c** A CCA based representation of the two leading DM coordinates extracted based on the proposed scaling (appear in Fig. 16d)

between the signals' arrival times to different stations. We note that this estimation is accurate up to a few kilometers. A map of the location estimates colored by source quarry is presented in Fig. 17a. Then, we apply Canonical Correlation Analysis (CCA) to find the most correlated representation. The transformed representations $\boldsymbol{U}$ and $\boldsymbol{V}$ are presented in Fig. 17b, c respectively. The two correlation coefficients between coordinates of $\boldsymbol{U}$ and $\boldsymbol{V}$ are 0.88 and 0.72.

In the second test case, we use additive Gaussian noise to "degrade" the signal. We use $g(y) = y + n_1$ as the defoemation function, where n_1 is drawn from a zero-mean Gaussian distribution with variance of σ_N^2. We estimate the scaling ϵ based on the proposed and alternative methods. Then, we apply CCA to the two leading DM coordinated and the estimated source locations. The top correlation coefficients for various values of σ_N^2 are presented in Fig. 18. Both the max-min method ϵ_{MaxMin} and Singer's ϵ_0 (Singer et al. 2009) scheme seem to break at the same noise level. The standard deviation approach is robust to the noise level, this is because it essentially performs whitening of the data. However, this also obscures some of the information content when the noise is of low power. The proposed approach seems to outperform all alternative schemes for this test case.

5.3 Classification of seismic events

Automatic classification of seismic events is useful as it may reduce false alarm warnings on one hand, and enable monitoring nuclear events on the other hand. To evaluate the proposed scaling for classification of seismic events, we use a set with 46 earthquakes and 62 explosions all of which were recorded in Israel. A low-dimensional mapping is extracted by using DM with various values of ϵ, and binary classification was applied using KNN ($k = 5$) and Support Vector Machine (SVM) in a leave-one-out fashion. The accuracy of the classification for each value of ϵ is presented in Fig. 19. The estimated values of ϵ_{Ge}, ϵ_{ρ_P} and ϵ_{ρ_Ψ} were annotated. It is evident that for classification the estimated values are indeed close to the optimal values, although they do not fully coincide. Nevertheless, they all achieve high classification accuracy.

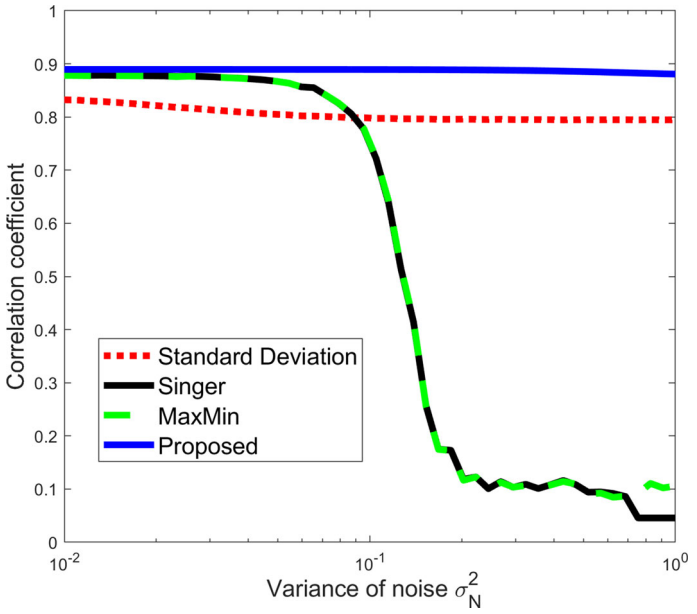


Fig. 18 Highest correlation coefficient between the DM representation extracted using various scaling schemes. The x-axis corresponds to the variance of the additive Gaussian noise

