Stockholms universitet

Diffusion map - A nonlinear dimensionality reduction method

Thi Thuy Nga Nguyen

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2020:6
Matematisk statistik
Juni 2020

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2020:6**
http://www.math.su.se

Diffusion map - A nonlinear dimensionality reduction method

Thi Thuy Nga Nguyen*

June 2020

Abstract

In this report, we discuss the diffusion map, a nonlinear dimensionality reduction method, which focuses on discovering the underlying geometrical structure of data. We first consider a given dataset as a weighted graph, then construct a Markov chain on this graph. The transition matrix P, expressed by a Gaussian kernel function with a single parameter σ - kernel width, characterizes the local connectivity on the graph. When running the Markov chain forward in time, P^t integrates the local connectivities to describe the global connectivity, leading to the idea of diffusion distance viewed as the average length of all paths connecting two nodes in the weighted graph. The diffusion distance defined from P^t thus contains the information of the intrinsic data geometry. We then use eigen-decomposition of P^t to define diffusion coordinates and the diffusion map. This map reorganizes the data according to the diffusion distance in which the points are close if they are highly connected in the weighted graph. Moreover, the diffusion map embeds the data in a lower-dimensional space where the Euclidean distance is an approximation of the diffusion distance.

Diffusion map is very robust to noise and easy to implement. Nevertheless, it is not trivial to choose the parameters appropriately: the kernel width σ, the timescale t, and the dimension of the embedding space q. While the kernel width relates to how transition probabilities describe local connectivity, the timescale t affects the ability of the diffusion distances to capture global connectivity, and the dimension q characterizes the intrinsic dimension of the representation we would like to discover. All of them are investigated by toy examples in our report.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: tthnga.nguyen@gmail.com. Supervisor: Chun-Biu Li.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Chun-Biu Li, for his advise, patience, guidance, encouragement as well as his enthusiasm and confidence. I am grateful to him for precious helping in proofreading and correcting my report. This report would not have been possible without his support.
I would also like to thank my friends, Marina and Arthur, for their willingness to read and provide valuable comments on this report. Special thanks to Marina for having accompanied and listened to me throughout my studies.
I appreciate the understanding and constant support given to me by my husband. Finally, my thanks go to my parents and my brother for their unfailing love.

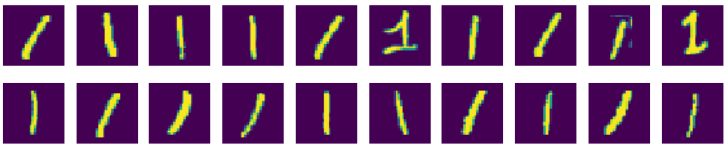
Contents

1 Introduction

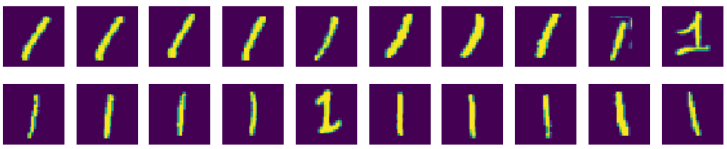
Today, we face a lot of complex and high-dimensional data under the rapid development of instrumentation technologies. Working with these data problems come out: difficulty in visualizing, high computational cost, sparseness, locality, boundary, and high bias [5, 10]. Dimensionality reduction is one common strategy to resolve these issues.

The goal of dimensionality reduction is to find a low-dimensional representation of a high-dimensional data capturing essential geometrical features. For example, an image of a handwriting digit 1[1] has a size of 28×28, and its dimension is thus 784. Observing Figure 1a, these images display digit 1 with different angles. Therefore, we can reorganize them by their angle, as shown in Figure 1b, where we need only one variable. Now, if we have similar images of ten digits from 0 to 9, an intuitive organization is to divide them into ten groups of the digit, and in each group, the images are arranged according to the angle of each digit. With this organization, we need only two variables, digit and angle, to express the important information of the data.

There are a lot of methods developed to solve the problem of finding the lower-dimensional description of the data. They are usually classified into linear and nonlinear methods. A linear method finds a linear transformation of the data and normally preserves the Euclidean distance between points such as Principal Component Analysis (PCA). However, when a data structure is nonlinear, the Euclidean distance between points, when it is large, provides very little information about the disconnection between data points. It thus fails to describe the underlying structure of data. Diffusion map [4], otherwise, is a nonlinear method preserving diffusion distance which is small when the points are connected via many points between them and large when they are disconnected. This distance better expresses the disconnection between points. The diffusion map is a topology-preserving method [6] which preserves the neighborhood relationships between subgroups of the data, similar to Locally Linear Embedding (LLE), Laplacian eigenmaps (LE) (see [11] for more details about these methods). The diffusion map is the main focus of our report.



(a)



(b)

Figure 1: (a) Images of digit 1 and (b) a reorganization by angle.

The report's outline is as follows. Section 2 gives an overview of two well-known methods for dimensionality reduction: PCA and Multidimensional Scaling (MDS). In Section 3, we introduce and explain the diffusion map. In Section 4, computational experiments on some toy examples are performed and discussed. Section 5 presents the conclusion. Finally, the discussion of potential studies is given in Section 6.

2 Linear dimensionality reduction methods

Linear dimensionality reduction methods are usually performed under the assumption that the data are linearly embedded into a higher-dimensional space. Finding out this linear transformation

[1] from MINST in `http://yann.lecun.com/exdb/mnist/`

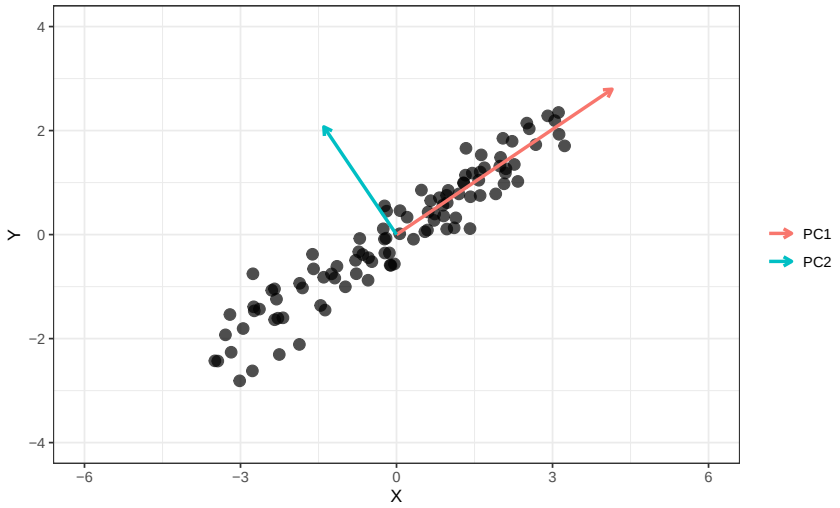


Figure 2: A example of PCA. A sample of 100 (black) points distributed along a straight line with adding the noise and the principal components. The first component (the red arrow) describes the data in the largest variance direction. The second component (the blue arrow) is orthogonal to the first one.

gives a lower-dimensional representation. Among these methods, PCA is very well popular. MDS less so, however, it has some similarities with PCA: both of these methods are solved by using eigenanalysis, which is also used in the construction of the diffusion map.

Let us assume that our data have n observations, and each of them contains d features. While PCA is solved on the data covariance matrix (of size $d \times d$) in feature space, MDS works with a distance matrix (of size $n \times n$) in observation space. Like MDS, the diffusion map is constructed by using the eigen-decomposition of a transition matrix, the row-normalization of a distance matrix in observation space, of size $n \times n$. Therefore, by comparing to MDS, we can get some insights regarding the eigenvalues and eigenvectors defined in the diffusion map process.

2.1 Principal Component Analysis (PCA)

PCA is the most common method in dimensionality reduction. The purpose of PCA is to find a new basis, which is a linear combination of the original basis, that best expresses the data. With the assumption that the directions of the large variance contain important information about the data while the directions of small variance correspond to the noise, the method transforms the original data to a new representation by rotating its coordinate system to capture the variance maximally.

To have a general overview of PCA, let us look at an example in Figure 2. In this example, a set of 100 points in a 2-dimensional plane is considered. These points are distributed along a line with a noise addition. Note that in real situations, noise can be incurred during data measurement, collection, or storage. By this construction, the data are actually defined in one dimension, which corresponds to the largest variance direction. This direction does not lie along with the basis of the data points (x, y) but rather lies along the best fit line. After using PCA, the basis is changed by rotating so that the projection of the data points on the first principal component has the largest possible variance. The second component on which the data are projected is orthogonal to the first one and has a very small variance. In this case, the second component can be considered as noise and neglected. The dimensionality can be reduced to one. In the next section, we will see how PCA is formulated mathematically.

2.1.1 Principal components

Consider a dataset $\boldsymbol{X}$ where each observation $\boldsymbol{x} \in \mathbb{R}^d$ is a random vector of d variables. Suppose that the dataset has n observations, $\boldsymbol{X}$ denotes a matrix of size $n \times d$. PCA helps to find a new representation $\boldsymbol{Y}$, a matrix of size $n \times m$, in which each element $\boldsymbol{y} \in \mathbb{R}^m$ has $m < d$ variables such that

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{W}. \tag{1}$$

The columns $\{\boldsymbol{w}_i\}_{i=1,\dots,m}$ of matrix $\boldsymbol{W}_{d\times m}$ are a set of new basis vectors. By this transformation, $\boldsymbol{y}_i$ is a projection of $\boldsymbol{x}_i$ onto the new basis $\{\boldsymbol{w}_i\}_{i=1,\dots,m}$.

The variables in the data, in many cases, contain important information and the noise. We assume that important information has a large variance, while noise has a small variance. In addition, there is an extra factor in the data - redundancy - represented by the covariance between variables [16]. When two variables are strongly correlated (as the example shown in Figure 2), corresponding to high linear redundancy, they describe the same information of the data. The goal of PCA is to find out the appropriate rotation of the basis of the original data which maximizes the variance describing the important information and minimizes the redundancy measured by the magnitude of the covariance. Thus, the optimal covariance matrix is diagonal, where the diagonal entries are ordered decreasing according to the variance, and the off-diagonal entries are all zero. With this matrix, the important information is described by the large variance directions while the noise corresponds to the small variance.

Nevertheless, the covariance matrix is usually unknown in reality where we only measure the data. For this reason, the unbiased sample covariance matrix $\boldsymbol{C}_{\boldsymbol{X}}$ can be considered. For the sake of simplicity, assuming that the data $\boldsymbol{X}$ is centered, that means $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i = \boldsymbol{0}_d$, where $\boldsymbol{0}_d$ is vector of d zero entries, the unbiased estimate of the covariance matrix is

$$\boldsymbol{C}_{\boldsymbol{X}} = \frac{1}{n-1}\boldsymbol{X}^\top\boldsymbol{X}.$$

Note that the centralization can be done in data preprocessing. $\boldsymbol{C}_{\boldsymbol{X}}$ is a square symmetric $d \times d$ matrix where the diagonal entries are the variance of a particular variable, and the off-diagonal entries are the covariance between d variables. Since $\boldsymbol{Y}$ is centered after the transformation (1) when $\boldsymbol{X}$ is centered, the covariance matrix of the new representation $\boldsymbol{Y}$ can be similarly estimated by

$$\boldsymbol{C}_{\boldsymbol{Y}} = \frac{1}{n-1}\boldsymbol{Y}^\top\boldsymbol{Y},$$

where $\boldsymbol{C}_{\boldsymbol{Y}}$ is a square symmetric $m \times m$ matrix. For the goal of PCA, the optimal covariance matrix of $\boldsymbol{Y}$ is diagonal, i.e. all off-diagonal entries are zero and all diagonal entries should be ordered descendingly according to variance. To diagonalize the covariance matrix $\boldsymbol{C}_{\boldsymbol{Y}}$ an intuitive method is assuming the basis vectors $\{\boldsymbol{w}_i\}_{i=1,\dots,m}$ are orthonormal, that means $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is $m \times m$ identity matrix. From this assumption, the solution of PCA in a natural way is a set of eigenvectors of the covariance matrix, which we will see below.

Using the transformation (1), the relationship between $\boldsymbol{C}_{\boldsymbol{Y}}$ and $\boldsymbol{C}_{\boldsymbol{X}}$ is

$$\boldsymbol{C}_{\boldsymbol{Y}} = \frac{1}{n-1}\boldsymbol{Y}^\top\boldsymbol{Y} = \frac{1}{n-1}\boldsymbol{W}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{W} = \boldsymbol{W}^\top\left(\frac{1}{n-1}\boldsymbol{X}^\top\boldsymbol{X}\right)\boldsymbol{W} = \boldsymbol{W}^\top\boldsymbol{C}_{\boldsymbol{X}}\boldsymbol{W}.$$

Since the covariance matrix $\boldsymbol{C}_{\boldsymbol{X}}$ is symmetric with all entries are real numbers there exists an eigen-decomposition

$$\boldsymbol{C}_{\boldsymbol{X}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, \tag{2}$$

where $\boldsymbol{\Lambda}_{d\times d}$ is a diagonal matrix whose entries are eigenvalues and the columns of $\boldsymbol{Q}_{d\times d}$ are orthonormal eigenvectors of $\boldsymbol{C}_{\boldsymbol{X}}$, i.e. $\boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{I}_d$. Moreover, since a covariance matrix is positive semi-definite, its eigenvalues are all real numbers and non-negative. Now assuming that the entries of $\boldsymbol{\Lambda}$ are in decreasing order, an intuitive solution for PCA problem is

$$\boldsymbol{W} \equiv \boldsymbol{Q}\boldsymbol{I}_{d\times m}, \tag{3}$$

where $\boldsymbol{I}_{d\times m}$ is the $d \times m$ identity matrix. This choice satisfies the assumption of $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_m$ and diagonalizes the covariance matrix of $\boldsymbol{Y}$ because

$$\boldsymbol{C_Y} = \boldsymbol{I}_{m\times d}\boldsymbol{Q}^\top \boldsymbol{C_X} \boldsymbol{Q}\boldsymbol{I}_{d\times m} = \boldsymbol{I}_{m\times d}\boldsymbol{Q}^\top \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top \boldsymbol{Q}\boldsymbol{I}_{d\times m} = \boldsymbol{I}_{m\times d}\boldsymbol{\Lambda}\boldsymbol{I}_{d\times m}.$$

The eigenvalues represent the variance of the data projection on the new basis defined by the orthonormal eigenvectors of $\boldsymbol{C_X}$. The column vectors of $\boldsymbol{Q}$, indicating the direction of the new representation $\boldsymbol{Y}$, define the *principal components*. And the identity matrix $\boldsymbol{I}_{d\times m}$ allows keeping m among d dimensions of the original data as follows

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Q}\boldsymbol{I}_{d\times m},$$

where $m \leq d$.

Note that, if $m < d$, $\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{I}_m$ but $\boldsymbol{W}\boldsymbol{W}^\top \neq \boldsymbol{I}_d$ because the rank of matrix $\boldsymbol{W}$ is m. In other words, the rows of $\boldsymbol{W}$ are not linearly independent. Additionally, $\boldsymbol{W}^\top$ is the generalized inverse [2] of $\boldsymbol{W}$ (since $\boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{W} = \boldsymbol{W}$) that implies the reconstruction $\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{W}^\top$. The columns of $\boldsymbol{W}^\top$ thus form the basis of $\boldsymbol{X}$. Their dependence shows that dataset $\boldsymbol{X}$ is embedded into a higher dimensional space than its intrinsic space. Discovering the intrinsic space is the goal of not only PCA but also all dimensionality reduction methods.

