Kernel Scaling for Manifold Learning and Classification

Ofir Lindenbaum[a], Moshe Salhov[b], Arie Yeredor[a], Amir Averbuch[b]

[a]*School of Electrical Engineering, Tel Aviv University, Israel*
[b]*School of Computer Science, Tel Aviv University, Israel*

Abstract

Kernel methods play a critical role in many dimensionality reduction algorithms. They are useful in manifold learning, classification, clustering and other machine learning tasks. Setting the kernel's scale parameter, also referred as the kernel's bandwidth, highly affects the extracted low-dimensional representation. We propose to set a scale parameter that is tailored to the desired application such as classification and manifold learning. The scale computation for the manifold learning task enables that the dimension of the extracted embedding equals the intrinsic dimension estimation. Three methods are proposed for scale computation in a classification task. The proposed frameworks are simulated on artificial and real datasets. The results show a high correlation between optimal classification rates and the computed scaling.

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) [1], Multidimensional Scaling (MDS) [2], Isomap [3] and Local Linear Embedding [4] aim to extract essential information from high-dimensional data points based on their pairwise connectivities. Graph-based kernel methods such as Laplacian Eigenmaps [5] and Diffusion Maps (DM) [6], construct a positive semi-definite kernel based on the multidimensional data points to recover the underlying structure. Such methods have been proven effective for tasks such as clustering [7], classification [8], manifold learning [9] and many more.

Kernel methods rely on computing a distance function (usually Euclidean) between all pairs of data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X} \subseteq \mathbb{R}^{D\times N}$ 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

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

Email address: ofirlin@gmail.com (Ofir Lindenbaum)

As shown, for example, in [4, 6], spectral analysis of such a kernel provides an efficient representation of the geometry of the lower dimensional data in the ambient space. The construction of such a kernel requires expert knowledge for setting two parameters, namely the scaling ϵ (Eq. (1.1)) and the target dimension d of the low dimensional space. We focus in this paper on setting the scale parameter ϵ also called kernel bandwidth.

Construction of the kernel matrix (Eq. 1.1) requires to set the value of the scale (width) parameter ϵ. This parameter is related to the statistics and to the geometry of the data points. The Euclidean distance is used for learning the geometry of the data. However, it is meaningful only locally when this distance is 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 [8]. 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 ways to set ϵ. A study by [10] suggests a method which enforces that most of the data is connected. The sum of the kernel is used in [11] to find a range of valid scales. The approach in [12] sets a dynamic scale and is applicable for spectral clustering. Others simply use the standard deviation of the data as ϵ.

Kernel methods are also used for Support Vector Machines [13], where the goal is to find a feature space that separates between the given classes. Methods such as [14, 15] use cross-validation to find the scale parameter which achieves peak classification results on a given training set. The study in [16] suggests an iterative approach that updates the scale until reaching a maximal separation between classes. A study in [17] relates 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.

In this paper, we propose new methodologies to set the scale parameter ϵ. The proposed frameworks address two types of problems: manifold learning and spectral based classification. For the manifold learning task, we estimate the manifold's intrinsic dimension and choose a scale so that this estimation corresponds to the estimated dimension that is based on the kernel $\boldsymbol{K}$ (Eq. 1.1). We provide an analysis of this approach as well as simulations that demonstrate its performance. For the classification task, we propose three methods. The first seeks 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 final method sets the scale which maximizes the within class transition probability. This approach does not require to compute an eigendecomposition. We 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 section 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.

2. Preliminaries

This section provides a brief description of two methods used in this study: kernel-based method for dimensionality reduction titled Diffusion Maps [6] and Dimensionality from Angle and Norm Concentration (DANCo). DANCo is an algorithm which estimates the intrinsic dimension of a manifold based on the ambient high dimensional data.

2.1. Diffusion Maps (DM)

DM [6] is a 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 $\boldsymbol{X} \subseteq \mathbb{R}^{D\times N}$, the DM framework is constructed based on the following steps:

1. A kernel function $\mathcal{K} : \boldsymbol{X} \times \boldsymbol{X} \longrightarrow \mathbb{R}$ is chosen. It is represented by a matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ which satisfies for all $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$ the following properties:
   Symmetry: $K_{i,j} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_i)$ and positive semi-definiteness: $\boldsymbol{v}_i^T \boldsymbol{K} \boldsymbol{v}_i \geq 0$ for all $\boldsymbol{v}_i \in \mathbb{R}^N$ and $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$. *These properties guarantee that the matrix* $\boldsymbol{K}$ *has real eigenvectors and non-negative real eigenvalues. In this study, we focus on the common choice of a Gaussian kernel (see Eq. 1.1)*

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

   as the affinity measure between two multidimensional data vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$;
2. Selection of the appropriate scale ϵ. It determines the connectivity of the kernel;
3. Computation of the diagonal sum of rows matrix $\boldsymbol{D}$ where $D_{i,i} = \sum_j K_{i,j}$. Re-normalization of the kernel by using $\boldsymbol{D}$ is

$$P_{i,j} = \mathcal{P}(\boldsymbol{x}_i, \boldsymbol{x}_j) = [\boldsymbol{D}^{-1}\boldsymbol{K}]_{i,j}. \tag{2.2}$$

   The resulting matrix $\boldsymbol{P} \in \mathbb{R}^{N\times N}$ is a row stochastic such that the expression $P_{i,j} = p(\boldsymbol{x}_i,\boldsymbol{x}_j)$ describes the transition probability from the point $\boldsymbol{x}_i$ to point $\boldsymbol{x}_j$ in one time step;
4. Spectral decomposition is applied to the matrix $\boldsymbol{P}$ to obtain a sequence of eigenvalues $\{\lambda_n\}$ and normalized eigenvectors $\{\boldsymbol{\psi}_n\}$ that satisfy $\boldsymbol{P}\boldsymbol{\psi}_n = \lambda_n\boldsymbol{\psi}_n, n = 0, ..., N-1$;
5. 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 i^{th} element of $\boldsymbol{\psi}_m$.
   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 $p(\boldsymbol{x}_j, :)$, $i, j = 1, ..., M$, 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 the entries $W_{i,i} = \frac{D_{i,i}}{\sum_{i=1}^{M} D_{i,i}}$. This equality is proved in [6].

6. A low dimension mapping $\boldsymbol{\Psi}_\epsilon^d(\boldsymbol{x}_i), i = 1, ..., N$ is set 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 $\lambda_{d+1}, ..., \lambda_N \longrightarrow 0$.

2.2. Dimensionality from Angle and Norm Concentration (DANCo) [18]

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 $\mathcal{M}$, the intrinsic dimension $\bar{d}$ is the minimum number of parameters needed to represent the manifold.

Defenition 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 a manifold, the intrinsic dimension is the smallest integer $\bar{d}$ such that a smooth map $f(\xi)$ describes the manifold $\mathcal{M} = f(\xi)$, $\xi \in \mathcal{R}^{\bar{d}}$.*

Methods such as [19, 20] 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 [21, 22], use K-NN distances to find a subspace around each point and based on some statistical assumption estimate $\bar{d}$. A survey of different approaches is presented in [23]. In this study, we use Dimensionality from Angle and Norm Concentration (DANCo) [18], which proved to be the most robust approach in our experiments. The estimation, which is denoted as $\hat{d}$, 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 $\widehat{S}(\boldsymbol{x}_i)$.
2. Calculate the normalized closest distance for $\boldsymbol{x}_i$ as $\rho(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_j \in \mathcal{S}(\boldsymbol{x}_i)} \frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||}{||\boldsymbol{x}_i - \widehat{S}(\boldsymbol{x}_i)||}$.
3. Use maximum likelihood to estimate $\hat{d}_{ML} = \arg\max \mathcal{L}(d)$, where the log likelihood is

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

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 $\tilde{\mathcal{S}}^\ell(\boldsymbol{x}_i) = \{\tilde{\boldsymbol{x}}_{s_j}\}_{j=1}^{\ell}$.

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

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

   Define the vector from the angles $\bar{\boldsymbol{\theta}}_i$ 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. 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 [18]. 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 Kullbacl-Leibler (KL) divergence between the distribution based on $\boldsymbol{X}$ and $\boldsymbol{Y}^d$. The estimator takes the following form

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

   where g is the pdf of the normalized distances and q is the VM pdf. Both g and q are described in [18].

The algorithm jointly uses the normalized distances and mutual angles to extract a robust estimation of the intrinsic dimension $\bar{d}$. This is done by finding the dimension that minimizes the KL divergence between an artificially generated data and the observed data. In section (3.2), we propose a framework which exploits the intrinsic dimension estimation $\hat{d}$ for choosing the scale parameter ϵ defined in Eq. (3.4).