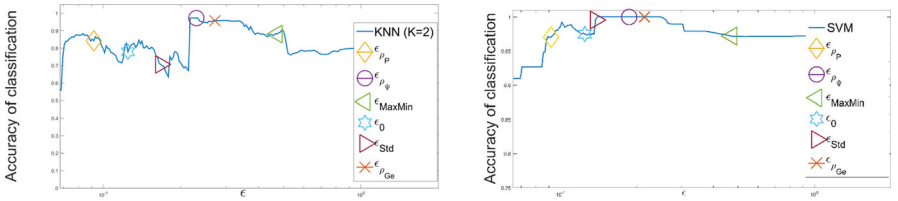


Fig. 19 Classification accuracy vs. value of ϵ. The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots. Left panel: Classification using KNN (K = 2). Right panel: Classification using Support Vector Machine (SVM)

6 Conclusions

The scaling parameter ϵ of the widely used Gaussian kernel is often crucial for machine learning algorithms. As happens in many tasks in the field, there does not seem to be one global scheme that is optimal for all applications. For this reason, we propose two new frameworks for setting a kernel's scale parameter tailored for two specific tasks. The first approach is useful when the high-dimensional data points lie on some lower dimensional manifold. By exploiting the properties of the Gaussian kernel, we extract a vectorized scaling factor that provides a natural feature selection procedure. Theoretical justification and simulations on artificial data demonstrate the strength of the scheme over alternatives. The second approach could improve the performance of a wide range of kernel based classifiers. The capabilities of the proposed methods are demonstrated using artificial and real datasets. Finally, we present an application for

the proposed approach that helps learn meaningful seismic parameters in an automated manner. In the future, we intend to generalize the approach for the multi-view setting recently studied in Lindenbaum et al. (2020), Salhov et al. (2019), Lederman and Talmon (2014).

Acknowledgements This work was supported by the Israel Science Foundation (ISF 1556/17). This research was partially supported by the US-Israel Binational Science Foundation (BSF 2012282), Blavatnik Computer Science Research Fund, Blavatink ICRC Funds and Pazy Foundation.

Appendix

Dimensionality from Angle and Norm Concentration (DANCo) Ceruti et al. (2014) DANCo is a recent method for estimating the intrinsic dimension based on high-dimensional measurements. The estimate is based on the following steps:

1. For each point $\boldsymbol{x}_i$, $i = 1, ..., N$, find the set of $\ell+1$ nearest neighbors $\mathcal{S}^{\ell+1}(\boldsymbol{x}_i) = \{\boldsymbol{x}_{s_j}\}_{j=1}^{\ell+1}$. Denote the farthest neighbor of $\boldsymbol{x}_i$ by $\widehat{S}(\boldsymbol{x}_i)$. The value of ℓ depends on the density of the dataset, and is usually not higher than 10.
2. Calculate the normalized closest distance for $\boldsymbol{x}_i$ as $\rho(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \mathcal{S}^{\ell+1}(\boldsymbol{x}_i)} \frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||}{||\boldsymbol{x}_i - \widehat{S}(\boldsymbol{x}_i)||}$.
3. Use Maximum Likelihood (ML) to estimate $\hat{d}_{ML} = \arg\max \mathcal{L}(d)$, where the log likelihood is

$$\mathcal{L}(d) = N \log \ell d + (d-1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \log \rho(\boldsymbol{x}_i) + (\ell - 1) \sum_{\boldsymbol{x}_i \in \boldsymbol{X}} \log(1 - \rho^d(\boldsymbol{x}_i)). \quad (6.1)$$

4. For each point $\boldsymbol{x}_i$, find the ℓ nearest neighbors and center them relative to $\boldsymbol{x}_i$. The translated points are denoted as $\tilde{\boldsymbol{x}}_{s_j} \triangleq \boldsymbol{x}_{s_j} - \boldsymbol{x}_i$. The set of ℓ nearest neighbors for point $\boldsymbol{x}_i$ is denoted by $\tilde{\mathcal{S}}^{\ell}(\boldsymbol{x}_i) = \{\tilde{\boldsymbol{x}}_{s_j}\}_{j=1}^{\ell}$. The distribution model is explained in Camastra (2003).
5. Calculate the $\binom{\ell}{2}$ angles for all pairs of vectors within $\tilde{\mathcal{S}}^{\ell}(\boldsymbol{x}_i)$. The angles are calculated using

$$\theta(\boldsymbol{x}_{s_j}, \boldsymbol{x}_{s_m}) = \arccos \frac{\tilde{\boldsymbol{x}}_{s_j} \cdot \tilde{\boldsymbol{x}}_{s_m}}{||\tilde{\boldsymbol{x}}_{s_j}|| ||\tilde{\boldsymbol{x}}_{s_m}||}. \quad (6.2)$$