One problem of PCA is to determine m, the number of retaining dimension. If the linear transformation in (1) exists, only the first m eigenvalues of Λ are non-zero. m is thus the number of non-zeros eigenvalues of the covariance matrix $\boldsymbol{C_X}$. However, in real situations, there are some noises in the observations as in the example in Figure 2. Choosing m, in this case, becomes more difficult. There are some shoosing methods, such as observing the gap (sudden fall) in the plot of the eigenvalues or using a threshold of neglected variance [11].

In summary, PCA gives a new representation $\boldsymbol{Y}$ of the dataset $\boldsymbol{X}$ by rotating the original basis under the following assumptions:

- The transformation is linear: we can define a new basis of linearly independent vectors expressing the data in small representation space.
- The directions of large variance describe the data's important information while the ones with small variance represent the noise. That implies that only the principal components corresponding to the large variance are interesting.
- The principal components are orthonormal, which means the vectors form the new basis is orthogonal and of unit one. That is an intuitive way that makes PCA solvable using linear algebra.

2.1.2 PCA and Singular Value Decomposition (SVD)

Another perspective of PCA can be seen by using SVD. Consider matrix

$$\boldsymbol{X}' = \frac{1}{\sqrt{n-1}}\boldsymbol{X};$$

it is following that the covariance matrix of $\boldsymbol{X}$ can be computed by

$$\boldsymbol{C_X} = \boldsymbol{X'}^\top \boldsymbol{X}'.$$

Now let us consider the SVD of the real matrix $\boldsymbol{X'}$ of size $n \times d$ given by

$$\boldsymbol{X}' = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{V}^\top,$$

where $\boldsymbol{\Sigma}'$ is an $n \times d$ rectangular matrix with non-negative real numbers on the diagonal, $\boldsymbol{U}$ is $n \times n$ and $\boldsymbol{V}$ is $d \times d$ orthonormal matrices, i.e. $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_n$ and $\boldsymbol{V}^\top \boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_d$. The diagonal entries of $\boldsymbol{\Sigma}'$ are known as the singular values, the columns of $\boldsymbol{U}$ are left singular vectors, and the columns of $\boldsymbol{V}$ are right singular vectors of $\boldsymbol{X'}$. The columns of $\boldsymbol{V}$ yield an orthonormal basis of the data observations (in $\mathbb{R}^d$) while the columns of $\boldsymbol{U}$ yield an orthonormal basis of the data variables (in $\mathbb{R}^n$) of the matrix $\boldsymbol{X'}$. Regarding the eigen-decomposition, $\boldsymbol{V}$ is the set of the

eigenvectors of $\boldsymbol{X'}^\top \boldsymbol{X'}$, the columns of $\boldsymbol{U}$ are the eigenvectors of $\boldsymbol{X'}\boldsymbol{X'}^\top$, and the singular values are the square root of the eigenvalues of both of matrices, $\boldsymbol{X'}^\top \boldsymbol{X'}$ and $\boldsymbol{X'}\boldsymbol{X'}^\top$ which have the same eigenvalues.

The matrix $\boldsymbol{C_X}$ can be rewritten as

$$\boldsymbol{C_X} = \left(\boldsymbol{U\Sigma' V}^\top\right)^\top \boldsymbol{U\Sigma' V}^\top = \boldsymbol{V\Sigma'}^\top \boldsymbol{U}^\top \boldsymbol{U\Sigma' V}^\top = \boldsymbol{V\Sigma'}^\top \boldsymbol{\Sigma' V}^\top$$

Comparing to the eigen-decomposition of $\boldsymbol{C_X}$ in (2), the right singular vectors $\boldsymbol{V}$ of $\boldsymbol{X'}$ are equivalent to the eigenvectors $\boldsymbol{Q}$ (also see in (2)) of $\boldsymbol{C_X}$ and the singular values $\boldsymbol{\Sigma'} = \boldsymbol{\Lambda}^{1/2}$. Therefore, when the singular values in $\boldsymbol{\Sigma'}$ are decreasingly ordered the new representation $\boldsymbol{Y}$ of PCA can be computed by

$$\boldsymbol{Y} = \boldsymbol{XV}\boldsymbol{I}_{d\times m}, \tag{4}$$

where the first m singular vectors of $\boldsymbol{X'}$ are the principal components.

Note that the columns of $\boldsymbol{V}$ are also the right singular vectors of $\boldsymbol{X}$. In particular, it can be written by

$$\boldsymbol{X} = \boldsymbol{U}\left(\sqrt{n-1}\boldsymbol{\Sigma'}\right)\boldsymbol{V}^\top.$$

Moreover, the factor $\sqrt{n-1}$ does not affect the order of singular values. Since the goal of PCA is to identify the principal components or the orthonormal vectors forming the basis of the new representation $\boldsymbol{Y}$, the matrix $\boldsymbol{V}$ in (4) can be thus computed from the dataset $\boldsymbol{X}$ instead of $\boldsymbol{X'}$. Using the properties of SVD of $\boldsymbol{X}$, we have

$$\boldsymbol{XV} = \boldsymbol{U\Sigma}, \tag{5}$$

where $\boldsymbol{\Sigma} = \sqrt{n-1}\boldsymbol{\Sigma'}$. This equivalence allows computing $\boldsymbol{Y}$ by using the left singular vectors of $\boldsymbol{X}$ instead of the rights and the singular values without the appearance of $\boldsymbol{X}$. Additionally, the singular values and left singular vectors are equivalent to the eigenvalues and eigenvectors of the Gram matrix $\boldsymbol{X}\boldsymbol{X}^\top$. That relates to the multidimensional scaling method which will be discussed in Section 2.2.

2.1.3 Advantages, disadvantages and limitations

Advantages: The reason why PCA is used widely comes from the following advantages:

- It is simple: easy to implement and understand based on linear algebra.
- Less sensitive to the outliers: the principal components are determined based on the covariance matrix of the whole dataset. The outliers usually have a small size do not have a significant influence on the final answer.
- Non-parametric and unique principal component: there is no need to choose any model parameter, and the principal component vector is unique [8]. Therefore, the users will get the same answer if they use the same dataset with the same units [16].

Disadvantages: Although there is no model parameter in PCA, data may have to be standardized during reprocessing, and we need to decide m the number of principal components to be used.

- Data standardization [8]: data variables can be expressed in different units such as kilograms, meters, years, and they can have very different variance. We need to standardize these variables by their standard deviation before PCA; otherwise, the principal components will be biased towards the high variance variables. However, standardization is not always necessary. For example, when a variable has noise with small variance, it becomes more important after standardization. That makes choosing the principal components more difficult and may lead to an incorrect answer. Therefore, standardizing data variables should be decided carefully based on the prior knowledge of the data.

- Difficulty in choosing m - intrinsic dimension: as discussed above, all d eigenvalues are usually non-zero in real situations. There is not a well-accepted objective way to identify m principal components [8]. The choice depends on the users' purpose and dataset. If we would like only to visualize, then two or three principal components are a good choice. If we would like to use the result of PCA for further analysis, there are some criteria to determine m, as discussed in [11].

Limitations: PCA is solved under the assumptions (mentioned at the end of Section 2.1.1) which do not always hold in real situations. PCA fails when the dataset has some of the following properties:

- Nonlinear transformation: when the data are nonlinearly embedded into a high-dimensional space, PCA cannot detect the intrinsic dimension, as shown in the example in Figure 3 and 4.
- Non-Gaussian distribution: PCA works on the covariance matrix which does not well describe the dataset if its points are non-Gaussian distributed. We can get a biased answer. For example, the points uniformly distributed in a rectangle are embedded linearly into three-dimensional space [11]. PCA cannot discover the rectangle; the first principal component captures the diagonal of the rectangle instead of the length.
- Non-orthogonal transformation: when the transformation is not orthogonal, PCA finds out the intrinsic data with some errors [11, 8]. For example, when data points are distributed as +-shape, it turns into an X-shape after a non-orthogonal transformation. PCA cannot discover the +-shape; the second principal component does not capture the second dimension of +-shape.

Finally, the Cartesian coordinates of data points are required to solve PCA. If the data are not numeric (e.g., categorical data), it must be mapped into numerical space before applying PCA.

2.2 Multidimensional Scaling (MDS)

MDS is a family of methods that preserve the distance. We can classify this family into: (1) classical MDS, (2) metric MDS, and (3) non-metric MDS [12, 11]. However, in this report, we only discuss the classical MDS with the assumption that the data are given in the Euclidean space where we can get the exact solution. The classical MDS, in general, works in any space.

The classical MDS is also known as another approach of PCA which preserves scalar product. A less familiar name of MDS is Principle Coordinates Analysis (PCoA). If the data points are given in the Cartesian coordinates, MDS and PCA provide the same result. However, unlike PCA, which requires the knowledge of the coordinates of the data points, MDS starts with the dissimilarities that are usually defined by a distance matrix. MDS's goal is to represent the structure of distance data from a high-dimensional space to a lower-dimensional space preserving the distance.

2.2.1 Embedding of the dataset

In reality, the data can be given as the distances between points instead of the coordinates system. Assuming that we have data distance information, the goal of MDS is to find data coordinates in as few dimensions as possible with the same distance.

Like PCA, the purpose of the classical MDS is to find out a representation $\boldsymbol{Y}_{n\times m}$ of the original data $\boldsymbol{X}_{n\times d}$ by a linear transformation $\boldsymbol{W}_{d\times m}$ as in (1) with the assumption that the columns of $\boldsymbol{W}$ are orthonormal, i.e. $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_m$. This assumption is to make the problem solvable using linear algebra. We have $\boldsymbol{W}\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{W}$, $\boldsymbol{W}^\top$ is a generalized inverse of $\boldsymbol{W}$ [2]. Therefore, the transformation in (1) can be rewritten as

$$\boldsymbol{X} = \boldsymbol{Y}\boldsymbol{W}^\top.$$

Note that it is not necessary to know $\boldsymbol{X}$ in MDS. Instead, the pairwise scalar products or pairwise distances are the input of MDS. Both of them are real symmetric matrices that can be diagonalized

by an orthogonal matrix. Another assumption of MDS is that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are centered for the reasons: (i) the solution is the same as the one using PCA, and (ii) the so-called double centering operation can be used to compute scalar products from the distances. Both of these reasons will be precise in the next section.

Note that, in a Euclidean space, rotation, reflection, or a combination of them preserve the scalar product while the translation does not. Therefore, by the centralization assumption, scalar products are not preserved after MDS in the original space. In other words, if $\boldsymbol{X}'$ is the original data and $\boldsymbol{X}$ is its centered version, the scalar product between points in the new representation $\boldsymbol{Y}$ corresponding to the centered $\boldsymbol{X}$ is different from the new representation $\boldsymbol{Y}'$ obtained from the original space $\boldsymbol{X}'$.

2.2.2 Gram matrix

As discussed in Section 2.1.2, when the coordinate information $\boldsymbol{X}$ is missing, we can get the new representation $\boldsymbol{Y}$ by using the eigenvalues and eigenvectors of the matrix $\boldsymbol{X}\boldsymbol{X}^\top$. In Euclidean space, this matrix is the Gram matrix. In general, the Gram matrix of the dataset $\boldsymbol{X}$ is an $n \times n$ symmetric matrix of the scalar products of all pair vectors in $\boldsymbol{X}$. The scalar product between two vectors $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by

$$s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = \boldsymbol{x}_i^\top \boldsymbol{x}_j, \tag{6}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$ are the column vectors. The Gram matrix of $\boldsymbol{X}$ can be thus written as

$$\boldsymbol{S}_{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{X}^\top.$$

It can be seen that the Gram matrix of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the same,

$$\boldsymbol{S}_{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{Y}\boldsymbol{W}^\top\boldsymbol{W}\boldsymbol{Y}^\top = \boldsymbol{Y}\boldsymbol{Y}^\top = \boldsymbol{S}_{\boldsymbol{Y}}. \tag{7}$$

The classical MDS thus preserves the scalar product after transforming.

Next, we will see how $\boldsymbol{Y}$ can be computed from this Gram matrix. By the symmetry, there exists the eigen-decomposition

$$\boldsymbol{S}_{\boldsymbol{Y}} = \boldsymbol{S}_{\boldsymbol{X}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top = \boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\left(\boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\right)^\top, \tag{8}$$

where $\boldsymbol{Q}$ is an $n \times n$ orthonormal matrix, i.e. $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}_n$, and $\boldsymbol{\Lambda}_{n\times n}$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{S}_{\boldsymbol{Y}}$. If the eigenvalues are arranged in decreasing order, the new representation $\boldsymbol{Y}$ can be computed by

$$\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{\Lambda}^{1/2}\boldsymbol{I}_{n\times m}. \tag{9}$$

Note that the Gram matrix is positive semi-definite; therefore, all its eigenvalues are non-negative. Besides that, the rank of this matrix is at most d, and it thus has at most d strictly positive eigenvalues while the rest ones are zero. The d associating eigenvectors form a new orthogonal basic of $\boldsymbol{Y}$, and the eigenvalues are proportional to the variance in Y along the corresponding axis [12]. Like PCA, choosing $m < d$ dominant eigenvalues is done under the assumption that the large variance directions contain the important information of the data while the small variance ones represent the noise.

2.2.3 Distance matrix

The Gram matrix is not usually known in reality, but instead, a pairwise distance matrix is generally available. Consider data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, the Euclidean square distance matrix $\boldsymbol{D} = \{d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{1\le i,j\le n}$ can be defined by using the scalar product as follows

$$\begin{aligned} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle = \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle - 2\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle \\ &= s(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2s(\boldsymbol{x}_i, \boldsymbol{x}_j) + s(\boldsymbol{x}_j, \boldsymbol{x}_j), \end{aligned} \tag{10}$$

where $\boldsymbol{x}$ is assumed to be centered, i.e. $\frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0}_d$, where $\boldsymbol{0}_d$ is vector of d zero entries. Note that we use the square distance d^2 instead of d in the matrix $\boldsymbol{D}$ for simplicity of notation in later computations. The input of MSD is usually the Euclidean distance matrix.

Using the centralization of $\boldsymbol{X}$ which is $\sum_{i=1}^{n} s(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{j=1}^{n} s(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ and Equation (10) we have

$$\sum_{i=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{i=1}^{n} s(\boldsymbol{x}_i, \boldsymbol{x}_i) + ns(\boldsymbol{x}_j, \boldsymbol{x}_j),$$
$$\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = ns(\boldsymbol{x}_i, \boldsymbol{x}_i) + \sum_{j=1}^{n} s(\boldsymbol{x}_j, \boldsymbol{x}_j),$$
$$\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = n\sum_{i=1}^{n} s(\boldsymbol{x}_i, \boldsymbol{x}_i) + n\sum_{j=1}^{n} s(\boldsymbol{x}_j, \boldsymbol{x}_j) = 2n\sum_{l=1}^{n} s(\boldsymbol{x}_l, \boldsymbol{x}_l).$$

By substituting these equations into the relationship between the pairwise distance and the scalar product in (10), the scalar product can be computed from the Euclidean distance as follows

$$\begin{aligned} s(\boldsymbol{x}_i, \boldsymbol{x}_j) &= -\frac{1}{2}\left(d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - s(\boldsymbol{x}_i, \boldsymbol{x}_i) - s(\boldsymbol{x}_j, \boldsymbol{x}_j)\right) \\ &= -\frac{1}{2}\left[d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{n}\left(\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)\right. \\ &\qquad \left. -\frac{1}{n}\left(\sum_{i=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\right)\right] \\ &= -\frac{1}{2}\left(d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{n}\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{1}{n}\sum_{i=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{x}_j)\right). \end{aligned}$$

In the matrix form, the Gram matrix, $\boldsymbol{S} = \boldsymbol{S_X} = \boldsymbol{S_Y}$, can be written as

$$\begin{aligned} \boldsymbol{S} &= -\frac{1}{2}\left(\boldsymbol{D} - \frac{1}{n}\boldsymbol{D}\mathbf{1}_n\mathbf{1}_n^\top - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{D} + \frac{1}{n^2}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{D}\mathbf{1}_n\mathbf{1}_n^\top\right) \\ &= -\frac{1}{2}\boldsymbol{J}\boldsymbol{D}\boldsymbol{J} \end{aligned} \tag{11}$$

where $\boldsymbol{J} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix, $\mathbf{1}_n$ is a column vector of length n with all entries equalling one, and $\boldsymbol{I}_n$ is the identity matrix of size $n \times n$. Equation (11) is called "double centering" operation. It is composed of subtracting all entries of the distance matrix $\boldsymbol{D}$ by the mean of the corresponding row, the mean of corresponding column, and adding the mean of all entries. The eigen-decomposition of the Gram matrix $\boldsymbol{S}$ computed from (11) gives the new representation $\boldsymbol{Y}$ as in (9).