3. Setting the Scale Parameter ϵ

3.1. Existing Methods

Several studies have proposed methods for setting the scale parameter ϵ. Some choose ϵ as the standard deviation of the data. This approach is good when the data is sampled from a uniform distribution. A max-min measure is suggested in [24] where the scale is set to

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

where $\mathcal{C} \in [2, 3]$. This approach attempts to set a small scale to maintain local connectivities. Another scheme in [11] 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 where

the Gaussian bell shape exists. The scheme in [11] 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 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. When $\epsilon \to \infty$, $\boldsymbol{K}$ approaches an all-ones matrix. We denote by ϵ_0 the minimal value within the range $\bar{\epsilon}$ (defined in Alg. 3.1). This value is used in the simulations presented in Section 4. A dynamic scale is proposed in [12]. It suggests to calculate a local-scale σ_i for each data point $\boldsymbol{x}_i, i = 1, ..., N$. The scale is chosen as the L_1 distance from the rth nearest neighbor of the point $\boldsymbol{x}_i$. Explicitly, the calculation for each point is

$$\sigma_i = ||\boldsymbol{x}_i - \boldsymbol{x}_r||, \tag{3.2}$$

where $\boldsymbol{x}_r$ is the rth nearest neighbor of the point $\boldsymbol{x}_i$. The value of the kernel for points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is

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

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

All the mentioned methods treat ϵ as a scalar. Thus, when data is sampled from various types of sensors these methods are not effective. In these cases, each feature $l = 1, .., D$, in a data vector $x_i[l]$ requires a different scale. In order to re-scale the vector, a diagonal positive scaling matrix $\boldsymbol{A} \succ 0$ is introduced. The rescaling of the feature vector $\boldsymbol{x}_i$ is set as $\widehat{\boldsymbol{x}}_i = \boldsymbol{A}\boldsymbol{x}_i, 1 \leq i \leq N$. The kernel matrix is rewritten as

$$K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) = exp\{-\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{A}\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)/2\varepsilon\}. \tag{3.4}$$

A standard way to choose the values for such a scaling matrix $A_{l,l}, l = 1, ..., D$, is achieved by using the standard deviation of the data. Explicitly, the values of $\boldsymbol{A}$ are set such that

$$A_{l,l} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[x_i(l) - \mu_l]^2}, l = 1, ..., D, \ \epsilon_{std} = 1, \tag{3.5}$$

where $x_i(l)$ is the l-th coordinate of the vector $\boldsymbol{x}_i$ and μ_l is the mean value of this coordinate.

3.2. Optimal Scale ϵ for Manifold Learning

In this section, we propose a framework for setting the scale parameter ϵ when some low dimensional manifold $\mathcal{M}$ is submerged within the dataset $\boldsymbol{X}$. We start by utilizing the results of [25, 26], which relate the scale parameter ϵ (Eq. 1.1) and the intrinsic dimension $\bar{d}$ (Definition 2.2.1). In [25] a range of valid values is suggested for ϵ, here we expand the results from [25, 26] by introducing a diagonal positive scaling matrix $\boldsymbol{A}$ (Eq. 3.4). This diagonal matrix enables a feature selection procedure which emphasizes the latent structure of the manifold. This procedure is describe in Algorithm 3.2.

Let $\boldsymbol{K}$ be the kernel function from Eq. 3.4 and a diagonal matrix s.t. $\boldsymbol{A} \succ 0, \hat{\boldsymbol{x}}_i = \boldsymbol{A} \cdot \boldsymbol{x}_i, i = 1, ..., N$. Following the analysis in [25], we have

$$\sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) = \sum_{i,j} exp\{-\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \boldsymbol{A}^T \boldsymbol{A} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)/2\varepsilon\}. \tag{3.6}$$

If the data points in $\boldsymbol{X}$ are independently uniformly distributed over a manifold $\mathcal{M}$ then the approximation in Eq. 3.7 holds with small errors. The sum in Eq. 3.6 is approximated by the mean value theorem as

$$\sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) \approx \frac{N^2}{Vol\left(\mathcal{M}\right)^2} \int_{\mathcal{M}} \int_{\mathcal{M}} exp\{-\left(\boldsymbol{x} - \boldsymbol{y}\right)^T \boldsymbol{A}^T \boldsymbol{A} \left(\boldsymbol{x} - \boldsymbol{y}\right)/2\varepsilon\} d\boldsymbol{x} d\boldsymbol{y}, \tag{3.7}$$

where $Vol\left(\mathcal{M}\right)$ is the volume of the manifold $\mathcal{M}$. For sufficiently small values of ε, the integral over the manifold $\mathcal{M}$ is approximated by

$$\sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) \approx \frac{N^2}{Vol\left(\mathcal{M}\right)^2} \int_{\mathbb{R}^{\bar{d}}} \int_{\mathbb{R}^{\bar{d}}} exp\{-\left(\boldsymbol{x} - \boldsymbol{y}\right)^T \boldsymbol{A}^T \boldsymbol{A} \left(\boldsymbol{x} - \boldsymbol{y}\right)/2\varepsilon\} d\boldsymbol{x} d\boldsymbol{y}, \tag{3.8}$$

where $\bar{d}$ is its intrinsic dimension (Definition 2.2.1). For a sufficiently small ε, the integration over a small patch around $\boldsymbol{x}$ and $\boldsymbol{y}$ is similar to the integration over the corresponding tangent plane. In the limit $\varepsilon \to 0$, $\sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) \to N$ as the kernel approaches the identity matrix, whereas, in the limit $\varepsilon \to \infty$, we have $\sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) \to N^2$ as the kernel approaches the all ones matrix. In the linear region between $[0, \infty)$ and under the assumption that the sampling is sufficiently dense, the integral in Eq 3.7 has a close form solution

$$S\left(\varepsilon\right) \triangleq \sum_{i,j} K\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right) \approx \frac{N^2}{Vol\left(\mathcal{M}\right)} \left(2\pi\varepsilon\right)^{\frac{\bar{d}}{2}} ||\boldsymbol{A}^T \boldsymbol{A}||_F. \tag{3.9}$$

We note that in many practical datasets, the sampling is not sufficiently dense. In these cases, the scaling can distort the similarity imposed by the kernel.

The purpose of the matrix $\boldsymbol{A}$ is to scale all the D features prior to the computation of the kernel $\boldsymbol{K}$ (Eq. 2.1) or its parametrization $\boldsymbol{\Psi(X)}$ (Eq. 2.3). The scaling is designed to minimize the absolute distance between the estimated dimension $\hat{d}$ (section 2.2) and the intrinsic dimension d_ϵ in the embedding space (defined in Eq. 3.11). The expression

presented in Eq. 3.11 describes the influence of ϵ on the extracted dimension d_ϵ. We start by taking the *log* of both sides of Eq. 3.9

$$\log S(\varepsilon) = \log\left(\sum_{i,j} K(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j)\right) \approx \frac{\bar{d}}{2}\log(\varepsilon) + \log\left(\frac{N^2}{Vol(\mathcal{M})}(2\pi)^{\frac{\bar{d}}{2}}|\boldsymbol{A}^T\boldsymbol{A}|\right). \tag{3.10}$$

By taking the derivative with respect to ε in Eq. 3.10 and by reformulating we get

$$d_\epsilon \approx 2\varepsilon\frac{\partial \log(S(\varepsilon))}{\partial\varepsilon} = \frac{\sum_{i,j} r_{i,j} e^{-r_{i,j}(A)/2\varepsilon}}{\varepsilon\sum_{i,j} e^{-r_{i,j}(A)/2\varepsilon}}, \tag{3.11}$$

where $r_{i,j}(\boldsymbol{A}) \triangleq (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{x}_j)$. The diagonal scaling matrix $\boldsymbol{A}$ is the result from the optimization problem of the form

$$\boldsymbol{A} = arg\min_{A\succ 0, \forall l\neq m, [A]_{l,m}=0, \varepsilon} |d_\epsilon - \hat{d}|. \tag{3.12}$$

The solution to the optimization problem in Eq.(3.12) can be found by an exhaustive search for a sufficiently small D. For a large D, we propose to use the greedy Algorithm 3.2 that computes both the scaling matrix $\boldsymbol{A}$ and the scale parameter ε.

Algorithm 3.2: Manifold Based Vector Scaling

Input: Dataset: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \subseteq \mathbb{R}^{D\times N}$.
Output: Normalized dataset $\hat{\boldsymbol{X}}$ and the diagonal scaling matrix $\boldsymbol{A}$ (Eq. (3.6))

1: Apply DANCo [18] to $\boldsymbol{X}$ to estimate $\hat{d}$.
2: Set $\boldsymbol{A} = \boldsymbol{I}_m$.
3: Set $\hat{\boldsymbol{X}}^{(\hat{d})} = \left(\boldsymbol{X}_{1:\hat{d},:} - mean(\boldsymbol{X}_{1:\hat{d},:})\right)./std(\boldsymbol{X}_{1:\hat{d},:})$.
4: **for** $l = \hat{d}+1$ *to* D **do**
    Solve Eq. (3.12) using $\hat{\boldsymbol{X}}^{(l)} = [\hat{\boldsymbol{X}}^{(l-1)}, \boldsymbol{X}_{l,:}]$ to find $A_{l,l}$ and ε
    Update $\hat{\boldsymbol{X}}^{(l)} = [\hat{\boldsymbol{X}}^{(l-1)}, \boldsymbol{X}_{l,:}\cdot A_{l,l}]/\sqrt{\varepsilon}$

Algorithm 3.2 is initialized by normalizing the first $\hat{d}$ coordinate using standard deviation. In each iteration, a coordinate is added, and an exhaustive search is performed to find the optimal normalization for the inspected coordinate. The exhaustive search results in two normalization factors $A_{l,l}$ and ε that are applied before the next iteration. In the lth iteration, this process is repeated by solving a two-dimensional exhaustive search on two views. The first view is the normalized result of the $l-1$ iteration $\hat{\boldsymbol{X}}^{(l-1)}$ and the second view is the lth feature. The computational complexity of Algorithm 3.2 with k hypotheses of ε and $A_{l,l}$ is $O(N^2k^2)$, since in the computation of a single scaling hypothesis N^2 operations are needed.