   For each point $\boldsymbol{x}_i$ concatenate all angles from Eq. (6.2) into a vector $\bar{\boldsymbol{\theta}}_i$ and the set of vectors by $\widehat{\boldsymbol{\theta}} \triangleq \{\bar{\boldsymbol{\theta}}_i\}_{i=1}^{N}$.
6. Estimate the set of parameters $\hat{\boldsymbol{\nu}} = \{\hat{\nu}_i\}_{i=1}^{N}$ and $\hat{\boldsymbol{\tau}} = \{\hat{\tau}_i\}_{i=1}^{N}$ based on a ML estimation using the von Mises (VM) distribution with respect to $\boldsymbol{X}$. The VM pdf describes the probability for θ given the mean direction ν and the concentration parameter $\tau \geq 0$. The VM pdf, as well as the ML solution, are presented in Ceruti et al. (2014). The means of $\hat{\boldsymbol{\nu}}$ and $\hat{\boldsymbol{\tau}}$ are denoted as $\hat{\mu}_{\nu}$ and $\hat{\mu}_{\tau}$, respectively.
7. For each hypothesis of $d = 1, ..., D$, draw a set of N data points $\boldsymbol{Y}^d = \{\boldsymbol{y}_i^d\}_{i=1}^{N}$ from a d-dimensional unit hypersphere.
8. Repeat steps 1-6 for the artificial dataset $\boldsymbol{Y}^d$. Denote the maximum likelihood estimated set of parameters as $\tilde{d}_{ML}$, $\tilde{\boldsymbol{\nu}}$, $\tilde{\boldsymbol{\tau}}$, $\tilde{\mu}_{\nu}$, $\tilde{\mu}_{\tau}$.

9. Obtain $\hat{d}$ by minimizing the Kullback-Leibler (KL) divergence between the distribution based on $\boldsymbol{X}$ and $\boldsymbol{Y}^d$. The estimator takes the following form

$$\hat{d} = \underset{d=1,\ldots,D}{\arg\min}\, \boldsymbol{KL}(g(\cdot;\ell,\hat{d}_{ML}), g(\cdot;\ell,\tilde{d}_{ML})) + \boldsymbol{KL}(q(\cdot;\hat{\mu}_\nu,\hat{\mu}_\tau), q(\cdot;\tilde{\mu}_\nu,\tilde{\mu}_\tau)),$$

where g is the pdf of the normalized distances and q is the VM pdf. Both g and q are described in Ceruti et al. (2014).

References

Abbas A, Abdelsamea MM, Gaber MM (2020) Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network, arXiv preprint arXiv:2003.13815

Belkin M, Niyogi P (2001) Laplacian eigenmaps and spectral techniques for embedding and clustering. NIPS 14:585–591

Beyreuther M, Hammer C, Wassermann M, Ohrnberger M, Megies M (2012) Constructing a hidden markov model based earthquake detector: application to induced seismicity. Geophys J Int 189:602–610

Blandford R (1982) Seismic event discrimination. Bull Seismol Soc Am 72:569–587

Camastra F (2003) Data dimensionality estimation methods: a survey. Pattern Recogn 36(12):2945–2954

Campbell C, Cristianini N, Shawe-Taylor J (1999) Dynamically adapting kernels in support vector machines. Adv Neural Inf Process Syst 11:204–210

Ceruti C, Bassis S, Rozza A, Lombardi G, Casiraghi E, Campadelli P (2014) Danco: Dimensionality from angle and norm concentration. Pattern Recogn 47(8):2569–2581

Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1–3):131–159

Cohen I, Tian Q, Zhou XS, Huang TS (2002) Feature selection using principal feature analysis. Univ. of Illinois at Urbana-Champaign,

Coifman RR, Lafon S (2006) Diffusion maps. Appl Comput Harmonic Anal 21:5–30

Coifman RR, Shkolnisky Y, Sigworth FJ, Singer A (2008) Graph laplacian tomography from unknown random projections. IEEE Trans Image Process 17(10):1891–1899

Del Pezzo E, Esposito A, Giudicepietro F, Marinaro M, Martini M, Scarpetta S (2003) Discrimination of earthquakes and underwater explosions using neural networks. Bull Seismol Soc Am 93(1):215–223

Dhillon IS, Guan Y, Kulis B (2004) Kernel k-means: spectral clustering and normalized cuts. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 551–556

Ding CH, He X, Simon HD (2005) On the equivalence of nonnegative matrix factorization and spectral clustering. In: SDM, vol 5. SIAM, pp 606–610

Fukunaga K, Olsen DR (1971) An algorithm for finding intrinsic dimensionality of data. IEEE Trans Comput 100(2):176–183

Gaspar P, Carbonell J, Oliveira JL (2012) On the parameter optimization of support vector machines for binary classification. J Integr Bioinform 9(3):201

Hammer C, Ohrnberger M, Fäh D (2013) Classifying seismic waveforms from scratch: a case study in the alpine environment. Geophys J Int 192:425–439

Hein M, Audibert J-Y (2005) Intrinsic dimensionality estimation of submanifolds in R D. In: Proceedings of the 22nd international conference on machine learning. ACM, pp 289–296

Jolliffe I (2002) Principal component analysis. Wiley Online Library

Joswig M (1990) Pattern recognition for earthquake detection. Bull Seismol Soc Am 80(1):170–186

Kortström J, Uski M, Tiira T (2016) Automatic classification of seismic events within a regional seismograph network. Comput Geosci 87:22–30

Kruskal JB, Wish M (1977) Multidimensional scaling. Sage Publications, Beverly Hills

Lafon S, Keller Y, Coifman RR (2006) Data fusion and multicue data matching by diffusion maps. IEEE Trans Pattern Anal Mach Intell 28(11):1784–1797

Lafon S, Keller Y, Coifman R (2006) Data fusion and multicue data matching by diffusion maps. IEEE Trans Pattern Anal Mach Intell 28(11):1784–1797