The relationship between the distance and the scalar product in Equation (10) implies that MDS preserves not only the scalar product but also the Euclidean distance if $\boldsymbol{X}$ is in Euclidean space. We mentioned that if the original data are not centered at the beginning, the Gram matrix of new representation $\boldsymbol{Y}$ will not be the same as the Gram matrix of the original space since centering. However, the centering does not change the pairwise distance. Therefore, MDS actually preserves the Euclidean distance.

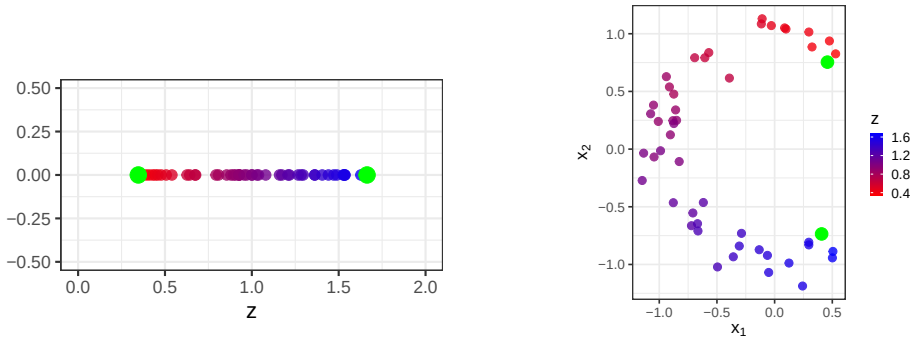
2.2.4 The equivalence of MDS and PCA

The classical MDS and the PCA give the same result if the Cartesian coordinates of the dataset $\boldsymbol{X}$ are known. In this case, the computation of $\boldsymbol{Y}$ in (4) and (9) are equal. Let us consider the SVD of $\boldsymbol{X}$ as $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma}$ is an $n \times d$ diagonal matrix containing singular values, $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_n$ and $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{I}_d$, we have

$$\boldsymbol{S_Y} = \boldsymbol{S_X} = \boldsymbol{X}\boldsymbol{X}^\top = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}'^\top\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}'^\top\boldsymbol{U}^\top.$$

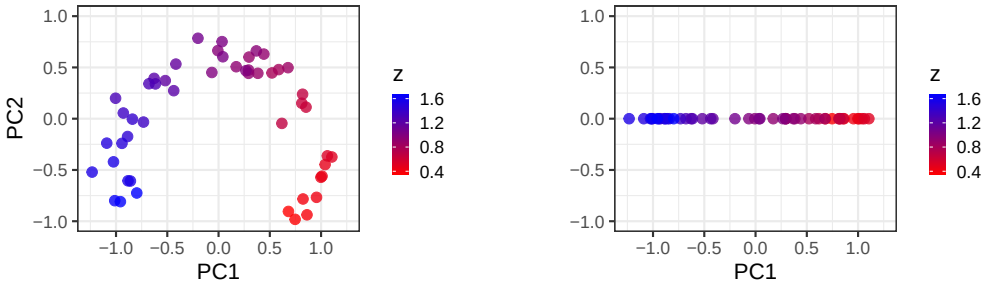
By identifying to (8), we can set $\boldsymbol{Q} = \boldsymbol{U}$ and $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}'\boldsymbol{\Sigma}'^\top$. The new representation $\boldsymbol{Y}$ in (9) becomes

$$\boldsymbol{Y} = \boldsymbol{U}\left(\boldsymbol{\Sigma}'\boldsymbol{\Sigma}'^\top\right)^{1/2}\boldsymbol{I}_{n\times m} = \boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{I}_{d\times m}. \tag{12}$$



(a) One-dimensional hidden space (b) Two-dimensional embedding space

Figure 3: C-curve data (of size 50): (a) one-dimensional hidden space, $z \sim U(\pi/3, 5\pi/3)$, and (b) two-dimensional embedding space, $\boldsymbol{x} = (\cos(z) + \varepsilon_1, \sin(z) + \varepsilon_2)$ where $\epsilon_i \sim N(0, 0.1^2)$. Two green points have the largest separation in z. The point color is encoded by z.



(a) (b)

Figure 4: PCA: (a) two and (b) one principal components of C-curve data (in Fig. 3b). The color is encoded by the value of z shown in Figure 3a.

By using (5), we have the same result of $\boldsymbol{Y}$ using PCA in (4) and using MDS in (12). PCA uses the right singular vectors of $\boldsymbol{X}$ while MDS uses the left ones. Under the assumption that the data coordinates $\boldsymbol{X}$ is unknown, the use of $\boldsymbol{V}$ of MDS is reasonable. With this left eigenvectors, we do not need $\boldsymbol{X}$ to compute the new representation $\boldsymbol{Y}$.

2.2.5 Advantages, disadvantages and limitations

The classical MDS has all the advantages, disadvantages, and limitations of PCA (discussed in Section 2.1.3). However, compared to PCA, MDS is more flexible since the input can be the coordinates, the Gram matrix, or the distance matrix, while PCA requires the coordinates. For this reason, MDS does not meet the difficulty in the standardization of the variables in PCA. Nevertheless, MDS can be more expensive in the computation if the number of observations of the data is larger than its dimension.

Both PCA and classical MDS are the linear methods of dimensionality reduction. They fail when the data are a nonlinear combination of its basis, as shown in the example of C-curve in Figure 3. Both of the two principal components (in Fig. 4a) do not provide a correct order in color as (1D) hidden space shown in Figure 3a. Figure 4b shows the superposed turns at the top and bottom of C-curve in the first PC. Therefore, PCA fails in discovering the (1D) hidden space of the data, and MDS fails too. A nonlinear method will be discussed in the next section.

3 Diffusion Map (DM)

One reason for the failure of PCA and MDS on a nonlinearly embedded data is due to the Euclidean distance they preserve. In a nonlinear data structure, the Euclidean distance gives very little information about the dissimilarity. For example, two green points in Figure 3b have the largest distance in (1D) hidden space (in Fig. 3a). However, their Euclidean distance is not the largest one. Diffusion maps method, in a different approach, works with diffusion distance on a weighted graph. The diffusion distance better describes the dissimilarity between points; that is, the points are far apart if they are poorly connected. Besides, the diffusion map embeds the data into a Euclidean space where the Euclidean distance corresponds to the diffusion distance defined in terms of a random walk on the weighted graph constructed from the data. We will now turn to the discussion on how to construct a weighted graph from data.

3.1 Local similarity and connectivity

The method starts with the idea that only highly correlated data points are meaningful. This correlation will be characterized by local similarity. Considering the dataset $\boldsymbol{X}$ ($\boldsymbol{X} \subset \mathbb{R}^d$ or a vector space) as a weighted undirected graph (where each data point is a node), we construct a Markov chain on this graph. The *connectivity* of two nodes is defined by the probability of jumping between these nodes. In this sense, a nonlinear kernel $k(.,.)$, which measures the similarity between nodes, is useful to describe this connectivity.

Kernel function: On the dataset $\boldsymbol{X}$, we construct a weighted graph in which each data point is a node and the weights are represented by a modification of Gaussian kernel function given by

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right) & \text{if } \boldsymbol{x}_i \neq \boldsymbol{x}_j \\ 0 & \text{if } \boldsymbol{x}_i = \boldsymbol{x}_j, \end{cases} \tag{13}$$

for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$, where σ is called *kernel width*. This function is the Gaussian kernel, not allowing self transition within the node. The reason for this choice will be discussed at the end of this section after introducing the transition matrix of a Markov chain.

The Gaussian kernel function mentioned above is only one example of the general functions $k(.,.)$ satisfying two following properties:

- symmetric, i.e. $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_j, \boldsymbol{x}_i)$,
- non-negative, i.e $k(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$ for all $\boldsymbol{x}_i, \boldsymbol{x}_j$.

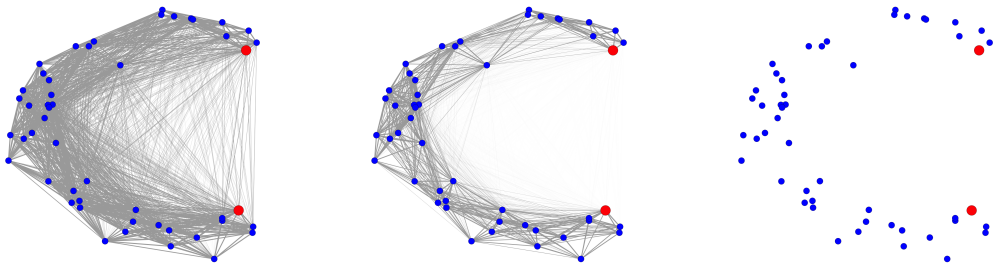
However, later in this report, we will only consider the Gaussian kernel with the kernel width σ in (13). This kernel function can be later interpreted as a scaled transition probability between nodes that must be non-negative. The second non-negative property comes from this reason. The first symmetric property is necessary to perform the eigen-decomposition of the transition matrix (see Appendix A).

The kernel function measures the *local similarity* of a node. Given node $\boldsymbol{x}_i$, the function in (13) has significant non-zero value within a neighborhood, and quickly goes to zero outside this area. A node $\boldsymbol{x}_j$ is similar to $\boldsymbol{x}_i$ if it lies inside the neighborhood of $\boldsymbol{x}_i$.

The function in (13) is the non-normalized form of the isotropic Gaussian density with mean $\boldsymbol{x}_i$ and variance σ^2. The missing constant is not necessary; it will be canceled out in the definition of the connectivity, that we will see later. Therefore, the kernel width has the same meaning as the standard deviation in the Gaussian distribution, and characterizes the neighborhood of $\boldsymbol{x}_i$.

Kernel width: In (13), σ defines the spatial extent within which the nodes are similar. As an illustrated example of the effect of σ, Figure 5 displays graphs of C-curve data with the edges weighted by the Gaussian kernel.

- When σ is large, in Figure 5a, this spatial extent is large. It allows us to jump between nodes apart directly, e.g., there is a direct link between two red nodes in the figure. In this case, we cannot see the geometrical structure of the data.



(a) $\sigma = 1$ (b) $\sigma = 0.5$ (c) $\sigma = 0.1$

Figure 5: The graphs are constructed on C-curve data (in Fig. 3). Two red nodes have the largest separation of z shown in Figure 3a. The edges of the graph are weighted by the Gaussian kernel in (13) with various σ.

- When σ is small, in Figure 5b, it is only possible to jump between close nodes. There is no direct link to go further nodes. We can see the curvature of C-shape by connecting together short links residing on the nonlinear geometrical structure of the data points.
- When σ is too small, as in Figure 5c, so that there is no link between any two nodes, the graph is disconnected. It does not capture any global geometrical relationship.

With different values of σ, the local structure of the data is described in different ways. This kernel width thus depends on the prior knowledge of geometry and the density of the data. If the data points are uniformly distributed, σ can be chosen as a constant such that

- It is not too small to ensure the graph is connected;
- It is small enough to describe the geometry of data such as the curvatures and the disconnection.

A simple way to choose a suitable width is to use the k-nearest neighbor ($k$nn) distance. In the computational experiments section, we will choose σ as the median of the $k$nn distances with a default value of 1-2% of the sample size for k in the `diffusionMap`[2] package of R.

Other criteria to choose the constant σ were discussed in [3, 7]. When the distribution of data points is non-uniform or anisotropic, more than one σ should be used to get a good description of the data structure. In [15], they suggested choosing different σ for different data points. However, we will not discuss these choices in detail, they are out of the scope of this report.

Transition matrix: On the weighted graph constructed from the dataset $\boldsymbol{X}$, we construct a Markov chain. We then introduce a diagonal matrix $\boldsymbol{D}$ with the entries,

$$D_i = \sum_{k=1}^{n} W_{ik} \tag{14}$$

that measures the degree of the node $\boldsymbol{x}_i$, where $W_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the weight. Note that D_i is proportional to the number of links to node $\boldsymbol{x}_i$. Thus, it relates to the point density: a low-density point corresponds to a node with a small degree and vice versa. Consider a row normalization matrix of $\boldsymbol{W}$,

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}. \tag{15}$$

We can see that $\sum_{j=1}^{n} P_{ij} = 1$, and $P_{ij} \geq 0$. The matrix $\boldsymbol{P}$ can be thus viewed as the Markov chain's transition matrix, where each entry,

$$P_{ij} = p(\boldsymbol{x}_j \mid \boldsymbol{x}_i) = \frac{W_{ij}}{D_i} = \frac{k(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{k=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_k)}, \tag{16}$$

[2] `https://cran.r-project.org/web/packages/diffusionMap/diffusionMap.pdf`

defines the *connectivity* of two nodes $\boldsymbol{x}_i, \boldsymbol{x}_j$

$$connectivity(\boldsymbol{x}_i, \boldsymbol{x}_j) = p(\boldsymbol{x}_j \mid \boldsymbol{x}_i) \propto k(\boldsymbol{x}_i, \boldsymbol{x}_j).$$

The term $p(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$ is the transition probability of moving to node $\boldsymbol{x}_j$ given that the chain starts at $\boldsymbol{x}_i$ in a single step. Although $\boldsymbol{W}$ inherits the symmetry of the kernel function, the matrix $\boldsymbol{P}$ is no longer symmetric. Note that the indices for an element of $\boldsymbol{P}$ are swapped when writing the corresponding conditional probability. In particular, P_{ij} is the probability of $\boldsymbol{x}_j$ given $\boldsymbol{x}_i$; therefore, $\sum_j P_{ij} = 1$. And if $\boldsymbol{p}_0$ is a column vector of the initial distribution of the chain, it has to multiply by the left of $\boldsymbol{P}$ to get the chain's distribution in the next step, i.e., $\boldsymbol{p}_1^\top = \boldsymbol{p}_0^\top \boldsymbol{P}$.

We reconsider the kernel function in (13), the case of $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = 0$ is equivalent to the self-transition probability $P_{ii} = 0$, which leads to the following conclusions:

- Due to excluding the self-transition, the Markov chain cannot stay a long time at low-density nodes. If the Gaussian kernel includes the self-transition, that is $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = 1$, the self-transition probability P_{ii} at a low-density node $\boldsymbol{x}_i$ is dominant among the transition probabilities $\{P_{ij}\}_{j=1,\dots,n}$. The Markov chain will thus stay in $\boldsymbol{x}_i$ a long time before jumping to another node. By setting $P_{ii} = 0$, the Markov chain will continuously move to another node.