The performance of Algorithm 3.2 depends on the order of the D features. We further propose to reorder the features using a soft feature selection procedure. The studies in

[27, 28, 29] 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 is termed Correlation Based Feature Permutation (CBFP) and is describe in Algorithm 3.3. The algorithm uses the correlation between the D features and $\hat{d}$ embedding coordinates to evaluate the impotence of each feature.

Algorithm 3.3: Correlation Based Feature Permutation (CBFP)

Input: Dataset: $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N\} \subseteq \mathbb{R}^{D\times N}$.

Output: Feature permutation vector $\boldsymbol{J}$, such that $\boldsymbol{J} = \sigma([1, ..., D])$ and $\sigma()$ is a permutation operation.

1: Apply DANCo [18] to $\boldsymbol{X}$ to estimate $\hat{d}$.

2: Compute ϵ_{MaxMin} based on Eq. 3.1.

3: Compute DM representation $\boldsymbol{\Psi}^{\hat{d}}$ using Eq. 2.5.

4: Compute feature-embedding correlation score defined as

$$C_i \triangleq \sum_{\ell=1}^{\hat{d}} |corr(\boldsymbol{X}_{i,:}, \boldsymbol{\Psi}^{\hat{d}}_{\ell,:})|, i = 1, ..., D. \tag{3.13}$$

5: Set $[\boldsymbol{V}, \boldsymbol{J}] = \text{sort}(\boldsymbol{C})$, where $\boldsymbol{V}, \boldsymbol{J}$ are the sorted values and corresponding indexes of $\boldsymbol{C}$.

6: Reorder the D features by $\tilde{\boldsymbol{X}} = \boldsymbol{X}(\boldsymbol{J}, :)$.

In section 4.1, we evaluate the performance of the proposed approach on artificial manifolds.

3.3. Optimal Scale ϵ for Classification

Classification algorithms use a metric space and an induced distance to compute the category of the unlabeled data points. As such, the specified space in which the classifier is applied affects the results. Dimensionality reduction is effective for capturing the essential intrinsic geometry of the data and neglecting the undesired information (such as noise). Dimensionality reduction could markedly improve classification results [8]. In this section, we demonstrate the influence of the scale parameter ϵ on the output of classification algorithms and propose three methods for choosing the scale parameter ϵ for a DM-based classification algorithm.

Given a training set $\boldsymbol{T} \subset \mathbb{R}^{D\times N}$ with N_C classes denoted by $\boldsymbol{C}_1, \boldsymbol{C}_2, ..., \boldsymbol{C}_{N_C}$. Each class contains N_P data points and the total data points is $N = N_P \cdot N_C$. For simplicity, we use the same number of data points in each class. Relaxing this restriction does not change the results presented in this section. We use a scalar scaling factor ϵ. The analysis provided in this section could be expanded to a vector scaling in a straightforward way.

3.3.1. The Geometric Approach

We start by describing an approach based on [8]. The basic idea is to find the scale parameter ϵ such that the classes are dense and far apart from each other in the embedding space. This 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 values of ϵ.
2. Denote by $\boldsymbol{\mu}_i$ the center of mass for class $\boldsymbol{C}_i$ and $\boldsymbol{\mu}_a$ is the center of mass for all the data points. Both $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_a$ are computed in the low dimension DM based embedding $\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n)$- see step 1.
3. For each class $\boldsymbol{C}_i$, compute the average square distance (in the embedding space) for the N_P data points from the center of mass $\boldsymbol{\mu}_i$ such that

$$D_{c_i} = \frac{1}{N_P} \sum_{\boldsymbol{x}_n \in \boldsymbol{C}_i} ||\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n) - \boldsymbol{\mu}_i||^2. \tag{3.14}$$

4. Compute the same measure for all data points such that

$$D_a = \frac{1}{N} \sum_{\boldsymbol{x}_n \in X} ||\boldsymbol{\Psi}_{\epsilon}^{d}(\boldsymbol{x}_n) - \boldsymbol{\mu}_a||^2. \tag{3.15}$$

5. Define

$$\rho_{\Psi} \triangleq \frac{D_a}{\sum_{i=1}^{N_C} D_{c_i}}. \tag{3.16}$$

6. Find ϵ which maximizes ρ_{Ψ}

$$\epsilon_{\rho_{\Psi}} = \underset{\epsilon}{arg\max}(\rho_{\Psi}). \tag{3.17}$$

The idea is that $\epsilon_{\rho_{\Psi}}$ (Eq. 3.17) inherits the inner structure of the classes and neglects the mutual structure. However, this approach requires to compute an eigendecomposition for each ϵ value. In section 4.2, we describe experiments that empirically evaluate the influence of ϵ on the performance of classification algorithms.

3.3.2. The Spectral Approach

In this section, we analyze the relation between the spectral properties of the kernel and the extracted DM-based representation.

The Ideal Case: We start with following a simple case presented in [30] for spectral clustering. The training set $\boldsymbol{T}$ consists of N_C classes $\boldsymbol{C}_1 \cup \boldsymbol{C}_2 \cup ... \cup \boldsymbol{C}_{N_C} = \boldsymbol{T}$ with N_P data points within each class. These N_C classes are assumed to be well separated in the ambient space. This separation is formulated using the following definitions:

1. The Euclidean gap is defined by

$$D_{Gap}(\boldsymbol{X}) \triangleq \min_{\boldsymbol{x}_i \in \boldsymbol{C}_l, \boldsymbol{x}_j \in \boldsymbol{C}_m, l \neq m} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2. \tag{3.18}$$

   This is the Euclidean distance between the two closest data points among all classes.

2. The maximal class width is defined by

$$D_{Class}(\boldsymbol{X}) \triangleq \max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{C}_l} ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2. \tag{3.19}$$

   This is the maximal Euclidean distance between data points from the same class.

We assume that $D_{Class} \ll D_{Gap}$ such that the classes are well separated. Therefore, the matrix $\boldsymbol{K}$ (Eq. 2.1) converges to the following block form

$$\bar{\boldsymbol{K}} = \begin{bmatrix} \boldsymbol{K}^{(1)} & 0 & ... & 0 \\ 0 & \boldsymbol{K}^{(2)} & ... & 0 \\ : & : & : & : \\ 0 & 0 & ... & \boldsymbol{K}^{(N_C)} \end{bmatrix}, \ \bar{\boldsymbol{P}} = \bar{\boldsymbol{D}}^{-1}\bar{\boldsymbol{K}}, \ \bar{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}^{(1)} & 0 & ... & 0 \\ 0 & \boldsymbol{P}^{(2)} & ... & 0 \\ : & : & : & : \\ 0 & 0 & ... & \boldsymbol{P}^{(N_C)} \end{bmatrix}, \tag{3.20}$$

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 $\epsilon \sim D_{Class}$ and the classes are connected.

Proposition 3.3.1. *Assume that $D_{Class} \ll D_{Gap}$ and the matrix $\bar{\boldsymbol{P}}$ (Eq. 3.20) has an eigenvalue $\lambda = 1$ with multiplicity N_P. Then, the first N_C coordinates of the DM mapping (Eq. 2.5) are piecewise constant. The explicit form of the first nontrivial eigenvector $\boldsymbol{\psi}_1$ is given by*

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

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. (3.20)), each block $\boldsymbol{P}^{(i)}, i = 1, ..., N_P$, is row stochastic. Thus,

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

$\boldsymbol{\psi}_i$ is the all one's vector at the rows that correspond to $\boldsymbol{P}^{(i)}$ (Eq. (3.20)) 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 $\boldsymbol{x}_i \in \boldsymbol{C}_l, l = 1, ..., N_C$, corresponds to a row within the sub-matrix $\boldsymbol{P}^{(l)}, l = 1, ..., N_C$ (Eq. (3.20)). 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 $\boldsymbol{T}$ then all the data points from within a class are mapped to a point in the embedded space.