Lederman RR, Talmon R (2014) Common manifold learning using alternating-diffusion, submitted, Tech. Report YALEU/DCS/TR1497, Tech. Rep
Lichman M (2013) UCI machine learning repository [Online]
Lindenbaum O, Yeredor A, Cohen I (2015) Musical key extraction using diffusion maps. Sig Process 117:198–207
Lindenbaum O, Bregman Y, Rabin N, Averbuch A (2018) Multi-view kernels for low-dimensional modeling of seismic events. IEEE Trans Geosci Remote Sens 56(6):3300–3310
Lindenbaum O, Yeredor A, Salhov M, Averbuch A (2020) Multiview diffusion maps. Inf Fusion 55:127–149
Lindenbaum O, Yeredor A, Averbuch A (2016) Bandwidth selection for kernel-based classification. In: IEEE international conference on the science of electrical engineering (ICSEE), pp 1–5. IEEE
Lin T, Zha H, Lee SU (2006) Riemannian manifold learning for nonlinear dimensionality reduction. In: European conference on computer vision. Springer, pp 44–55
Lu Y, Cohen I, Zhou XS, Tian Q (2007) Feature selection using principal feature analysis. In: Proceedings of the 15th ACM international conference on Multimedia. ACM, pp 301–304
Luo W (2011) Face recognition based on laplacian eigenmaps. In: International conference on computer science and service system (CSSS). IEEE, pp 416-419
Moser J (1965) On the volume elements on a manifold. Trans Am Math Soc 120(2):286–294
Ng AY, Jordan MI, Weiss Y et al (2002) On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 2:849–856
Ohrnberger M (2001) Continuous automatic classification of seismic signals of volcanic origin at Mt. Merapi, Java, Indonesia, PhD thesis, University of Potsdam
Pettis KW, Bailey TA, Jain AK, Dubes RC (1979) An intrinsic dimensionality estimator from near-neighbor information. IEEE Trans Pattern Anal Mach Intell 1:25–37
Rabin N, Bregman Y, Lindenbaum O, Ben-Horin Y, Averbuch A (2016) Earthquake-explosion discrimination using diffusion maps. Geophys J Int 207(3):1484–1492
Rabin N, Bregman Y, Lindenbaum O, Ben-Horin Y, Averbuch A (2016) Multi-channel fusion for seismic event detection and classification. In: IEEE international conference on the science of electrical engineering (ICSEE). IEEE, pp 1–5
Rodgers AJ, Lay T, Walter WR, Mayeda KM (1997) A comparison of regional-phase amplitude ratio measurement techniques. Bull Seismol Soc Am 87(6):1613–1621
Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by local linear embedding. Science 290(5500):2323–2326
Ruano AE, Madureira G, Barros O, Khosravani HR, Ruano MG, Ferreira PM (2014) Seismic detection using support vector machines. Neurocomputing 135:273–283
Salhov M, Lindenbaum O, Silberschatz A, Shkolnisky Y, Averbuch A (2019) Multi-view kernel consensus for data analysis and signal processing. Applied and Computational Harmonic Analysis
Scholkopf B, Smola AJ (2001) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press
Sethy PK, Behera SK (2020) Detection of Coronavirus Disease (COVID-19) Based on Deep Features. *Preprints 2020, 2020030300*
Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22(8):888–905
Shuai W, Bo Kang K, Jinlu M, Xianjun Z, Mingming X, Jia G, Mengjiao C, Jingyi Y, Yaodong L, Xiangfei M, Bo X (2020) A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19) medRxiv
Singer A, Erban R, Kevrekidis I, Coifman RR (2009) Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. PNAS 106(38):16090–16095
Song F, Guo Z, Mei D (2010) Feature selection using principal component analysis. In: International conference on system science, engineering design and manufacturing informatization (ICSEM), 2010, vol 1. IEEE, pp 27–30
Song Y, Zheng S, Li L, Zhang X, Zhang X, Huang Z, et al. (2020) Deep learning enables accurate diagnosis of novel coronavirus (COVID-19) with CT images. medRxiv
Staelin C (2003) Parameter selection for support vector machines, Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1,
Stewart GW (1990) Matrix perturbation theory. Citeseer,
Taseska M, Van Waterschoot T, Habets EA, Talmon R (2019) Nonlinear filtering with variable-bandwidth exponential kernels. IEEE Trans Signal Process

Tenenbaum J, de Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323

Tiira T (1996) Discrimination of nuclear explosions and earthquakes from teleseismic distances with a local network of short period seismic stations using artificial neural networks. Phys Earth Planet Inter 97(1–4):247–268

Trunk GV (1976) Stastical estimation of the intrinsic dimensionality of a noisy signal collection. IEEE Trans Comput 100(2):165–171

Vasiloglou N, Gray AG, Anderson DV (2006) Parameter estimation for manifold learning, through density estimation. In: 2006 16th IEEE signal processing society workshop on machine learning for signal processing, pp 211–216

Verveer PJ, Duin RPW (1995) An evaluation of intrinsic dimensionality estimators. IEEE Trans Pattern Anal Mach Intell 17(1):81–86

Wang D, Shi L, Cao J (2013) Fast algorithm for approximate k-nearest neighbor graph construction. In: 2013 IEEE 13th international conference on data mining workshops, pp 349–356

Wang L, Wong A (2020) COVID-net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images, arXiv preprint arXiv:2003.09871

Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. In: Advances in neural information processing systems, pp 1601–1608

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.