- The probability of jumping to other nodes becomes large because the row summation D_i in (14) becomes small. The graph is thus connected with a small kernel width σ. Furthermore, when the data are non-uniform, the choice of a constant σ is more difficult. The kernel function with a large σ cannot describe the underlying data structure (such as curvature and disconnection), while a small σ causes loss of connection with low-density nodes. Excluding the self-transition at each node on the graph helps the low-density nodes keep connected with a small σ.

- If $\boldsymbol{x}_i$ has a higher density than $\boldsymbol{x}_j$ then the ratio $\frac{D_j}{D_i} < 1$. This ratio becomes smaller because of excluding the self-transition, which leads to a small ratio $\frac{p(\boldsymbol{x}_j|\boldsymbol{x}_i)}{p(\boldsymbol{x}_i|\boldsymbol{x}_j)}$. It means that this kernel function makes a wider gap between $p(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$ and $p(\boldsymbol{x}_i \mid \boldsymbol{x}_j)$. Therefore, it emphasizes the difference in density between data points.

3.2 Timescale t and relationship of local and global structure

As previously mentioned, P_{ij} defines the connectivity between nodes on the weighted graph (where each node is a data point). However, when the data have a nonlinear structure, given a small enough kernel width σ, this connectivity is not enough to describe the connection between the long-distance nodes (along the underlying data structure). For example, in figure 5, the edges' thickness of C-curve graph represents the scaled values of P_{ij}, a thicker edge shows a large value. When $\sigma = 0.5$ is small, there is no direct link between two red points that have the longest distance in the one-dimensional parametrization z used to generate the data (shown in Figure 3a). A feasible connectivity path between them is indirect and along the C-curve. Such path consists of multiple short and direct links between nodes presented by the thickness, as shown in Figure 5b. By introducing *timescale* t, the matrix $\boldsymbol{P}^t$ describes the connectivity between the long-distance nodes along the data structure.

As a normalization of the kernel function, which measures the local similarity between points, P_{ij} contains the information of the local structure of the data. Besides, P_{ij} is the transition probability of moving from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ in one step. Using the properties of a time-homogeneous Markov chain, the probability of going to $\boldsymbol{x}_j$ from $\boldsymbol{x}_i$ in t steps is represented by P_{ij}^t. We denote $p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_i) = P_{ij}^t$. For a large enough $t > 1$, when moving the chain forward, P_{ij}^t is the summation of all paths of length t between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The probability of each path of length t is the product of the local connectivities,

$$p(\boldsymbol{x}_j \mid \boldsymbol{x}_h)p(\boldsymbol{x}_h \mid \boldsymbol{x}_k) \cdots p(\boldsymbol{x}_l \mid \boldsymbol{x}_i),$$

where $\boldsymbol{x}_h, \boldsymbol{x}_k, \boldsymbol{x}_l$ are arbitrary nodes on the graph. These paths have high probability if the nodes are highly connected via other nodes. Additionally, the probability of jumping to a high-density

node $\boldsymbol{x}_j$ is larger than that of a low-density node because $\boldsymbol{x}_j$ has more connecting links. Therefore, the paths along the underlying data structure via highly connected nodes have a higher probability to be reached than the paths via low-density nodes.

In this way, with a large enough t, the matrix $\boldsymbol{P}^t$ integrates the local connectivities to provide the global measure for the connectivity of the data. It thus reveals the global data structure. Nevertheless, by the Markov chain properties, when t is too large, the transition probability tends to a stationary distribution.

Stationary distribution: We will show that the transition probability tends to be a unique stationary distribution as t goes to infinity. This stationary distribution is independent of the initial state of the Markov chain.

Assuming that a kernel width σ is chosen so that the graph is fully connected, it implies that the chain is irreducible. And when the timescale $t > 1$, the probability of returning to itself after t steps is non-zero, i.e. $p_t(\boldsymbol{x}_i \mid \boldsymbol{x}_i) > 0$. The chain is thus aperiodic. Using the properties of a time-homogeneous Markov chain, the transition probability $\boldsymbol{P}^t$ tends to be a unique stationary distribution over the nodes $\boldsymbol{\pi}$ when t goes to infinity, that is

$$\lim_{t\to\infty} p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_i) = \pi(\boldsymbol{x}_j),$$

where $\boldsymbol{\pi}$ is a column vector of length n. We can verify that this stationary distribution has the form

$$\pi(\boldsymbol{x}_j) = \frac{D_i}{\sum_{k=1}^n D_k}, \tag{17}$$

since $\boldsymbol{\pi}^\top \boldsymbol{P} = \boldsymbol{\pi}^\top$,

$$\sum_{i=1}^n \pi(\boldsymbol{x}_i) P_{ij} = \sum_{i=1}^n \frac{D_i}{\sum_{k=1}^n D_k} \frac{W_{ij}}{D_i} = \frac{\sum_{i=1}^n W_{ij}}{\sum_{k=1}^n D_k} = \pi(\boldsymbol{x}_j),$$

using the definition of P_{ij} in (16).

Additionally, the stationary probabilities satisfy the detailed balance condition

$$\pi(\boldsymbol{x}_i) p(\boldsymbol{x}_j \mid \boldsymbol{x}_i) = \frac{D_i}{\sum_{k=1}^n D_k} \frac{W_{ij}}{D_i} = \frac{D_j}{\sum_{k=1}^n D_k} \frac{W_{ij}}{D_j} = \pi(\boldsymbol{x}_j) p(\boldsymbol{x}_i \mid \boldsymbol{x}_j)$$

since $\boldsymbol{W}$ is symmetric. Using this condition, we will show that the stationary distribution depends on the local density of nodes. If $\boldsymbol{x}_i$ is high density, it has large D_i (e.g. middle nodes in C-curve shown in Figure 5), and if $\boldsymbol{x}_j$ is low density, D_j is small (such as two red nodes in Figure 5). From (16), we have $p(\boldsymbol{x}_j \mid \boldsymbol{x}_i) < p(\boldsymbol{x}_i \mid \boldsymbol{x}_j)$ since $\boldsymbol{W}$ is symmetric, leading to $\pi(\boldsymbol{x}_i) > \pi(\boldsymbol{x}_j)$ using the balance condition. Consequently, the value of $\pi(\boldsymbol{x})$ depends on the local density and location of $\boldsymbol{x}$: if $\boldsymbol{x}$ is high density and easy to get in from other nodes, $\pi(\boldsymbol{x})$ is large; while this value is small if $\boldsymbol{x}$ has low-density and poor connection to other nodes.

This observation clarifies the last comment in the previous section about the choice of the Gaussian kernel function excluding self-transition: the ratio of the transition probabilities $\frac{p(\boldsymbol{x}_j|\boldsymbol{x}_i)}{p(\boldsymbol{x}_i|\boldsymbol{x}_j)}$ is smaller than that on the graph weighted by the Gaussian kernel allowing self-transition, where $\boldsymbol{x}_i$ is higher density than $\boldsymbol{x}_j$. The ratio $\frac{\pi(\boldsymbol{x}_j)}{\pi(\boldsymbol{x}_i)}$ becomes thus smaller by using the balance condition. As a consequence, the Markov chain's stationary probabilities, excluding self-transition, emphasize the point density.

Timescale: We will discuss the effect of timescale t on the connectivity of the graph and a criterion of choosing t.

Let us look at the example of $\boldsymbol{P}^t$ constructed on the C-curve set with a different value of t shown in Figure 6. Note that the order of nodes follows the one-dimensional parametrization z_i shown in Figure 3a, where $z_1 < z_2 < \ldots < z_{50}$. In the first row of the figure, the color of a node $\boldsymbol{x}_j$ is encoded by the transition probability of moving from the big top node $\boldsymbol{x}_1$ to $\boldsymbol{x}_j$ in t steps, i.e., $p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_1)$.

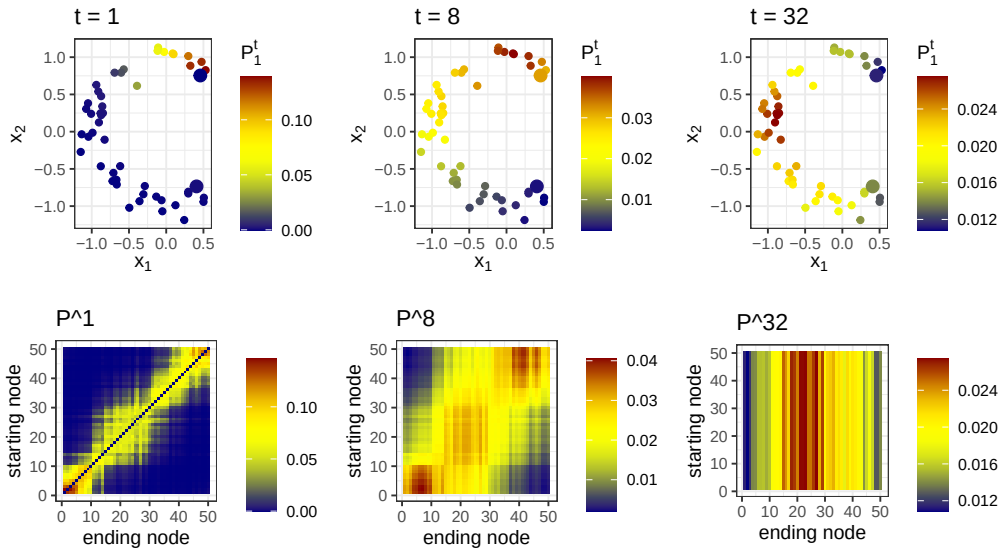


Figure 6: Diffusion at time $t = 1, 8, 32$ on the C-curve dataset, as shown in Figure 3. The first row shows the data points with the color encoded by the first row of the matrix P^t where the row corresponds to the big top point. Two big points have the longest distance in one-dimensional parametrization z used to generate the data (Fig. 3a). The second row displays the value of matrix P^t where the order of the rows is the order of z, the first row (the bottom of the image) corresponds to the smallest value of z, and the last row corresponds to the largest z. The first and last rows correspond to two big points in the C-curve data in the first row of the figure.

- When $t = 1$, the probabilities of jumping from the big top node to its neighbors are significantly higher than zero, it is impossible to go out of the neighborhood. The transition probability describes the connectivity between nodes locally.
- When $t = 8$, the probabilities are different along the C-shape. The probability of getting out of the neighborhood of $\boldsymbol{x}_1$ is larger. $\boldsymbol{P}^t$ represents the global connectivity of data points and reveals the underlying structure.
- When $t = 32$, $\boldsymbol{P}^t$ does not describe the connectivity any more. The probabilities of moving to some middle nodes are large, while they are quite similar between nodes near the top and the bottom of C-curve. These probabilities are the stationary distribution of the Markov chain on C-curve data.

The images in the second row of Figure 6 represent the matrix $\boldsymbol{P}^t$. Note that the first row of $\boldsymbol{P}^t$ is displayed at the bottom of each image, while the last row is shown at the top.

- Matrix $\boldsymbol{P}$ is close to symmetric (but it is not symmetric). The reason is that the connectivity is described locally within the spatial extent where the nodes are similar. When $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar, the probabilities $p(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$ and $p(\boldsymbol{x}_i \mid \boldsymbol{x}_j)$ are approximately the same. When they are not similar, their transition probabilities are almost zero.
- $\boldsymbol{P}^8$ is asymmetric. It represents the global connectivity. Given a node $\boldsymbol{x}_i$, the probability $p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$ (a row of the matrix $\boldsymbol{P}^t$) is non-zero for a non-neighbor $\boldsymbol{x}_j$ of $\boldsymbol{x}_i$. Besides, $p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_i)$ varies depending on how long to go from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ along C-shape.
- $\boldsymbol{P}^{32}$ has constant columns which approximate the stationary distribution. Given an arbitrary initial distribution $\boldsymbol{p}_0$, $\boldsymbol{P}^{32}$ maps $\boldsymbol{p}_0$ into the stationary distribution, $\boldsymbol{p}_0^\top \boldsymbol{P}^{32} = \boldsymbol{\pi}^\top$.

Timescale t plays an essential role in discovering the underlying data structure. Given a small enough kernel width σ, the matrix $\boldsymbol{P}^t$ reveals the underlying structure of the data with a suitable t:

- t should not be too small. When t is too small, $\boldsymbol{P}^t$ cannot describe the connectivity of the long-distance points. Therefore, it cannot provide the global structure of the data.
- t should not be too large. If t is too large, $\boldsymbol{P}^t$ maps all distributions to the stationary distribution. It does not describe the connectivity between nodes and does not provide the data structure.

Besides, we will see later that t is a scale parameter. With various t, the global structure of the data is represented in different scales. Also, t affects the outcomes of dimensionality reduction and the accuracy of the method. Therefore, we will discuss the criterion for choosing t later in Section 3.5.

3.3 Eigen-decomposition of the transition matrix

The matrix $\boldsymbol{P}^t$ contains interesting information about the global structure, which helps discover the underlying structure of the data. However, evaluating $\boldsymbol{P}^t$ of size $n \times n$ is computationally expensive when t gets large. Eigen-decomposition is a classical technique for this problem. Moreover, using eigenvalues and eigenvectors helps to define diffusion coordinates and diffusion map as well as obtain the relationship between diffusion distance and the Euclidean distance.

The matrix $\boldsymbol{P}$ defined in (15) has non-negative entries but is not symmetric, although $\boldsymbol{W}$ is symmetric. Hence, the existence of the eigen-decomposition of $\boldsymbol{P}$ is not guaranteed. However, matrix

$$\boldsymbol{P'} = \boldsymbol{D}^{1/2}\boldsymbol{P}\boldsymbol{D}^{-1/2}$$

is symmetric and has the same eigenvalues as $\boldsymbol{P}$ (shown in Appendix A). From the symmetry of $\boldsymbol{P'}$, there exists a set of eigenvalues $\{\lambda_l\}_{l=0,\dots,n-1}$ and an orthonormal set of column eigenvectors $\{\boldsymbol{\phi}_l\}_{l=0,\dots,n-1}$ such that

$$\boldsymbol{P'} = \sum_{l=0}^{n-1} \lambda_l \boldsymbol{\phi}_l \boldsymbol{\phi}_l^\top.$$

By letting

$$\boldsymbol{\psi}_l = \boldsymbol{D}^{-1/2}\boldsymbol{\phi}_l \quad \text{and} \quad \boldsymbol{\varphi}_l = \left(\boldsymbol{\phi}_l^\top \boldsymbol{D}^{1/2}\right)^\top = \boldsymbol{D}^{1/2}\boldsymbol{\phi}_l, \tag{18}$$

we can decompose $\boldsymbol{P}$ as follows

$$\boldsymbol{P} = \sum_{l=0}^{n-1} \lambda_l \boldsymbol{\psi}_l \boldsymbol{\varphi}_l{}^\top, \tag{19}$$

where $\boldsymbol{\psi}_l$ is the column right eigenvector and $\boldsymbol{\varphi}_l$ is the column left eigenvector. From Appendix B, the eigenvalues satisfy

$$1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_{n-1}|.$$

We have the following eigen-decomposition of $\boldsymbol{P}^t$

$$\boldsymbol{P}^t = \sum_{l=0}^{n-1} \lambda_l^t \boldsymbol{\psi}_l \boldsymbol{\varphi}_l^\top. \tag{20}$$

Eigenvectors: Since $\{\boldsymbol{\phi}_l\}_{l=0,\dots,n-1}$ is orthonormal, $\boldsymbol{\psi}_l$ and $\boldsymbol{\varphi}_l$ are bi-orthonormal such as

$$\langle \boldsymbol{\psi}_l, \boldsymbol{\varphi}_m \rangle = \boldsymbol{\psi}_l^\top \boldsymbol{\varphi}_m = \delta_{lm} = \begin{cases} 1 & \text{if } l = m \\ 0 & \text{if } l \neq m, \end{cases}$$

where $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = \sum_i x_i y_i$ is the inner (dot or scalar) product of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Besides, the left eigenvector corresponding to the largest eigenvalue $\lambda_0 = 1$, $\boldsymbol{\varphi}_0$, satisfies

$$\boldsymbol{\varphi}_0^\top \boldsymbol{P} = \boldsymbol{\varphi}_0^\top.$$

This eigenvector is thus a stationary distribution. As shown above, the stationary distribution is unique. Therefore, $\boldsymbol{\varphi}_0 = \boldsymbol{\pi}$, and $\boldsymbol{\varphi}_0$ has the form as in (17). The corresponding right eigenvector $\boldsymbol{\psi}_0 = \mathbf{1}_{n\times 1}$ because

$$\boldsymbol{P}\mathbf{1}_{n\times 1} = \mathbf{1}_{n\times 1},$$

which comes from the row of $\boldsymbol{P}$ summing to one.