Corollary 3.3.1. *Using the first N_C eigenvectors of $\bar{\boldsymbol{P}}$ (Eq. (3.20)) 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 $D_{Gap}(\boldsymbol{\Psi}_\epsilon^{N_C}) = 2$ and $D_{Class}(\boldsymbol{\Psi}_\epsilon^{N_C}) = 0$ by Eqs. (3.18) and (3.19), respectively.*

Proof. Based on the representation in Proposition 3.3.1 along with Eq. (3.18), we get

$$D_{Gap}(\boldsymbol{\Psi}_\epsilon^{N_C}) = \min_{\boldsymbol{x}_i \in \boldsymbol{C}_l, \boldsymbol{x}_j \in \boldsymbol{C}_m, l \neq m} ||\boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_i) - \boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_j)||^2 = \sum_{r=1}^{N_C} \lambda_r \cdot (\psi_r(i) - \psi_r(j))^2 =$$

$$\sum_{r=1}^{N_C-2} 1 \cdot (0-0)^2 + 1 \cdot (1-0)^2 + 1 \cdot (0-1)^2 = 2.$$

In a similar way, we get by Eq. (3.19)

$$D_{Class}(\boldsymbol{\Psi}_\epsilon^{N_C}) = \max_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{C}_l} ||\boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_i) - \boldsymbol{\Psi}_\epsilon^{N_C}(\boldsymbol{x}_j)||^2 = \sum_{r=1}^{N_C} \lambda_r \cdot (\psi_r(i) - \psi_r(j))^2 =$$

$$\sum_{r=1}^{N_C-2} 1 \cdot (0-0)^2 + 1 \cdot (0-0)^2 + 1 \cdot (1-1)^2 = 0.$$

□

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 $D_{Class} \ll D_{Gap}$. The data points in real datasets are not completely disconnected, and we can assume they have a low connectivity. This connectivity implies that off-diagonal values of the matrix $\boldsymbol{K}$ (Eq. 2.2) are non-zeros. We analyze this more realistic scenario by assuming that the kernel matrix 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 diagonal terms are non-zero and the matrix $\boldsymbol{K}$ (Eq. 2.2) has the following form

$$\tilde{\boldsymbol{K}} = \bar{\boldsymbol{K}} + \widehat{\boldsymbol{W}}, \tag{3.21}$$

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

$$\widehat{\boldsymbol{W}} = \begin{bmatrix} -\boldsymbol{W}^{(1,1)} & \boldsymbol{W}^{(1,2)} & ... & \boldsymbol{W}^{(1,N_C)} \\ \boldsymbol{W}^{(2,1)} & -\boldsymbol{W}^{(2,2)} & ... & \boldsymbol{W}^{(2,N_C)} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{W}^{(N_C,1)} & \boldsymbol{W}^{(N_C,2)} & ... & -\boldsymbol{W}^{(N_C,N_C)} \end{bmatrix}, \boldsymbol{W}^{(l,m)} = \boldsymbol{W}^{(m,l)}, l, m = 1, ..., N_C. \tag{3.22}$$

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. (3.20)) is similar to the extracted representation using $\tilde{\boldsymbol{K}}$ (Eq. (3.21)). For this purpose we use the following theorem.

Theorem 3.1. *(***Davis-Kahan) [31]** *Given $\bar{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ hermitian matrices where $\tilde{\boldsymbol{A}} = \bar{\boldsymbol{A}} + \widehat{\boldsymbol{B}}$ is a perturbed version of $\bar{\boldsymbol{A}}$. Set an interval S. Denote the eigenvalues within S by $\lambda_S(\bar{\boldsymbol{A}})$ and $\lambda_S(\tilde{\boldsymbol{A}})$ with a corresponding set of eigenvectors $\boldsymbol{V}_1$ and $\tilde{\boldsymbol{V}}_1$ for $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}}$, respectively. Let*

$$\delta = \min\{|\lambda_S(\tilde{\boldsymbol{A}}) - s| : \lambda_S(\tilde{\boldsymbol{A}}) \notin S, s \in S\}. \tag{3.23}$$

Then, the distance $d(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1) = ||\sin\Theta(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1)||_F \leq \frac{||\widehat{\boldsymbol{B}}||_F}{\delta}$, where $\sin\Theta(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1)$ is a diagonal matrix with the canonical angles on the diagonal.

In other words, the theorem states that the eigenspace spanned by the perturbed kernel $\hat{\boldsymbol{K}}$ is similar to the eigenspace spanned by the ideal kernel $\hat{\boldsymbol{K}}$. The distance between these eigenspaces is bounded by the ratio $\frac{||\widehat{\boldsymbol{W}}||_F}{\delta}$. 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 $\tilde{\boldsymbol{\Psi}}_\epsilon^{N_C}$.

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

$$d(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \tilde{\boldsymbol{\Psi}}_\epsilon^{N_C}) = ||\sin\Theta(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \tilde{\boldsymbol{\Psi}}_\epsilon^{N_C})||_F \leq \frac{||\widehat{\boldsymbol{W}}||_F ||\bar{\boldsymbol{D}}^{-1/2}||_F^2}{1 - \tilde{\lambda}_{N_{P+1}}}, \tag{3.24}$$

where $\widehat{\boldsymbol{W}}$ is the perturbation matrix defined in Eq. (3.21) and $\bar{D}_{i,i} = \sum_j \bar{K}_{i,j}$ of $\bar{\boldsymbol{D}}$ is a diagonal matrix whose elements are the sum of rows.

Proof. Define $\bar{\boldsymbol{A}} \triangleq \bar{\boldsymbol{D}}^{-1/2}\bar{\boldsymbol{K}}\bar{\boldsymbol{D}}^{-1/2}$. Based on Eq. 3.21 and the fact that the sum of rows in the perturbation matrix $\widehat{\boldsymbol{W}}$ is zero, we get that

$$\tilde{\boldsymbol{A}} = \bar{\boldsymbol{A}} + \bar{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{W}}\bar{\boldsymbol{D}}^{-1/2}. \tag{3.25}$$

We are now ready to use Theorem 3.1. By setting $S = [1, \lambda(\tilde{\boldsymbol{A}})_{N_C}]$, the first N_C eigenvectors of $\bar{\boldsymbol{A}}$ and $\tilde{\boldsymbol{A}}$ are denoted as $\bar{\boldsymbol{V}}_1$ and $\tilde{\boldsymbol{V}}_1$, respectively. Based on the analysis of the "ideal" matrix $\bar{\boldsymbol{P}}$, we know that its first N_C eigenvalues are equal to 1 and it implies that $\lambda_i(\bar{\boldsymbol{A}}) = 1, i = 1, ..., N_C$. Due to the decaying property of the eigenvalues $\lambda_{N_C+1}(\tilde{\boldsymbol{A}}) < 1$ and so are $\lambda_i(\bar{\boldsymbol{A}}), \lambda_i(\tilde{\boldsymbol{A}}) \in S, i = 1, ..., N_C$. Using the definition of δ from Eq. (3.23), we conclude that $\delta = 1 - \tilde{\lambda}_{N_{P+1}}$. Setting $\bar{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}} = \bar{\boldsymbol{D}}^{-1/2}\widehat{\boldsymbol{W}}\bar{\boldsymbol{D}}^{-1/2}$, the Davis-Kahan Theorem 3.1 states that the distance between the eigenspaces $\bar{\boldsymbol{V}}_1$ and $\tilde{\boldsymbol{V}}_1$ is bounded such that

$$d(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1) = ||\sin\Theta(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1)||_F \leq \frac{||\widehat{\boldsymbol{W}}||_F||\bar{\boldsymbol{D}}^{-1/2}||_F^2}{1 - \tilde{\lambda}_{N_{P+1}}}.$$

The eigen-decomposition $\bar{\boldsymbol{A}}$ 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 $\tilde{\boldsymbol{A}}$ and choosing the eigenspaces set by the first N_C eigenvectors, we get $d(\bar{\boldsymbol{V}}_1, \tilde{\boldsymbol{V}}_1) = d(\bar{\boldsymbol{\Psi}}_\epsilon^{N_C}, \tilde{\boldsymbol{\Psi}}_\epsilon^{N_C})$. □

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

$$\epsilon_{Ge} = \underset{\epsilon}{argmax}(Ge) = \underset{\epsilon}{argmax}(1 - \lambda_{N_C+1}(\tilde{\boldsymbol{P}})) \tag{3.26}$$

is optimal for classification based on a diffusion representation with N_C coordinates.