Moreover, as described in Appendix A, the set of the left eigenvectors $\{\boldsymbol{\varphi}_l\}_{l=0,..,n-1}$ is an orthonormal basis in the so-called *diffusion space* $l_2(\mathbb{R}^n, \boldsymbol{D}^{-1})$, where matrix $\boldsymbol{D}$ is defined in (14). In this space, the norm is defined as follows

$$\|\boldsymbol{x}\|^2_{l_2(\mathbb{R}^n,\boldsymbol{D}^{-1})} = \boldsymbol{x}^\top \boldsymbol{D}^{-1}\boldsymbol{x}, \tag{21}$$

for any column vector $\boldsymbol{x} \in \mathbb{R}^n$.

Now let us consider an arbitrary initial distribution $\boldsymbol{p}_0$ of n nodes. Note that $\boldsymbol{p}_0$ is a column vector of length n. Since $\{\boldsymbol{\varphi}_l\}_{l=0,..,n-1}$ forms a basis of $\mathbb{R}^n$, there exists a set of $a_l \in \mathbb{R}$ such that

$$\boldsymbol{p}_0 = \sum_{l=0}^{n-1} a_l \boldsymbol{\varphi}_l. \tag{22}$$

We can check that

$$\langle \boldsymbol{p}_0, \boldsymbol{\psi}_l \rangle = \boldsymbol{p}_0^\top \boldsymbol{\psi}_l = \sum_{m=0}^{n-1} a_m \boldsymbol{\varphi}_m^\top \boldsymbol{\psi}_l = \sum_{m=0}^{n-1} a_m \delta_{ml} = a_l.$$

Therefore, the coefficient $a_l \in \mathbb{R}$ depends on not only the initial distribution $\boldsymbol{p}_0$ but also the right eigenvectors $\{\boldsymbol{\psi}_l\}$ of the transition matrix. Since $\boldsymbol{\psi}_0(\boldsymbol{x}_i) = 1$ for all $\boldsymbol{x}_i$, then $a_0 = \sum_{i=1}^n p_0(\boldsymbol{x}_i) = 1$.

Given an initial distribution $\boldsymbol{p}_0$, the probability that the Markov chain is in node x_j after t steps is

$$p_t(\boldsymbol{x}_j) = \sum_{i=1}^{n} p_0(\boldsymbol{x}_i) p_t(\boldsymbol{x}_j \mid \boldsymbol{x}_i),$$

or equivalently, using (22) and (20),

$$\begin{aligned}
\boldsymbol{p}_t^\top = \boldsymbol{p}_0^\top \boldsymbol{P}^t &= \sum_{m=0}^{n-1} a_m \boldsymbol{\varphi}_m^\top \sum_{l=0}^{n-1} \lambda_l^t \boldsymbol{\psi}_l \boldsymbol{\varphi}_l^\top \\
&= \sum_{m=0}^{n-1} \sum_{l=0}^{n-1} a_m \lambda_l^t \boldsymbol{\varphi}_m^\top \boldsymbol{\psi}_l \boldsymbol{\varphi}_l^\top \\
&= \sum_{l=0}^{n-1} a_l \lambda_l^t \boldsymbol{\varphi}_l^\top
\end{aligned}$$

where the last equation comes from the fact that $\boldsymbol{\psi}_l$ and $\boldsymbol{\varphi}_l$ are bi-orthonormal. We know that $a_0 = 1$, $\lambda_0 = 1$, and $\boldsymbol{\varphi}_0 = \boldsymbol{\pi}$, we have

$$\boldsymbol{p}_t^\top - \boldsymbol{\pi}^\top = \sum_{l=1}^{n-1} a_l \lambda_l^t \boldsymbol{\varphi}_l^\top = \boldsymbol{p}_0^\top \sum_{l=1}^{n-1} \lambda_l^t \boldsymbol{\psi}_l \boldsymbol{\varphi}_l^\top. \tag{23}$$

Note that $\boldsymbol{\psi}_l^\top \boldsymbol{\varphi}_l = 1$ but $\boldsymbol{\psi}_l \boldsymbol{\varphi}_l^\top$ is a matrix of size $n \times n$. We now turn to discussing the implications on the diffusion properties from the eigen-decomposition in (23)

Eigenvalues: As mentioned above, all eigenvalues are smaller or equal to 1. If the graph is fully connected, as t goes to infinity, λ_l^t tends to 0 for $l > 0$, and the probability $p_t(\boldsymbol{x}_i)$ converges to the stationary probability $\pi(\boldsymbol{x}_i)$ (by Equation (23)). The value of λ_l thus characterizes the convergence speed of the associating direction in the diffusion space: when λ_l is small, the convergence is fast,

and slow for a large λ_l. In other words, the eigenvalues describe the diffusion timescales when the Markov chain goes through the associating directions in the diffusion space. The larger λ_l is, the longer the timescale it takes to go through the corresponding direction. Note that this space has n dimensions that are the sample size and is independent of the dimension of the feature space of $\boldsymbol{X}$.

Spectral decay: In the following discussion for the spectral decay, we consider a weighted graph without a specific kernel function.

The decay of the eigenvalues of $\boldsymbol{P}$ describes the connectivity between nodes in the graph:

- If the graph is fully connected where the weights are all one, the transition matrix has constant entries and rank 1. The matrix has thus one non-zero eigenvalue (which is equal to 1 as shown in Appendix B), and all of the rest eigenvalues are 0. That is the fastest decay case. From an arbitrary node, it can immediately jump to any other node. The connectivity is maximal. All nodes can be considered as one point which requires only one dimension to describe.

- When all nodes in the graph are disconnected, i.e., the transition matrix is the identity matrix. $\boldsymbol{P}$ has full rank n, it thus has n non-zero eigenvalues. These eigenvalues are all equal to 1 since $\boldsymbol{P}\boldsymbol{\nu} = \boldsymbol{\nu}$ for any vector $\boldsymbol{\nu}$ in $\mathbb{R}^n$. That is the lowest decay case where the Markov chain cannot jump from node to node. There is no connection between node, we thus need all n dimensions to describe n isolated nodes.

- The decay usually lies in between the two extreme cases above. It relates to how fast the Markov chain diffuses via nodes: a fast decay shows a fast diffusion, and a low decay accounts for a low diffusion on the graph. The rate of decay depends on the chosen kernel width, the underlying structure, and the intrinsic dimension of the data.

The spectral decay helps determine the dimensions in the diffusion space to describe the data, and achieve the dimensionality reduction. We will discuss that later in Section 3.5 and 4.

3.4 Diffusion distance and diffusion map

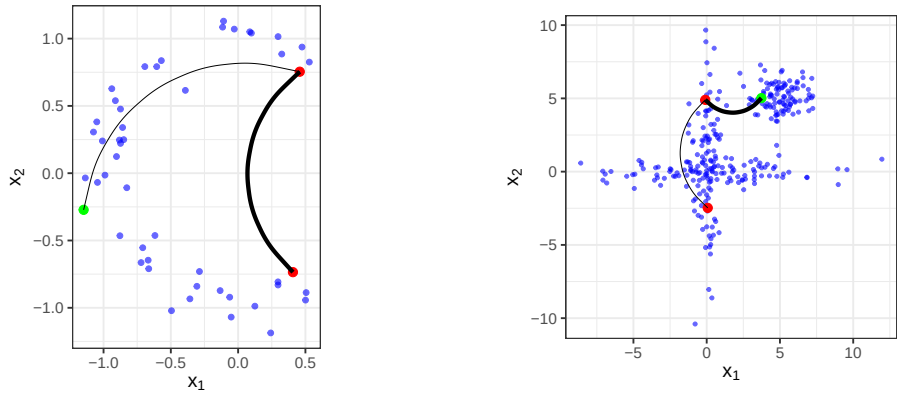
As described in the previous sections, the matrix $\boldsymbol{P}^t$ reveals the global structure of the data. Besides, if the graph constructed on the data is fully connected, there exists an eigen-decomposition of $\boldsymbol{P}^t$. The diffusion space $l_2(\mathbb{R}^n, \boldsymbol{D}^{-1})$ is defined as the basis formed with the left eigenvectors. Putting all together, the eigenvalues and right eigenvectors of $\boldsymbol{P}^t$, diffusion distance, and diffusion map are constructed to characterize the geometry of the data.

Diffusion distance: Again, we consider the Markov chain constructed on the weighted graph constructed from the dataset $\boldsymbol{X}$ and the transition matrix $\boldsymbol{P}$. For a $t \geq 1$ ($t \in \mathbb{N}$), *diffusion distance*, D_t, of two points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}$ is defined as follows

$$\begin{aligned} D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 &= \|\boldsymbol{P}_i^t - \boldsymbol{P}_j^t\|^2_{l_2(\mathbb{R}^n, \boldsymbol{D}^{-1})} \\ &= (\boldsymbol{P}_i^t - \boldsymbol{P}_j^t)\boldsymbol{D}^{-1}(\boldsymbol{P}_i^t - \boldsymbol{P}_j^t)^\top \\ &= \sum_{k=1}^{n} \frac{1}{D_k}(p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i) - p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_j))^2 \end{aligned} \tag{24}$$

where $\boldsymbol{P}_i^t = p_t(. \mid \boldsymbol{x}_i)$ is i^{th} row vector of the matrix $\boldsymbol{P}^t$, and the norm $\|.\|_{l_2(\mathbb{R}^n, \boldsymbol{D}^{-1})}$ is defined in (21). $\boldsymbol{D}$ is the diagonal degree matrix of the graph, and D_k is the degree of $\boldsymbol{x}_k$ defined in (14). If $\boldsymbol{x}_k$ is low-density, then D_k is small and the probabilities $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$, $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_j)$ are close for any $\boldsymbol{x}_i$, $\boldsymbol{x}_j$. Therefore, the weights $\frac{1}{D_k}$ makes the difference, $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i) - p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_j)$, to be more significant on low-density nodes.

As explained in Section 3.2, the probability $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$ is large when $\boldsymbol{x}_k$ and $\boldsymbol{x}_i$ are highly connected via other nodes along the underlying data structure and vice versa. From the definition, the diffusion distance $D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is small when $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$ and $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_j)$ are close for all $\boldsymbol{x}_k$,



(a) C-curve data

(b) non-spherically symmetric data

Figure 7: Diffusion distance on (a) C-curve (when $t = 8$) and (b) a non-spherically symmetric data (when $t = 64$), represented by the thickness of the connected curves between big points (a thicker curve corresponds to a longer distance). In C-curve, two red points have the largest distance in the hidden space (shown in Fig. 3a). In non-spherically symmetric data, two red points belong to a cluster, and the green one lies on another cluster.

meaning that both $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ well connect to $\boldsymbol{x}_k$ or poorly connect to this node. That happens when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are highly connected. When $\boldsymbol{x}_i, \boldsymbol{x}_j$ are poorly connected, the difference of $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_i)$ and $p_t(\boldsymbol{x}_k \mid \boldsymbol{x}_j)$ is large, leading to the diffusion distance of these nodes being large as well. Consequently, this distance reflects the underlying structure in terms of the connectivity of the data points. The points are close if they are similar. Moreover, since D_t is defined as the summation of many paths, it is very robust to noise perturbation.

Setting a large enough value for t so that $\boldsymbol{P}^t$ describes the global connectivity of the data well enough, diffusion distance reveals the intrinsic geometry of the data. For example, in Figure 7 we consider two datasets: C-curve (see Figure 3), where a straight line is nonlinearly embedded into two-dimensional space; and non-spherically symmetric data containing two clusters in different distribution. We obtain the following results:

- In Figure 7a, the Euclidean distance between the red and the green points is larger than the distance between two red ones. However, the diffusion distance represented by the thickness of the connected curves (i.e., a thicker curve represents a longer distance) makes more sense where the distance between two red points is larger. In this way, the diffusion distance contains the information of the underlying geometry of the data.

- The two red points, in Figure 7b, lie on the same cluster while the green one belongs to another cluster. Although the Euclidean distance between two red points is larger than that of the top red and the green points, the diffusion distance gives an opposite result: two red points are closer than the red and the green ones. Diffusion distance, therefore, provides a good measure of the disconnectivity of the data points.

Note that the diffusion distance with a small t does not provide a satisfying measure for the global connectivity since the matrix $\boldsymbol{P}^t$ only describes local connectivities. For instance, the distance between the two red points in C-curve is smaller than that between the one between the top red and green points; or in non-spherically symmetric data, the top red point is closer to the green one than to the bottom red one.

The diffusion distance defined in (24), however, is computationally expensive. Using the eigen-decomposition of $\boldsymbol{P}$ described in the previous section is a classic technique for lower computational cost. As shown in Appendix C, the diffusion distance in (24) can be rewritten in terms of eigenvalues and eigenvectors as follows

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=1}^{n-1} \lambda_l^{2t} \left(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j)\right)^2, \tag{25}$$

where λ_l is the eigenvalue and $\boldsymbol{\psi}_l$ is the right eigenvector of $\boldsymbol{P}$. Since $\boldsymbol{\psi}_0 = \mathbf{1}_{n\times 1}$, the term in the sum in (25) for $l = 0$ is omitted. From that, the dimensionality of the diffusion space is reduced to $n - 1$.

Diffusion coordinates and diffusion map: Each vector $\lambda_l^t \boldsymbol{\psi}_l = (\lambda_l^t \psi_l(\boldsymbol{x}_1), ..., \lambda_l^t \psi_l(\boldsymbol{x}_n))$ of length n in the sum in (25) defines a *diffusion coordinate* of data points. And a so-called diffusion map, $\Psi_t : \boldsymbol{X} \to \mathbb{R}^{n-1}$ embeds the dataset $\boldsymbol{X}$ into a Euclidean space of $n - 1$ dimensions, that is

$$\Psi_t(\boldsymbol{x}) = \begin{pmatrix} \lambda_1^t \psi_1(\boldsymbol{x}) \\ \lambda_2^t \psi_2(\boldsymbol{x}) \\ \vdots \\ \lambda_{n-1}^t \psi_{n-1}(\boldsymbol{x}) \end{pmatrix} \in \mathbb{R}^{n-1}. \tag{26}$$

Each dimension in the new space is characterized by one diffusion coordinate.

Using the diffusion map and the diffusion coordinates, the diffusion distance in (25) between two data points, $\boldsymbol{x}_i, \boldsymbol{x}_j$, is equal to the Euclidean distance between two mapped points, $\Psi_t(\boldsymbol{x}_i), \Psi_t(\boldsymbol{x}_j)$,

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\Psi_t(\boldsymbol{x}_i) - \Psi_t(\boldsymbol{x}_j)\|.$$

The equivalence with the Euclidean distance makes diffusion distance be more feasibly used by classical clustering methods, requiring the Euclidean distance, such as k-means.

Furthermore, in terms of the diffusion coordinates we can see the scaling role of t:

- The timescale t affects only the eigenvalues, not the eigenvectors. It thus rescales the diffusion coordinates.
- When t is increasing, the distance in (25) is smaller since $|\lambda_l| \leq 1$. The discovered embedding characterized by the diffusion coordinates becomes more global.

Besides, λ_l^{2t} goes to zero exponentially in t. The terms, $\lambda_l^{2t} \left(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j)\right)^2$, in the sum in (25) associated with the zero values of λ_l^{2t} can be removed without changing the diffusion distance. Moreover, by keeping only the first q diffusion coordinates the Euclidean space, the dimensionality of the discovered embedding is reduced to q. The Euclidean distance in the q-dimensional space is an approximation of the diffusion distance.

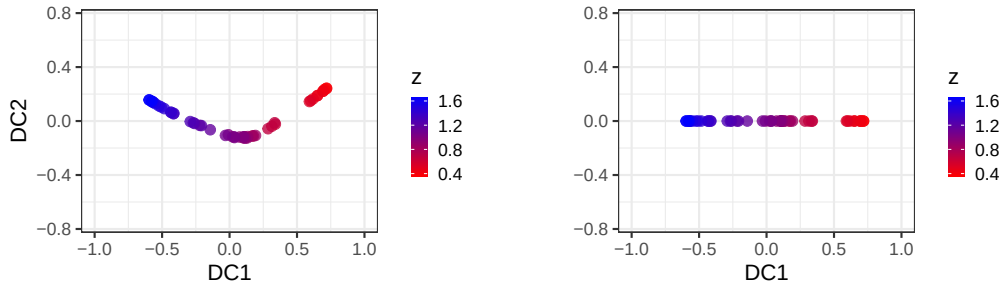
3.5 Intrinsic geometry and dimensionality reduction

The diffusion map embeds the data points $\boldsymbol{X}$ into a Euclidean space $\mathbb{R}^{n-1}$ in the way that the diffusion distance in the original space is equal to the Euclidean distance in the diffusion space. With a large enough timescale t, the diffusion distance reveals the underlying intrinsic geometry of the data. Therefore, the diffusion map reorganizes the data according to the mutual diffusion distance and preserves the intrinsic geometry of the data.

Figure 8 shows an example to illustrate the discovering intrinsic data geometry of the diffusion map. Again, the C-curve data (in Fig. 3b) was considered. We performed the diffusion map with the chosen parameters, $\sigma = 0.5$ and $t = 8$. These choices come from the discussion about the kernel width in Figure 5 and the timescale in Figure 6. Unlike PCA and MDS, the diffusion map successfully discovers the (1D) hidden data (shown in Fig. 3a) with the first diffusion coordinate in which the color order is preserved (Fig. 8b).

The dimensionality reduction is achieved by retaining q diffusion coordinates associated with the dominant eigenvalues such that the Euclidean distance in the truncated space is still a good approximation of the diffusion distance. In many cases, a spectral gap (a sudden fall) appears right after the q^{th} eigenvalue. The dimensionality can be reduced to q [3] by choosing a large enough timescale t to reach a given accuracy $\delta > 0$ for the diffusion distance. More precisely, t is chosen such that $q = \max\{l : |\lambda_l|^t > \delta |\lambda_1|^t\}$ [4].

If there is not any spectral gap, the problem of identifying the retained dimensions becomes more difficult. Since $n - 1$ eigenvectors are describing the data which are usually in fewer dimensions, many of the eigenvectors capture the same direction of the original space. The significant



(a) (b)

Figure 8: Diffusion coordinates: (a) the first two, and (c) the first diffusion coordinates (DC) computed on C-curve data (shown in Fig. 3b) with $\sigma = 0.5$ and $t = 8$. The color is encoded by the value of z in Figure 3a.

eigenvectors can thus be separated by redundancy. Such redundancy is a combination or function of the previous eigenvectors. Further analysis of relations between eigenvectors is required to remove this redundancy [13].

4 Computational experiments

All computational experiments were done in R using the package `diffusionMap`[3].

4.1 S-shape

4.1.1 Data generation

S-shape is a two-dimensional manifold (see in Fig. 9a) embedded into a three-dimensional space using a nonlinear transformation, as shown in Figure 9. We will see if the diffusion map can discover the underlying data in two dimensions.

S-shape is generated from uniformly distributed points in $\mathbb{R}^2$ embedded into $\mathbb{R}^3$. In particular, $x_1 \sim U(0,1)$ and $x_2 \sim U(0,1)$, the data point is a nonlinear mapping

$$\boldsymbol{y} = \begin{pmatrix} \sin(w) \\ Hx_2 \\ \text{sign}(w)(\cos(w) - 1) \end{pmatrix} \tag{27}$$

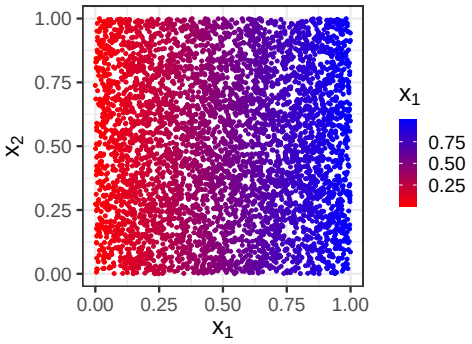
where $w = 3\pi(x_1 - 0.5)$, $\text{sign}(w)$ is the sign of w, and H is the width (corresponding to y_2 axis) of S-shape.

In the three-dimensional space, the point distribution is not isotropic as in the two-dimensional hidden space. It is because of the difference between the width and the length along S-shape. Later in this report, we will perform the diffusion map on S-shape with two different width values. One with a large width $H = 8$ (shown in Fig. 9b), where the width is approximately the same as the length, and the point distribution is nearly isotropic. And another one with a small width $H = 2$ (shown in Fig. 9c), where there is a large difference between the length and the width, and the point distribution is anisotropic.

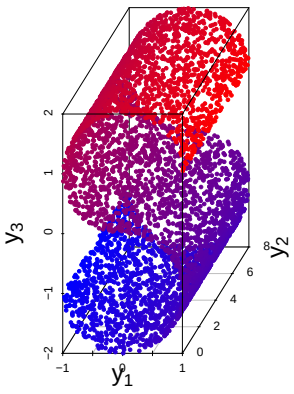
These data are examples of the failure in describing the nonlinear structure of the Euclidean distance. Similar to C-curve data, nonlinear embedding does not preserve the Euclidean distance order in S-shaped. For instance, the largest separation in x_1 in Figure 9a does not have the greatest Euclidean distance in the three-dimensional space in Figure 9b and 9c. Therefore, the methods based on Euclidean distance fail to discover the embedding.

From the generation in (27), there is a parameter H to control the width of S-shape in 3-D embedding space. If the data were stretched, we would get a rectangular shape whose length is approximately equal to 8 and whose width is H. The diffusion process will change when the value

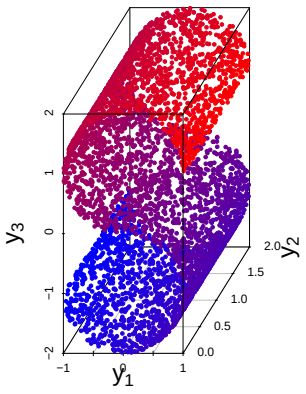
[3] `https://CRAN.R-project.org/package=diffusionMap`



(a)



(b)



(c)

Figure 9: S-shape data of size 5000: (a) two-dimensional manifold where the points are uniformly distributed, (b) three-dimensional nonlinear embedding with a large width $H = 8$, and (c) three-dimensional embedding with a small width $H = 2$. The color in both of them is encoded by the value of x_1 in (a).

of H is altered. We will look at two examples, one with large and one with small width to get insights about the two different possibilities.

4.1.2 S-shape with large width

First, we set a width $H = 8$ that is approximately the same as the length of the S-shape (as shown in Fig. 9b). The point distribution is quite isotropic in this case. The isotropic Gaussian kernel seems to be a reasonable choice.

Kernel width: Choosing a good kernel width is the first thing we have to do before computing the diffusion map. As discussed above, the kernel width depends on the distribution, the density, and the geometry of the data. In this case, the points are uniformly distributed in the hidden space. In the embedding space, the data distribution is still uniform, but it is quite different along the length and the width of the S-shape. However, the difference is not significant, and the kernel width can be chosen as a constant.

The kernel width is estimated by using the k-nearest neighbor ($k$nn) distance. In particular, the median of $k$nn is an estimate of σ, where k is 1% of the sample size, that gives $\sigma = 0.5$.

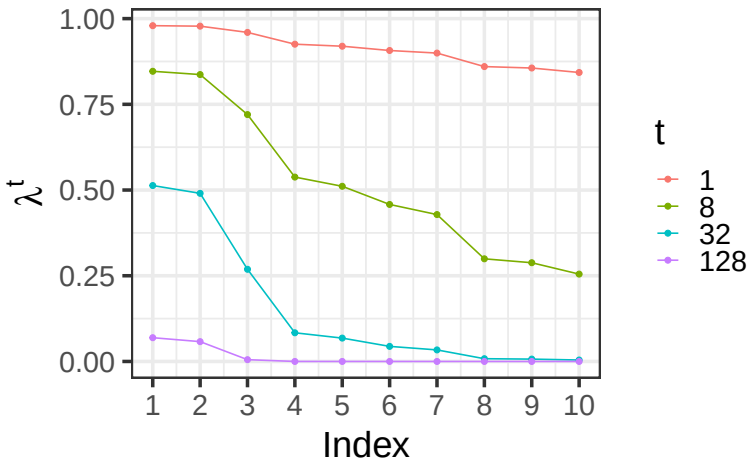


Figure 10: The first ten from $n-1$ eigenvalues of the matrix $\boldsymbol{P}^t$ constructed on S-shape with large width ($H = 8$, Fig. 9b) using the estimated kernel width $\sigma = 0.5$.

Eigenvalues: The spectral gap is next verified by plotting the first ten eigenvalues of the matrix $\boldsymbol{P}^t$ in Figure 10. It is easy to see a gap after the second eigenvalue (in the green and blue lines corresponding to $t = 8$ and 32). The first two eigenvalues are both large and followed by a big fall. This gap is evidence that the dimension of the intrinsic geometry is 2.

Figure 10 also provides a way to choose the timescale parameter t by observing the value of λ_l^t. Choosing $t \geq 128$ results in eigenvalues dropping to almost zero passed the third one, it seems to be a good choice to get a good approximation of the diffusion distance.

Besides, the spectral decay displayed in Figure 10 shows the connectivity of the graph:

- When $t = 1$, the eigenvalues are quite flat. It closes to the extreme case where the graph is disconnected. The connectivity of the graph is local in this case. In particular, the nodes are only connected to their neighbors and disconnected to other points outside their neighborhood. We need many diffusion coordinates associating with non-zeros eigenvalues to describe the data. The dimension is thus very large.

- When t gets larger, the connectivity of the graph goes from local to global. There is a connection between long-distance nodes. The dimension necessary to describe the data is smaller. It is equal to the number of non-zero eigenvalues

- When t gets very large ($t > 128$), all eigenvalues drop to zero, and the graph reaches the stationary state, where all possible links between nodes are possible. As the diffusion distances between all data points approach zero, all data points can be considered as one node, and the dimension necessary to describe the data is zero.

Diffusion coordinates: The embedding of the S-shape data into the first two diffusion coordinates, displayed in Figure 11, shows that the diffusion map successfully unfolds the two-dimensional manifold. However, we do not get a perfect rectangle in the diffusion space and the points are non-uniform. More precisely, the left and right boundaries are not equal; the points on them are denser than the points in the middle region where there are holes (easy to see when $t = 1$). The reason is that the density of sampling points is not entirely uniform. In particular,

- The point densities in the left and the right boundary are not equal. This leads to the unequal of these boundary. They will become more equally if the sample size gets larger.

- In the diffusion space, the connectivity is normalized. That is, the total connectivity of each point is always one. The points near the boundary are highly connected to their neighbors and poorly connected to further points. Their connectivity to the spatial extent near the boundary becomes higher than in the middle region. On the other hand, points located in

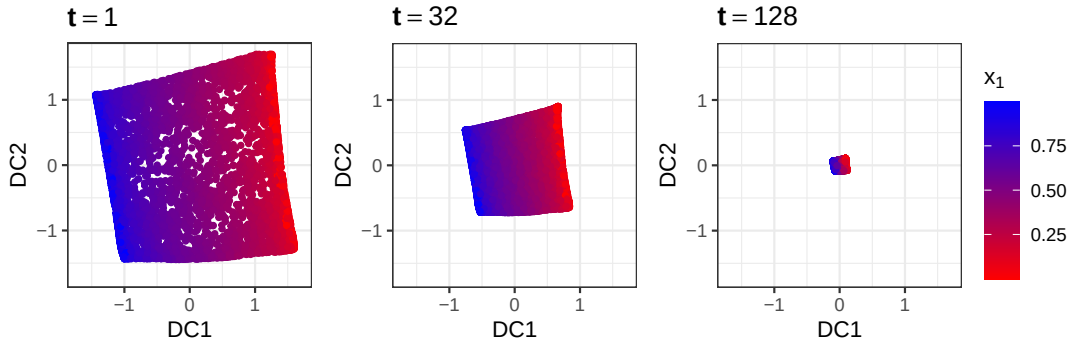


Figure 11: The first two diffusion coordinates (DC) of the embedding space of the S-shape with large width ($H = 8$, Fig. 9b) in various timescales t. The color is encoded by the value of x_1 in Figure 9a. The estimated kernel width $\sigma = 0.5$ was used to compute the diffusion map.

the middle have reasonably isotropic connectivity. That is why density grows relatively on the boundary compared to spatial extents inside the "rectangle".

- The low-density points in the middle of the (3-D) S-shape cause the holes inside the discovered manifold. The holes' size depends on the spatial extent within which we define the local similarity, which is associated with the kernel width. A larger value of σ makes smaller holes. Besides, when the timescale t gets larger, the connectivity becomes global and the holes are smaller.

The role of t in scaling is represented clearly in this case: when t increases, the embedding manifolds in Figure 11 become smaller.

Comparison with other methods: We investigate the performance of PCA, MDS on the S-shape with a large width dataset, and compare them to the diffusion map.

PCA: Since the data points are given in Cartesian coordinates, the results of PCA and MDS are the same as shown in Figure 12.

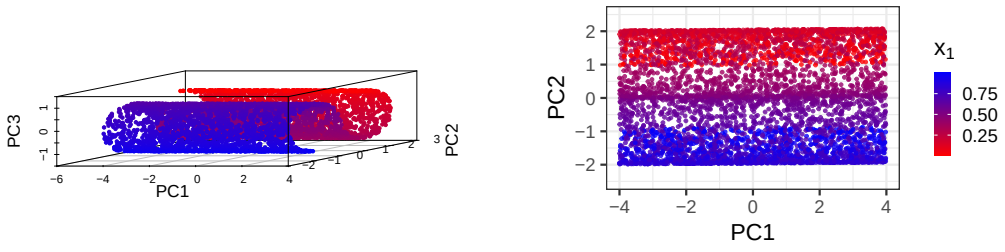
In Figure 12a, we obtained the following results for PCA:

- The first principal component (PC1) captures y_2 which has the largest variance in the three-dimensional S-shape (Fig. 9b);
- The second PC, which is orthogonal to PC1, captures the most variation in the remaining direction, corresponding to y_3;
- The third PC represents y_1, which is orthogonal to y_2 and y_3.