3.3.3. The Probabilistic Approach

We introduce here notation 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 $\boldsymbol{P}$, we define for any two subsets $\boldsymbol{A}, \boldsymbol{B} \subset \boldsymbol{T}$

$$P(\boldsymbol{A}, \boldsymbol{B}) = \sum_{\boldsymbol{x}_i \in \boldsymbol{A}, \boldsymbol{x}_j \in \boldsymbol{B}} P_{i,j}. \tag{3.27}$$

Given N_C classes $\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3, ..., \boldsymbol{C}_{N_C} \subset \boldsymbol{T}$, the Normalized Cut (Ncut) [32] is defined by the following measure

$$Ncut(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}) = \sum_{l=1}^{N_C} P(\boldsymbol{C}_l, \bar{\boldsymbol{C}}_l). \tag{3.28}$$

In clustering problems, a partition is searched such that the Ncut is minimized [33, 34]. We use this intuition for a more relaxed classification problem. We now define a generalized Ncut using the following matrix

$$\widehat{P}_{i,j} = \begin{cases} \tilde{P}_{i,j}, & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases}, \tag{3.29}$$

and the generalized Ncut as $GNcut(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}) \triangleq \sum_{l=1}^{N_C} \widehat{P}^{(\epsilon)}(\boldsymbol{C}_l, \bar{\boldsymbol{C}}_l)$. The idea is to remove the probability of "staying" at a specific node from the within class transition probability. Let

$$\rho_P \triangleq 1 - GNcut(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}). \tag{3.30}$$

We search for ϵ which maximizes ρ_P such that

$$\epsilon_{\rho_P} = \underset{\epsilon}{arg\max}(\rho_P). \tag{3.31}$$

Proposition 3.3.2. *By maximizing ρ_P, the sum of within class transition probability is maximized.*

Proof. By the enforced stochastic model, the transition probability between point x_i and point x_j is equal to $p(x_i, x_j) = P_{i,j}$, therefore
$\rho_P \triangleq 1 - cut(\boldsymbol{C}_1, ..., \boldsymbol{C}_{N_C}) = 1 - \sum_{l=1}^{N_C} \widehat{P}^{(\epsilon)}(\boldsymbol{C}_i, \bar{\boldsymbol{C}}_i) = \sum_{l=1}^{N_C} \widehat{P}^{(\epsilon)}(\boldsymbol{C}_l, \boldsymbol{C}_l) =$
$\sum_{l=1}^{N_C} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_i \in \boldsymbol{C}_l, i \neq j} \frac{exp\{-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\epsilon}\}}{\sum_j exp\{-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\epsilon}\}}$. The last term represents the sum of within the class transition probability. □

The heuristic approach presented in Eq. 3.31 provides yet another criterion for setting a scale parameter which captures the geometry of the given classes.

4. Experimental Results

4.1. Manifold Learning

In this section, we evaluate the performance of the proposed approach by embedding a low dimensional manifold which lies in a high dimensional space. The experiment is constructed by projecting a 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

$$\boldsymbol{Y}_i = \begin{bmatrix} {y_i}^1 \\ {y_i}^2 \\ {y_i}^3 \end{bmatrix} = \begin{bmatrix} 6\theta_i \cos(\theta_i) \\ h_i \\ 6\theta_i \sin(\theta_i) \end{bmatrix}, i = 1, ..., 2000, \tag{4.1}$$

  where $\theta_i, h_i, i = 1, ..., 2000$, are drawn from a uniform distributions within the intervals $[\frac{3\cdot\pi}{2}, \frac{9\cdot\pi}{2}], [0, 100]$, respectively.

- We project the Swiss roll $\boldsymbol{Y}$ 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 concatenate the projected Swiss Roll with Gaussian noise as follows

$$\boldsymbol{X}_i = \begin{bmatrix} \boldsymbol{N}_T \cdot \boldsymbol{Y}_i \\ \boldsymbol{N}_i^1 \end{bmatrix}, i = 1, ..., 2000. \tag{4.2}$$

  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.

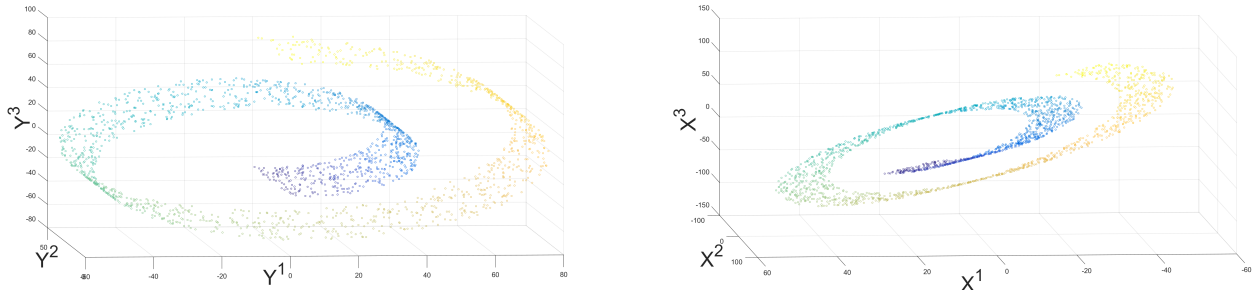


Figure 4.1: Left: "clean" Swiss Roll ($\boldsymbol{Y}$ in Eq. (4.1)). Right: 3 coordinates of the projected Swiss Roll ($\boldsymbol{X}$ in Eq. (4.2)). Both figures are colored by the value of the underlying parameter $\theta_i, i = 1, ..., 2000$ (Eq. (4.1)).

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

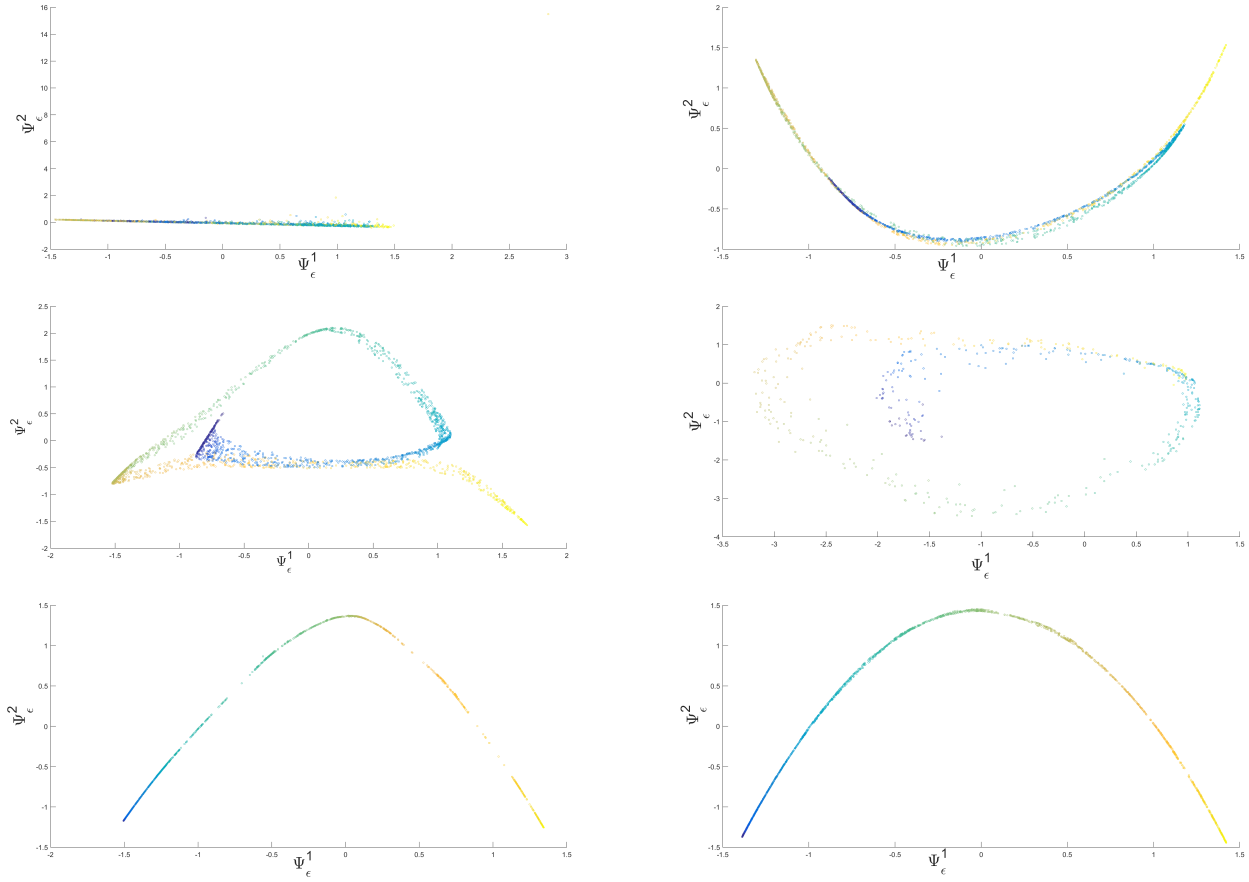


Figure 4.2: Extracted DM-based embedding using different methods for choosing the scale parameter ϵ. Top left: standard deviation scalings, the matrix A and scaling ϵ_{std} are computed by Eq. (3.5). Top right: the ϵ_0 scaling, the calculation of ϵ_0 is described in Alg. (3.1) and [11]. Mid left: the MaxMin scaling, the value ϵ_{MaxMin} is defined by Eq. (3.1). Mid right: K-NN based scaling [12]. 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 [11] applied to the clean Swiss roll Y that is defined by Eq. (4.1).

A high dimensional dataset $\boldsymbol{X}$ is generated from various values of σ_N,σ_{N_T}, D_1 and D_2. DM is applied to $\boldsymbol{X}$ using:

- The standard deviation normalization as defined in Eq. (3.5).
- The ϵ_0 scale, which is described in 3.1 and in [11].
- The MaxMin scale parameter, which is defined in Eq. 3.1 and in [24].
- The K-NN based scaling [12].
- The proposed scale parameters $\boldsymbol{A}, \epsilon$, which are presented in Algorithm 3.2.

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

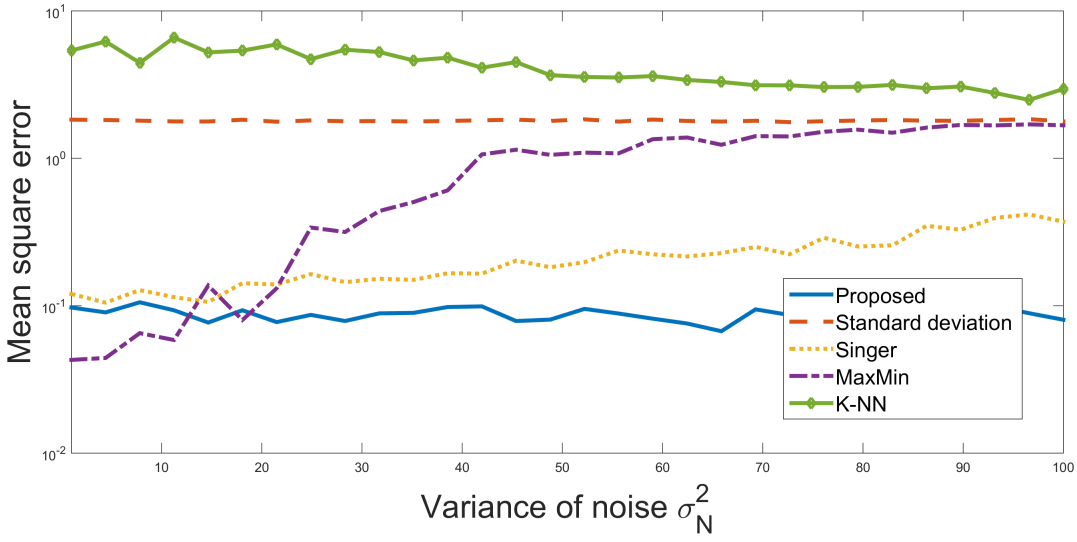


Figure 4.3: The mean square error of the extracted embedding. A comparison between the proposed normalization and alternative methods which are detailed in Section 3.

Each embedding is computed using an eigen-decomposition, therefore, the embedding's coordinates could be the same up to scaling and rotation. To overcome this problem, 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 following error term

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

This is 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 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. 4.2. The MSE is presented in Fig. 4.3. It is evident that Algorithm 3.2 is able to extract a more precise embedding than the alternative normalizations. 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.

4.2. Classification

In this section, we provide empirical support for the theoretical analysis from Section (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 $\rho_P, \rho_{\boldsymbol{\Psi}}$, Ge (Eqs. 3.31, 3.17, 3.26, respectively) are correlated with the quality of the classification.

4.2.1. Classification of a Gaussian Mixture

We first generate two classes using a Gaussian mixture. The following steps describe the generation of the data:

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 center of masses for the generated classes $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$.
2. 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 200 data points as by $\boldsymbol{C}_1 \cup \boldsymbol{C}_2 = \boldsymbol{T} \subset \mathbb{R}^6$.

An example of 3-dimensions from this type of dataset is presented in Fig. 4.4. DM is applied to a six dimensional training set $\boldsymbol{T}$ and a 2-dimensional representation $\boldsymbol{\Psi}^2_\epsilon(\boldsymbol{T})$ is extracted. As presented in Fig. 4.4 (right), the 2-dimensional embedding encapsulates the structure of the classes, whereas Fig. 4.4 (left) presents three coordinates from the original ambient space.

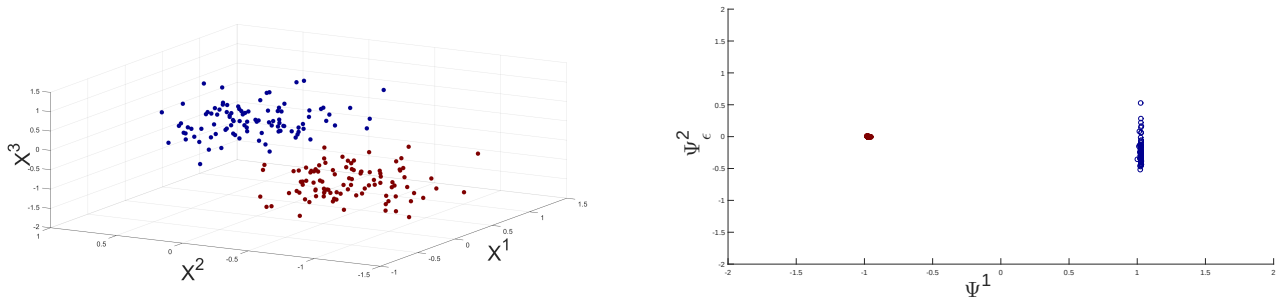


Figure 4.4: Left: an example of the Gaussian distributed data points. Right: a 2-dimensional mapping of the data points.

The first experiment is dedicated to the ideal case (Section 3.3). 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 4.5 (left), we present the first extracted diffusion coordinate using various values of ϵ. It is evident that the classes separation is highly influenced by ϵ. A comparison between ρ_P,ρ_Ψ and Ge is presented in Fig. 4.5 (right). This comparison provides evidence of the high correlation between ρ_P (Eq. 3.31), ρ_Ψ (Eq. 3.17) and the generalized eigengap (Eq. 3.26).

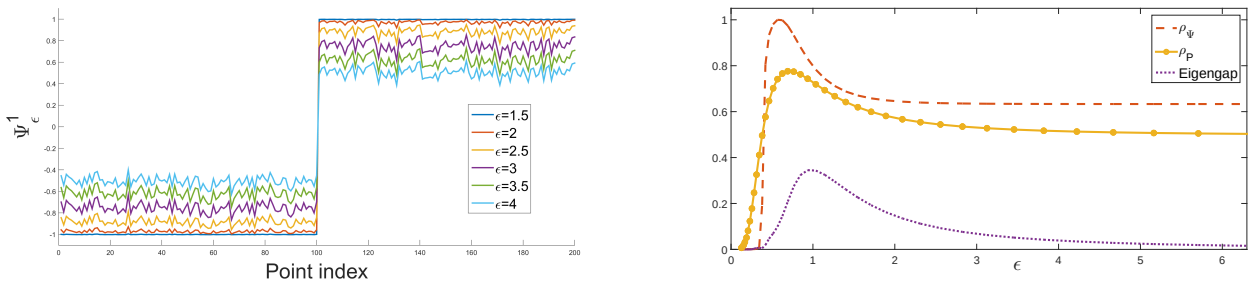


Figure 4.5: Left: the first eigenvector $\boldsymbol{\Psi}^1$ computed for various values of ϵ. Right: a comparison between ρ_P, ρ_Ψ and Ge.

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 assumed to model an unknown underlying nonlinear physical process. The following describes how these classes are generated:

1. Set the number of classes N_C and a gap parameter G. Each class $\boldsymbol{C}_l, l = 1, ..., N_C$, consists of $N_P = 100$ data points drawn from a uniformly dense distribution within the line $[(l-1) \cdot L_C, l \cdot L_C - G], l = 1, ..., N_C$. L_C is the class length that it is set such that $L_C = \frac{1}{N_C}$.
2. Denote $r_i \in \mathbb{R}^1, i = 1, ..., N_C \cdot N_P$, as the unity of all 1-dimensional points from all classes $C_l, l = 1, ..., N_C$.
3. Project 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 $\boldsymbol{N}_i^2 \in \mathbb{R}^3, i = 1, ...N_C \cdot N_P$, are i.i.d drawn from a Gaussian distribution with zero mean and a covariance matrix $\boldsymbol{\Lambda} = \sigma_S \cdot \boldsymbol{I}$. Two examples of the spiral-based classes are presented in Fig. 4.6. For both examples, we use $N_C = 4, N_P = 100, \sigma_S = 0.4$ with different values for the gap parameter G.

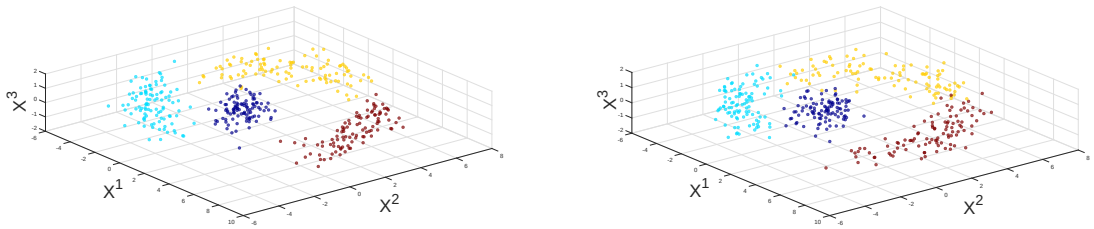


Figure 4.6: Two examples of the generated three-dimensional spiral that are based on Eq. (4.6) using $N_C = 4$ classes with $N_P = 100$ data points within each class. The gaps are set to be $G = 0.02, 0.04$ left and right, respectively.