Figure 12b shows the projection of the S-shape on the first two PCs. The points in the top and bottom turns of S-shape are superposed (represented by the color). Therefore, PCA fails to discover the (2D) hidden space of the data.

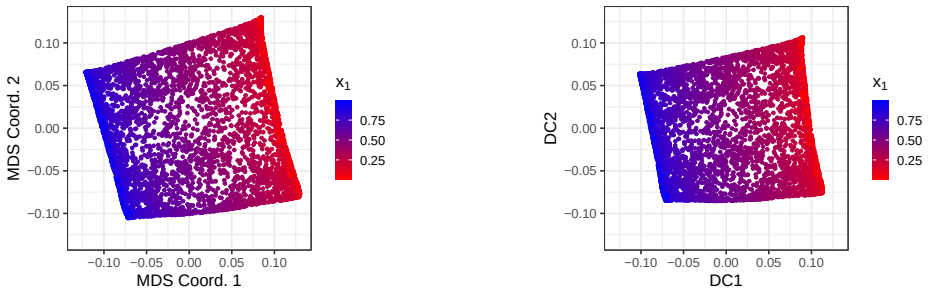
MDS using the diffusion distance: As discussed above, MDS does not require the coordinates in its input. Next, we look at the approach using the diffusion distance defined in (24) as the input of MDS.

Figure 13 displays (2-D) embeddings found by MDS using the diffusion distance (in Fig. 13a) with parameters $\sigma = 0.5, t = 128$, and by diffusion map (in Fig. 13b) with the same parameters. Note that the Euclidean distance using the full expansion of coordinates in these two cases is equal. We get a quite similar result in shape and point distribution, but a slight difference in the scale and angle.



(a) (b)

Figure 12: PCA: (a) Three and (b) two principal components of the S-shape with large width ($H = 8$, Fig. 9b). The color is encoded by the value of x_1 in Figure 9a.

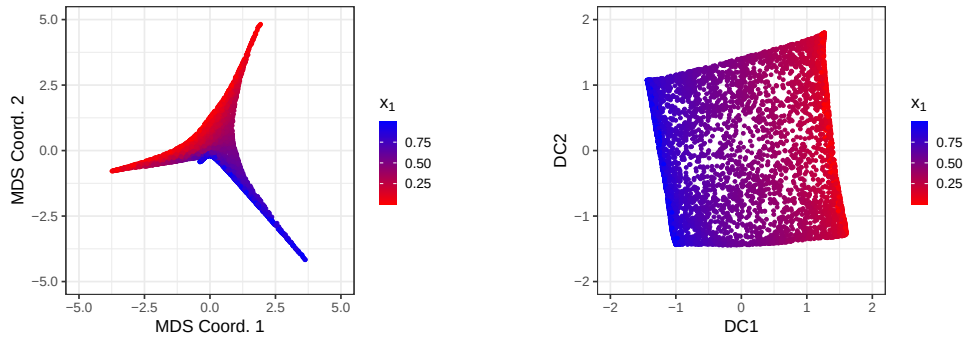


(a) (b)

Figure 13: Two-dimensional embeddings of S-shape with large width ($H = 8$, Fig. 9b) found by: (a) MDS using diffusion distance and (b) diffusion map. The estimated kernel width $\sigma = 0.5$ and the timescale $t = 128$ were used to compute both of them. The color is encoded by the value of x_1 in Figure 9a.

- $t = 128$ is large enough to get a global connectivity description of the data from the diffusion distance. Moreover, two dimensions are enough to capture the data structure. It is not surprising that MDS found similar embedding as the diffusion map.
- The scaling difference between the embeddings found by MDS (in Fig. 13a) and diffusion map (in Fig. 13b) comes from two unequal errors in the approximations of the diffusion distance by the Euclidean distance. The error is because the high indices (from the third) of the coordinates vanish in the Euclidean distance. In MDS, the coordinates are determined by the eigenvalues and eigenvectors of the (symmetric) Gram matrix (obtained by using double centering of the distance matrix). At the same time, the diffusion coordinates are associated with the eigen-decomposition of the (asymmetric) transition matrix. The difference in coordinates between MDS and diffusion map causes a gap between two errors, leading to the difference in the scale.
- The difference in the angle between Figures 13b and 13a is due to the coordinate system being rotated in MDS.

Although the embeddings are similarly found by MDS using diffusion distance and the diffusion map in the case of large timescale $t = 128$, they are very different when $t = 1$, as shown in Figure 14. While in both settings, the diffusion map gives similar shapes in different scales, the first two coordinates are very different in MDS. When $t = 1$, the diffusion distance does not describe the global data structure well: it only provides a measure for the local similarity. Therefore, like PCA above (shown in Fig. 12), MDS does not give a correct embedding. The difference in the results between PCA and MDS can be affected by the use of kernel function and the normalization. The first two diffusion coordinates, in another way, form a shape very close to the expected embedding



(a) (b)

Figure 14: Two-dimensional embeddings of S-shape with large width ($H = 8$, Fig. 9b) found by: (a) MDS using diffusion distance and (b) diffusion map. The estimated kernel width $\sigma = 0.5$ and the timescale $t = 1$ were used to compute both of them. The color is encoded by the value of x_1 in Figure 9a.

as shown in Figure 13b. There is a difference in scaling affected by the timescale t. That is an advantage of the diffusion map.

4.1.3 S-shape with small width

We consider the three-dimensional S-shape with a small width $H = 2$, displayed in Figure 9c. In this case, the width is much smaller than the length. The point distribution is thus anisotropic. However, we simply use the isotropic Gaussian kernel with a constant kernel width to construct the diffusion map. Note that the result can be more interesting if two different kernel widths or an anisotropic Gaussian kernel function are used. However, we did not consider these cases, which are out of the scope of this report.

Kernel width: A constant kernel width is estimated by using the $k$nn distance as the previous example, that gives $\sigma = 0.25$. It is half of the last kernel width. It shows that the points are much denser, although the length of S-shape does not change.

Eigenvalues: A spectral gap is not clearly visible in Figure 15 showing the first ten eigenvalues of the matrix $\boldsymbol{P}^t$. However, the spectral decay slows down after the fifth eigenvalue until the seventh one. Therefore, the first seven eigenvectors were analyzed to determine the essential dimensions and the number of retained diffusion coordinates.

Diffusion coordinates: The first seven diffusion coordinates are shown in Figure 16 where the timescale $t = 1$. We can see that:

- The first to the fourth DCs describe the length along S-shape. While the first DC is enough to capture the length, three remaining DCs are all the functions of the first, in other words, they are redundant.
- The fifth to the seventh DCs give more information about the structure of the data. They describe the width of S-shape. Why is the width's information of the S-shape given by the fifth coordinate far from the first coordinate but not from the second one (like with the case of large width)? A reason is that the point distribution is anisotropic. The points are denser in the direction of the width comparing to the direction along S-shape, which leads to the connectivity becomes much higher. Therefore, the fifth eigenvalue is much smaller than the first one. Using more than one kernel width can give a more exciting result.
- The first and fifth DCs are important and should be kept in a sensible and low-dimensional representation of the data. The DCs between them are redundant and should be removed

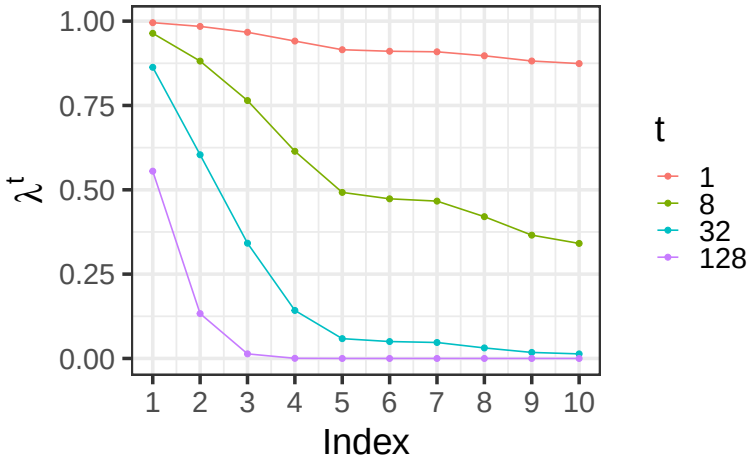
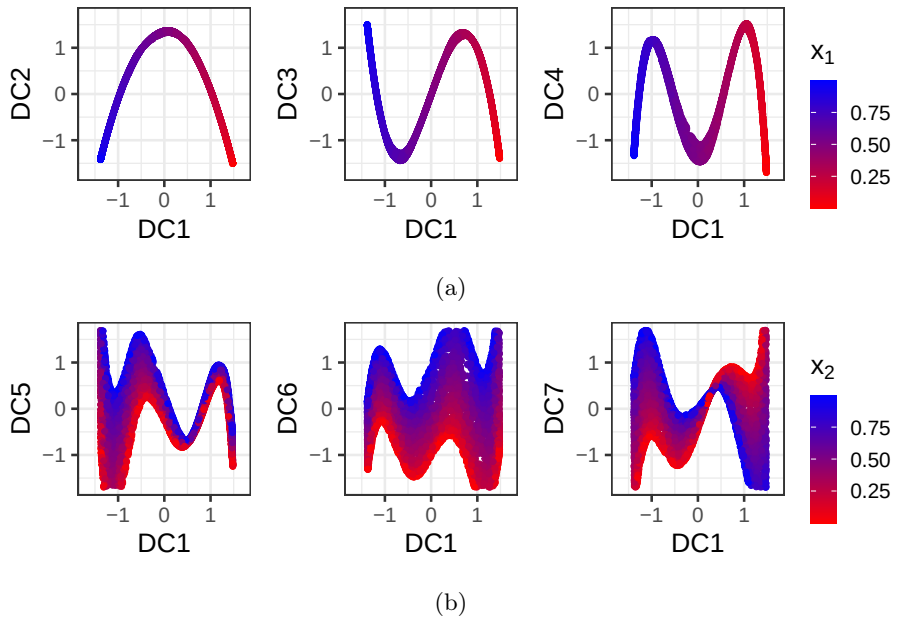


Figure 15: The first ten from $n-1$ eigenvalues of the matrix $\boldsymbol{P}^t$ constructed on S-shape with small width ($H = 2$, Fig. 9c) using the estimated kernel width $\sigma = 0.25$.



(a)

(b)

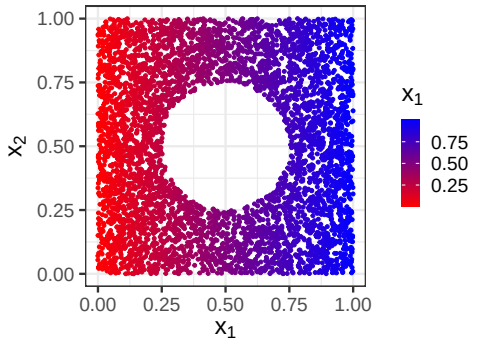
Figure 16: (a) The second, third, fourth diffusion coordinates (DC) versus the first DC with the timescale $t = 1$ of the S-shape with a small width ($H = 2$, Fig. 9c). The color encoded by the value of x_1 shown in Figure 9a. (b) The fifth, sixth, seventh DC versus the first DC (with $t = 1$). The color encoded by the value of x_2 shown in Figure 9a. The estimated kernel width $\sigma = 0.25$ was used to compute the diffusion map.

- Nevertheless, the first 5 DCs should be retained to get a good approximation of the diffusion distance using the Euclidean distance. The timescale should be not too large, 32 or smaller, so that the fifth eigenvalue does not become too small.

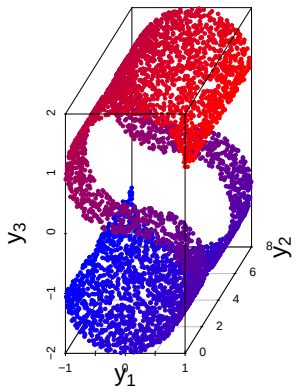
4.2 S-shape with hole

4.2.1 Data generation

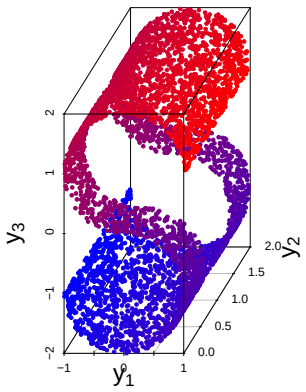
The dataset, shown in Figure 17a, is an example of a non-convex manifold. It is a modification of S-shape by discarding all points inside the circle located in the middle of the manifold. In particular,



(a)



(b)



(c)

Figure 17: S-shape with hole data of size 5000: (a) two-dimensional hidden manifold where the points are uniformly distributed with removing those inside the circle of radius 0.5 centered at $(0.5, 0.5)^\top$, and (b) three-dimensional nonlinear embedding with a large width $H = 8$, and (c) three-dimensional embedding with a small width $H = 2$. The color is encoded by the value of x_1 in (a).

all uniformly distributed points landing into the circle centered at $(0.5, 0.5)$ are discarded in two-dimensional embedding space. This manifold is then embedded into $\mathbb{R}^3$ using the Equation (27) as in the case of normal S-shape. Like the previous example, we will consider two cases: one with a large width $H = 8$ (shown in Fig. 17b) and another with a small width $H = 2$ (Fig. 17c).

In a non-convex manifold like S-shape with hole, the geodesic distance (distance along the manifold [11]) between points near the left and right boundaries is longer than the Euclidean distance since it goes around the hole. We will evaluate whether there is a difference between the diffusion distance and the Euclidean distance and how the diffusion map outlines a non-convex manifold.

4.2.2 Experiment with diffusion map

We will see whether the diffusion map works well with a non-convex manifold. The S-shape with hole and large width, shown in Figure 17b, is considered to compare with the case without the hole. We also discuss the result of the diffusion map for the case with a small width (shown in Fig. 17c).

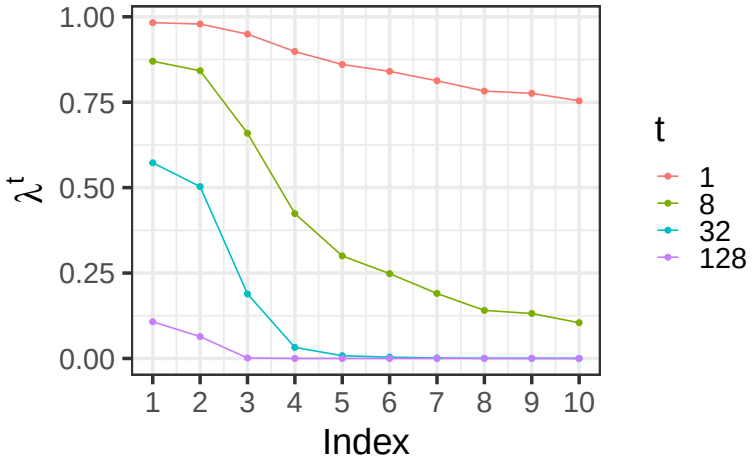
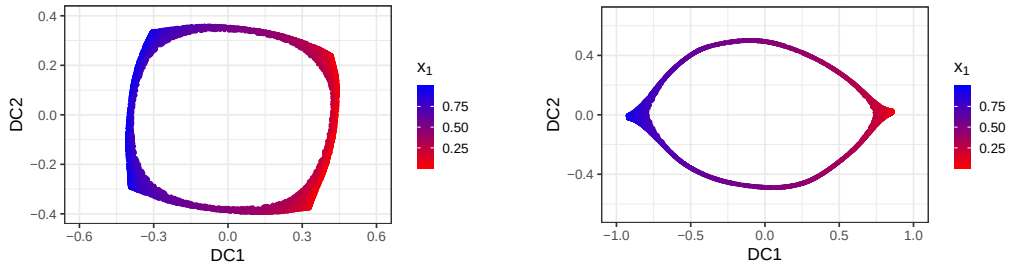


Figure 18: The first ten from $n-1$ eigenvalues of the matrix $\boldsymbol{P}^t$ constructed from S-shape with hole and a large width (in Fig. 17b where $H = 8$) using the estimated kernel width $\sigma = 0.45$.