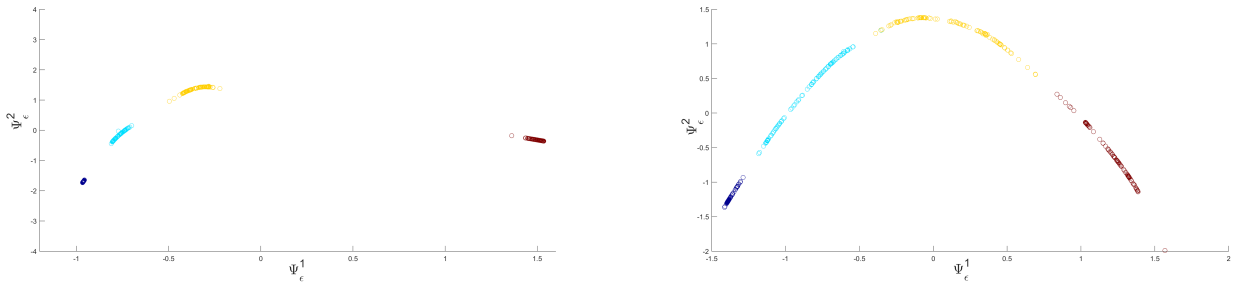


Figure 4.7: A 2-dimensional mapping extracted from both spirals presented in Fig. (4.6).

To evaluate the advantage of the proposed scale parameters ϵ_Ψ and ϵ_P (Eqs. 3.17 and 3.31, respectively) that are applied to classification tasks, we calculate the ratios ρ_P and ρ_Ψ for various values of ϵ, and then we compute the classification results performed in the low-dimensional embedding. Examples of embeddings which are extracted from the spirals in Fig. 4.6 are presented in Fig. 4.7. The classification results are measured based on Leave-One-Out cross validation. Classification in the ambient space is performed using K-NN ($K = 1$). It is evident from Fig. (4.8) that the classification results in the ambient space are highly influenced by the scale parameter ϵ. Furthermore, peak classification results occur at a scale parameter ϵ, which corresponds to the maximal values of ρ_P and ρ_Ψ.

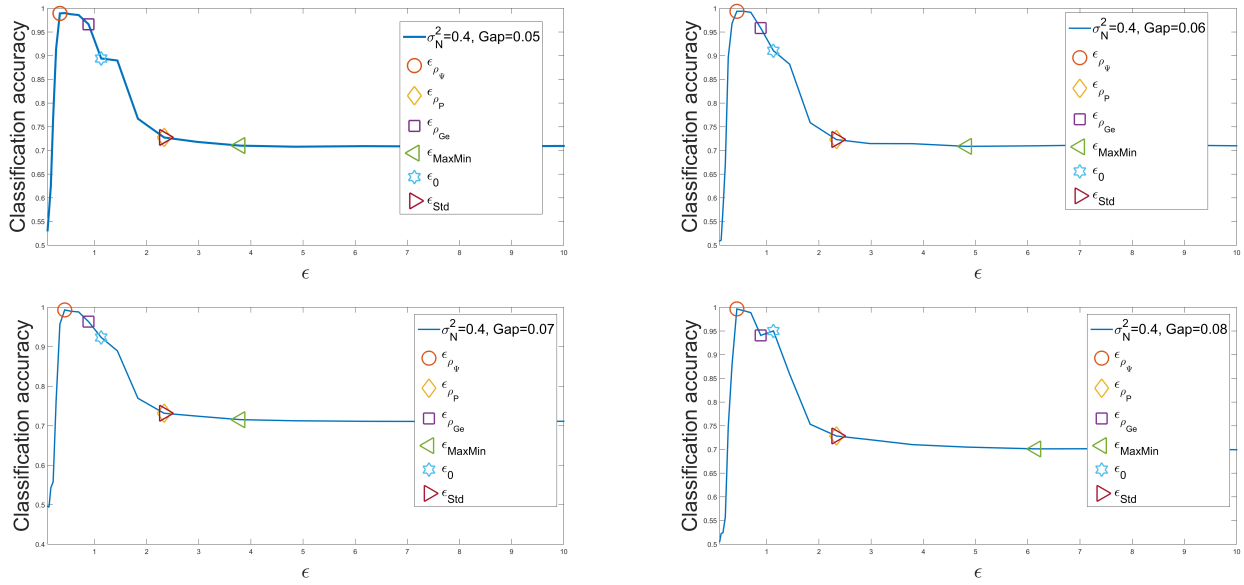


Figure 4.8: Accuracy of classification in the spiral artificial dataset for different values of the gap parameter G. The data is generated based on Eq. 4.6. The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots.

4.2.2. Classification of Handwritten Digits

In the following experiment, we use the dataset from the UCI machine learning repository [35]. The dataset consists of 2000 data points describing 200 data points 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 the Zerkine moment (ZER), morphological (MOR), profile correlations (FAC) and the Karhunen-love coefficients (KAR) as our features space denoted by $\boldsymbol{X}^1, \boldsymbol{X}^2, \boldsymbol{X}^3, \boldsymbol{X}^4$, respectively.

We compute the proposed ratios ρ_P **and** ρ_Ψ for various values of ϵ, and estimate the optimal scale based on Eqs. 3.17, 3.31. We evaluate the extracted embedding by performing 20-fold cross validation (5% left out as a test set). The classification is done by applying K-Nearest Neighbors in the d-dimensional embedding. In Fig. 4.9, we present the classification results and the proposed optimal scales ϵ for classification. The proposed scale intersects with the scale that provides maximal classification rate.

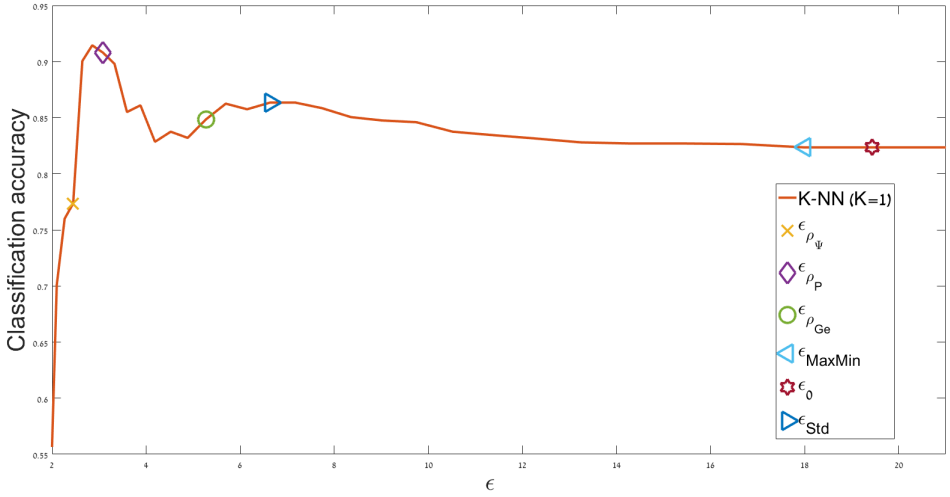


Figure 4.9: Accuracy of classification in the multiple features dataset. K-NN ($K = 1$) is applied in a $d = 4$ dimensional diffusion based representation. The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots.

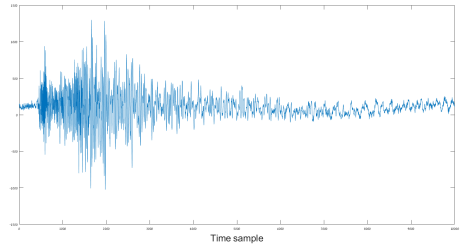
4.2.3. Classification of seismic events

Discrimination between earthquakes and explosions is a critical component in nuclear test monitoring and it is also critical for creating reliable earthquake catalogs. Recently, the problem has been approached using machine learning frameworks such in [36, 37, 38, 39]. The DM framework is utilized in [40] to extract a low dimensional representation for the seismic events.

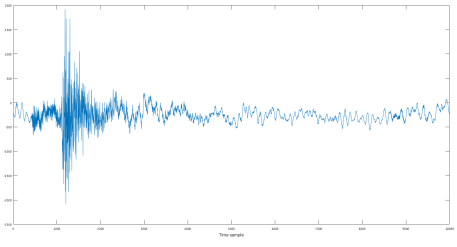
In the following experiment, we use a dataset with 46 earthquakes and 62 explosions that were recorded in Israel. The collected waveforms occurred in the Dead Sea area between the years 2004-2015. The waveforms were manually annotated by a specialist from the Geophysical Institute of Israel (GII). The feature extracted from each waveform is a Sonogram [41] with some modifications. The Sonogram is basically a time-frequency representation equally tempered on a logarithmic scale. The feature extraction process is detailed in [40].

Both the raw data and the extracted features of an explosion and an earthquake are

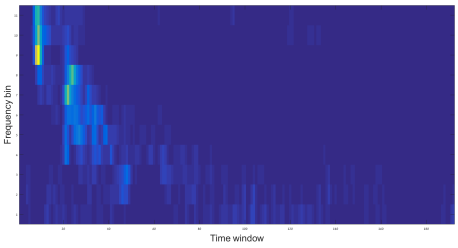
presented in Fig. 4.10. A low dimensional mapping is extracted by using DM with various values of ϵ. A binary classification is performed using K-NN ($K = 5$) in a leave-one-out fashion. The accuracy of the classification for each value of ϵ is presented in Fig. 4.11. The estimated values of $\epsilon_{Ge}, \epsilon_{\rho_P}$ and $\epsilon_{\rho_{Psi}}$ were annotated. It is evident that for classification the estimated values are indeed close to the optimal values. The values do not intersect with the optimal value. However, they all achieve high classification accuracy.



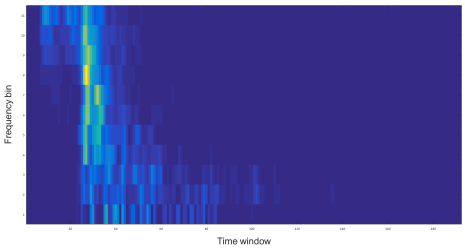
(a) Seismic recording of an explosion



(b) Seismic recording of an earthquake



(c) Sonogram of an explosion



(d) Sonogram of an earthquake

Figure 4.10: 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.

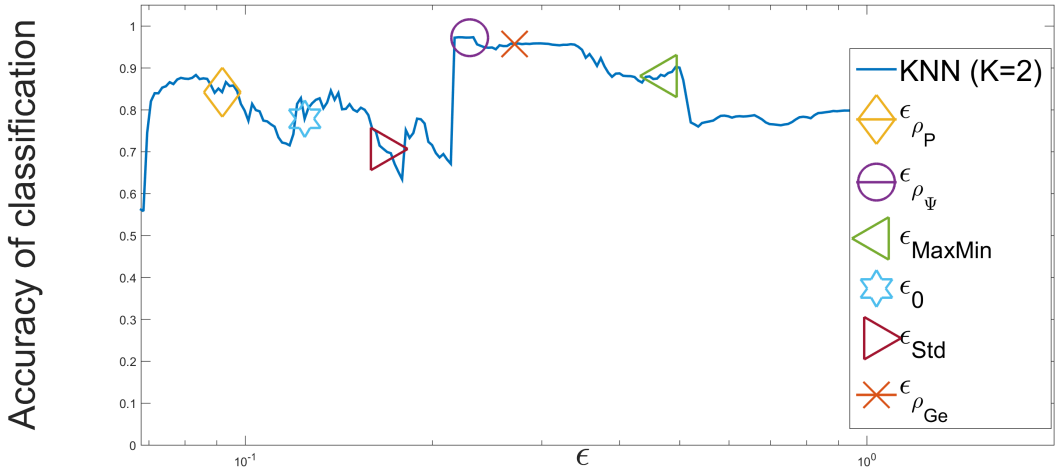


Figure 4.11: Classification accuracy vs. value of ϵ. The proposed scales $(\epsilon_\Psi, \epsilon_{Ge}, \epsilon_P)$ and existing methods $(\epsilon_0, \epsilon_{\text{MaxMin}}, \epsilon_{std})$ are annotated on the plots.

5. Conclusions

In this paper, we presented two new frameworks for setting the scale parameter for kernel based dimensionality reduction. The first approach is useful when the high dimensional data points lie on some lower dimensional manifold. Theoretical justification and simulations on artificial data demonstrate the strength over alternative schemes. The second approach is quiet intuitive and easy to implement could improve the classification results for application that uses kernel based dimensionality reduction. We aim to generalize the suggested approach in this paper to a multi-view scenario as was studied by [42, 43, 44].

Acknowledgment

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.

[1] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.

[2] J. B. Kruskal and W. M, "Multidimensional scaling," *Sage Publications. Beverly Hills*, 1977.

[3] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[4] S. T. Roweis and L. K. Sau, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290.5500, pp. 2323–2326, 200.

[5] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *NIPS*, vol. 14, 2001, pp. 585–591.

[6] R. R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, vol. 21, pp. 5–30, 2006.

[7] W. Luo, "Face recognition based on laplacian eigenmaps," 2011, pp. 416 – 419.

[8] O. Lindenbaum, A. Yeredor, and I. Cohen, "Musical key extraction using diffusion maps," *Signal Processing*, vol. 117, pp. 198–207, 2015.

[9] T. Lin, H. Zha, and S. U. Lee, "Riemannian manifold learning for nonlinear dimensionality reduction," in *European Conference on Computer Vision*. Springer, 2006, pp. 44–55.

[10] S. Lafon, Y. Keller, and R. R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1784–1797, 2006.

[11] A. Singer, R. Erban, I. Kevrekidis, and R. R. Coifman, "Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps," vol. 106, no. 38, 2009, pp. 16 090–16 095.

[12] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems*, 2004, pp. 1601–1608.

[13] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[14] P. Gaspar, J. Carbonell, and J. L. Oliveira, "On the parameter optimization of support vector machines for binary classification," *J Integr Bioinform*, vol. 9, no. 3, p. 201, 2012.

[15] C. Staelin, "Parameter selection for support vector machines," *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1*, 2003.

[16] C. Campbell, N. Cristianini, and J. Shawe-Taylor, "Dynamically adapting kernels in support vector machines," *Advances in neural information processing systems*, vol. 11, pp. 204–210, 1999.

[17] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 131–159, 2002.

[18] C. Ceruti, S. Bassis, A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli, "Danco: Dimensionality from angle and norm concentration," *arXiv preprint arXiv:1206.3881*, 2012.

[19] K. Fukunaga and D. R. Olsen, "An algorithm for finding intrinsic dimensionality of data," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 176–183, 1971.

[20] P. J. Verveer and R. P. W. Duin, "An evaluation of intrinsic dimensionality estimators," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 17, no. 1, pp. 81–86, 1995.

[21] G. V. Trunk, "Stastical estimation of the intrinsic dimensionality of a noisy signal collection," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 165–171, 1976.

[22] K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," *IEEE Transactions on pattern analysis and machine intelligence*, no. 1, pp. 25–37, 1979.

[23] F. Camastra, "Data dimensionality estimation methods: a survey," *Pattern recognition*, vol. 36, no. 12, pp. 2945–2954, 2003.

[24] S. Lafon, Y. Keller, and R. Coifman, "Data fusion and multicue data matching by diffusion maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28 no. 11, p. 17841797, 2006.

[25] R. R. Coifman, Y. Shkolnisky, F. J. Sigworth, and A. Singer, "Graph Laplacian Tomography From Unknown Random Projections," *Image Processing, IEEE Transactions on*, vol. 17, no. 10, pp. 1891–1899, Oct. 2008.

[26] M. Hein and J.-Y. Audibert, "Intrinsic dimensionality estimation of submanifolds in r d," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 289–296.

[27] I. Cohen, Q. Tian, X. S. Zhou, and T. S. Huang, "Feature selection using principal feature analysis," *Univ. of Illinois at Urbana-Champaign*, 2002.

[28] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 301–304.

[29] F. Song, Z. Guo, and D. Mei, "Feature selection using principal component analysis," in *System Science, Engineering Design and Manufacturing Informatization (ICSEM), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 27–30.

[30] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849–856, 2002.

[31] G. W. Stewart, *Matrix perturbation theory*. Citeseer, 1990.

[32] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[33] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 551–556.

[34] C. H. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering." in *SDM*, vol. 5. SIAM, 2005, pp. 606–610.

[35] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[36] M. Beyreuther, C. Hammer, M. Wassermann, M. Ohrnberger, and M. Megies, "Constructing a hidden markov model based earthquake detector: Application to induced seismicity," *Geophysical Journal International*, vol. 189, pp. 602–610, 2012.

[37] C. Hammer, M. Ohrnberger, and D. F$'$ah, "Classifying seismic waveforms from scratch: A case study in the alpine environment," *Geophysical Journal International*, vol. 192, pp. 425–439, 2013.

[38] E. Del Pezzo, A. Esposito, F. Giudicepietro, M. Marinaro, M. Martini, and S. Scarpetta, "Discrimination of earthquakes and underwater explosions using neural networks," *Bulletin of the Seismological Society of America*, vol. 93, no. 1, pp. 215–223, 2003.

[39] T. Tiira, "Discrimination of nuclear explosions and earthquakes from teleseismic distances with a local network of short period seismic stations using artificial neural networks," *Physics of the earth and planetary interiors*, vol. 97, no. 1-4, pp. 247–268, 1996.

[40] N. Rabin, Y. Bregman, O. Lindenbaum, Y. Ben-Horin, and A. Averbuch, "Earthquake-explosion discrimination using diffusion maps," *Geophysical Journal International*, vol. 207, no. 3, pp. 1484–1492, 2016.

[41] M. Joswig, "Pattern recognition for earthquake detection," *Bulletin of the Seismological Society of America*, vol. 80, no. 1, pp. 170–186, 1990.

[42] O. Lindenbaum, A. Yeredor, M. Salhov, and A. Averbuch, "Multiview diffusion maps," *arXiv preprint arXiv:1508.05550*, 2015.

[43] M. Salhov, O. Lindenbaum, A. Silberschatz, Y. Shkolnisky, and A. Averbuch, "Multi-view kernel consensus for data analysis and signal processing," *arXiv preprint arXiv:1606.08819*, 2016.

[44] R. R. Lederman and R. Talmon, "Common manifold learning using alternating-diffusion," submitted, Tech. Report YALEU/DCS/TR1497, Tech. Rep., 2014.