(a) (b)

Figure 19: The first two diffusion coordinates of (2D) embeddings of: (a) S-shape with hole and a large width (as shown in Fig. 17b) using the estimated kernel width $\sigma = 0.45$ and the timescale $t = 128$, and (b) S-shape with hole and a small width (as shown in Fig. 17c) using the estimated kernel width $\sigma = 0.23$ and the timescale $t = 128$. The color in both plots is encoded by the value of x_1 in Figure 17a.

Kernel width: Again, the median of $k$nn distances is used to estimate the kernel width, that gives $\sigma \approx 0.45$. This value is smaller than the case without hole because the data points become denser when the same number points (5000) is located in a smaller spatial extent (missing a large hole inside).

Eigenvalues: The spectral decay in Figure 18 is similar to that of the S-shape without the hole in Figure 10, although all eigenvalues are larger. With the hole's presence, the points become less connected between the left and right boundaries in the embedding space. It follows that the speed convergence to the stationary distribution is slower. Therefore, all eigenvalues get larger than those in the case without the hole. Besides, like the S-shape with a large width, there is evidence for a two-dimensional embedding with the appearance of a spectral gap after the second coordinate.

Diffusion coordinates: Looking at two first diffusion coordinates in Figure 19a the two-dimensional embedding is decided, and the color order is preserved. However, the hole is enlarged comparing to the hidden manifold (shown in Fig. 17a). With the presence of the hole, the points near a boundary are more highly connected than the points near another boundary. The boundaries are nearly disconnected and only linked to each other by longer paths.

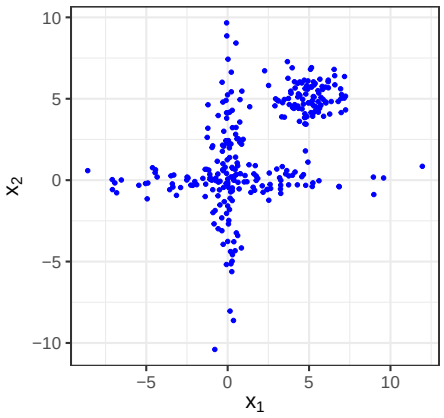


Figure 20: non-spherically symmetric data: 300 two-dimensional points are distributed in two clusters, a star (non-spherical) and a disk (spherical). The distribution of points is non-uniform in both and different between the two clusters.

We get an interesting result in Figure 19b, where the two-dimensional manifold of the data with a small width is nicely discovered. Unlike the case without the hole, the embedding is represented by the first two diffusion coordinates. However, the length of the left and right boundaries of the hidden manifold (shown in Fig. 17a) are not described correctly. It is because, in the embedding space (shown in Fig. 17c), the points near each of these boundaries are strongly connected but nearly disconnected to those in another boundary.

4.3 Non-spherically symmetric data

4.3.1 Data

The data have 300 two-dimensional points which are distributed in two clusters: a non-spherical star shape and a disk, as shown in Figure 20. The points distribution is non-uniform and different between clusters: the points are denser in the disk cluster than the star one. It is an example showing that some clustering methods based on the Euclidean distance fail, such as k-means. The reason is that the Euclidean distance cannot describe well the disconnection of two clusters: many points in the star cluster have a larger distance than two points separated into two clusters. We can evaluate if the diffusion distance is able to describe the dissimilarity between points and separate the disconnection of the data.

4.3.2 Experiment with diffusion map

Kernel width: The data points are non-uniformly distributed, as shown in Figure 20. Besides, the point density is different between the two clusters: it is denser in the disk than in the star. However, we choose a constant for the kernel width by using the median of $k$nn distances. The sample size in this example is quite small (300 observations). The k is thus chosen to be 3% of the sample size, that gives $\sigma \approx 0.8$.

Diffusion coordinates: Figure 21 shows a clear separation of the two clusters in the data (shown in Fig. 20) using only the first diffusion coordinate. In the plot of the first two DCs (the left one in Fig. 21), some blue points appear below the star cluster's other blue points. These points are associated with three points at the bottom of the star in the right plot. It is because with a small, constant kernel width, these points are dissimilar to the star cluster's rest points. However, in the diffusion space (the left plot), these points are closer to the blue cluster than to the red one. Therefore, the diffusion distance successfully separates the clusters in this example, whereas k-means fails.

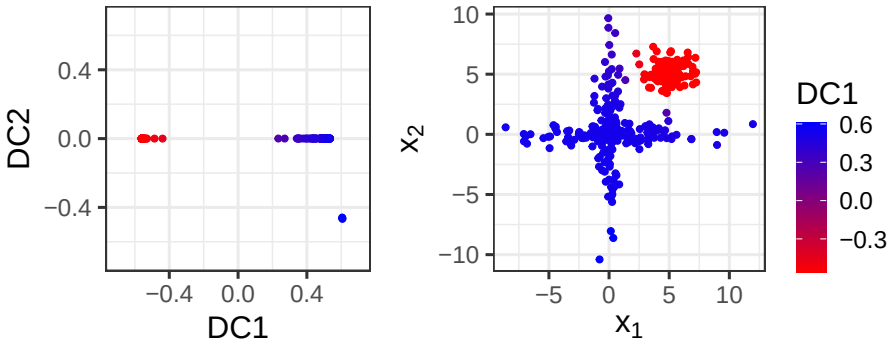


Figure 21: Left plot: the first two diffusion coordinates of the data shown in Figure 20, and the right plot: the original data with color (in both plots) encoded by the value of the first DC.

5 Conclusion

In this report, the construction of the diffusion map in the context of dimensionality reduction was explained in detail. It starts with the weighted graph on data, the local connectivity was defined by using the Gaussian kernel function. The combinations of local connectivities to form global connectivities was discussed based on diffusion on the Markov chain. Finally, the diffusion map, together with diffusion distance and diffusion coordinates, were defined. Specifically, the diffusion map embeds the data into a lower-dimensional Euclidean space in which the distance is approximately the diffusion distance in the original feature space. The diffusion map is easy to implement from the construction as it only uses eigen-decomposition; also, it is very robust to noise. However, choosing the parameters defining its behavior can be tricky.

We also discussed the meaning, the role of the method's parameters (the kernel width σ, timescale t, and the dimensions q of the discovered embedding) with simple illustrating examples. While the kernel width characterizes the local connectivity, the timescale plays an essential role in global geometrical information captured by the diffusion distance. The dimension q relates to the intrinsic dimension of the underlying nonlinear manifold where the data reside on, and directly defines the reduction in dimensionality. From that, a simple method for choosing these parameters was suggested: σ is chosen by analyzing the local density of data points that is defined as their k-nearest neighbor distances; an analysis of the spectral decay or diffusion coordinates is required to choose q; and the choice of t, together with q, are set to best approximate the diffusion distance by using the Euclidean distance.

To get insights into the method, some toy examples were performed and discussed at length. It appears that the diffusion map successfully captures a low-dimensional embedding when data are nonlinearly embedded into a higher dimensional space. Both uniformly distributed and non-convex manifolds were verified. Two comparisons were made between the diffusion map and PCA, and between diffusion map and MDS, using the diffusion distance. They confirmed the ability of this method to capture the geometry at multiple scales. Besides, we also showed that the diffusion distance provides us with an appropriate similarity measure for clustering of non-convex dataset.

6 Discussion

Choosing kernel width plays an important role in the success of the diffusion map. With a good kernel width, the intrinsic data structure is uncovered correctly. In this report, we chose σ by using k-nearest neighbor distances. However, when the sample size is small, choosing a suitable k is more difficult. Moreover, problems arise when the data are anisotropic or/and non-uniform. There are other criteria for the choice of a constant σ discussed in [17], and [7] could be considered in further study. Besides, in [15] Rohrdanz, Zheng, Maggioni, and Clementi used different σ for

each data point, which would be very interesting to try. An anisotropic Gaussian kernel function proposed in [18] was applied in some studies in [19] and [9].

Diffusion distance is useful to describe the dissimilarity of data points and a powerful tool to discover the clusters. In [14], Richards, Freeman, Lee, and Schafer proposed diffusion k-means, a clustering method combining k-means and diffusion maps. It was applied to solve problems in astrophysics.

Coifman and Lafon introduced a family of diffusion maps in [4] with the presence of a parameter $\alpha \in \mathbb{R}$, which specifies the influence of the density on the diffusion. The method we discussed in this report corresponds to $\alpha = 0$ where the effect of the density is maximal. In [4], they also discussed other cases when $\alpha = 1/2$ for an intermediate effect and $\alpha = 1$ for the null effect that gave good improvements on the existing results. The embedding found by diffusion maps with $\alpha = 1$ is independent of the points density. It thus provides another view of the data apart from the statistics. An application of this case can be seen in [7].

The applications of diffusion maps, in reality, are an interesting problem we would like to study further. For example, in biology, it was used to reduce the dimension of single-cell data and order the cells as an attempt to capture the expected differentiation structure [7]. In [20], it was applied in dimensionality reduction for multi-dimensional gene expression data, which provides useful information to cluster cancer samples. In image processing, the diffusion maps help to speed up the vector multiplication recognition system and increase the accuracy in face recognition [1].

References

[1] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz. Fast high dimensional vector multiplication face recognition. In *2013 IEEE International Conference on Computer Vision*, pages 1960–1967, 2013.

[2] Adi. Ben-Israel and T. N. E. Greville. *Generalized inverses : theory and applications*. Springer, New York, 2nd ed. edition, 2003.

[3] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni, and B. Nadler. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Modeling & Simulation*, 7(2):842–864, 2008.

[4] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. Special Issue: Diffusion Maps and Wavelets.

[5] J De la Porte, BM Herbst, W Hereman, and SJ Van Der Walt. An introduction to diffusion maps. In *Proceedings of the 19th Symposium of the Pattern Recognition Association of South Africa (PRASA 2008), Cape Town, South Africa*, pages 15–25, 2008.

[6] Antonio Gracia, Santiago GonzÃąlez, Victor Robles, and Ernestina Menasalvas. A methodology to compare dimensionality reduction algorithms in terms of loss of quality. *Information Sciences*, 270:1 – 27, 2014.

[7] Laleh Haghverdi, Florian Buettner, and Fabian J. Theis. Diffusion maps for high-dimensional single-cell analysis of differentiation data. *Bioinformatics*, 31(18):2989–2998, 05 2015.

[8] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.

[9] Dan Kushnir, Ali Haddad, and Ronald R. Coifman. Anisotropic diffusion on sub-manifolds with application to earth structure classification. *Applied and Computational Harmonic Analysis*, 32(2):280 – 294, 2012.

[10] Stéphane S Lafon. *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004.

[11] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[12] A. Mead. Review of the development of multidimensional scaling methods. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 41(1):27–39, 1992.

[13] Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis G. Kevrekidis. Diffusion maps - a probabilistic interpretation for spectral embedding and clustering algorithms. In Alexander N. Gorban, Balázs Kégl, Donald C. Wunsch, and Andrei Y. Zinovyev, editors, *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 238–260, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[14] Joseph W. Richards, Peter E. Freeman, Ann B. Lee, and Chad M. Schafer. Accurate parameter estimation for star formation history in galaxies using SDSS spectra. *Monthly Notices of the Royal Astronomical Society*, 399(2):1044–1057, 10 2009.

[15] Mary A. Rohrdanz, Wenwei Zheng, Mauro Maggioni, and Cecilia Clementi. Determination of reaction coordinates via locally scaled diffusion map. *The Journal of Chemical Physics*, 134(12):124116, 2011.

[16] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.

[17] A. Singer. From graph to manifold laplacian: The convergence rate. *Applied and Computational Harmonic Analysis*, 21(1):128 – 134, 2006. Special Issue: Diffusion Maps and Wavelets.

[18] Amit Singer and Ronald R. Coifman. Non-linear independent component analysis with diffusion maps. *Applied and Computational Harmonic Analysis*, 25(2):226 – 239, 2008.

[19] Amit Singer, Radek Erban, Ioannis G. Kevrekidis, and Ronald R. Coifman. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proceedings of the National Academy of Sciences*, 106(38):16090–16095, 2009.

[20] Rui Xu, Steven Damelin, Boaz Nadler, and Donald C. Wunsch. Clustering of high-dimensional gene expression data with feature filtering methods and diffusion maps. *Artificial Intelligence in Medicine*, 48(2):91 – 98, 2010. Artificial Intelligence in Biomedical Engineering and Informatics.

Appendices

Appendix A Decomposition of the transition matrix

Given the dataset $\boldsymbol{X}$ of size n, we consider a weighted graph on this set such that each point in $\boldsymbol{X}$ is a node on the graph. The weights of the graph are characterized by a kernel function $k(.,.)$. Let $\boldsymbol{D}$, defined in (14), be a diagonal matrix consisting the row sums of the weight matrix $\boldsymbol{W}$ where $W_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We construct a Markov chain $\{Y_0, Y_1, ...\}$ on this graph. The transition matrix $\boldsymbol{P}$ can be explained as the row-normalized matrix of $\boldsymbol{W}$ and written as

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}. \tag{28}$$

Note that $\boldsymbol{W}$ is symmetric but $\boldsymbol{P}$ is not. Now consider a matrix $\boldsymbol{P'} = \{P'_{i,j}\}_{1\leq i,j\leq n}$ which is

$$\boldsymbol{P'} = \boldsymbol{D}^{1/2}\boldsymbol{P}\boldsymbol{D}^{-1/2}. \tag{29}$$

We now show that the matrix $\boldsymbol{P'}$ is symmetric.

Since $\boldsymbol{D}$ is diagonal matrix we have $\boldsymbol{D}^{-1} = \boldsymbol{D}^{-1/2}\boldsymbol{D}^{-1/2}$ and $\boldsymbol{D}^{1/2}\boldsymbol{D}^{-1/2} = \boldsymbol{I}_n$. Replace $\boldsymbol{P}$ in (28) into (29) we get

$$\boldsymbol{P'} = \boldsymbol{D}^{1/2}\boldsymbol{D}^{-1}\boldsymbol{W}\boldsymbol{D}^{-1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}.$$

This implies that $\boldsymbol{P'}$ is symmetric since the symmetry of $\boldsymbol{W}$.

As $\boldsymbol{P'}$ is symmetric, there exists a set of eigenvalues $\{\lambda_l\}_{l=0,...,n-1}$ and an orthonormal set of eigenvectors $\{\boldsymbol{\phi}_l\}_{l=0,...,n-1}$ such that

$$\boldsymbol{P'} = \sum_{l=0}^{n-1} \lambda_l \boldsymbol{\phi}_l \boldsymbol{\phi}_l^\top. \tag{30}$$

Additionally, it can be seen that

$$\boldsymbol{P'}\boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1} = \boldsymbol{D}^{1/2}\boldsymbol{P}\boldsymbol{D}^{-1/2}\boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1} = \boldsymbol{D}^{1/2}\boldsymbol{P}\mathbf{1}_{n\times 1} = \boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1}$$

where $\mathbf{1}_{n\times 1}$ is a column vector of size n with all entries 1. The last equality comes from the fact that $\sum_{j=1}^{n} P_{ij}1 = 1$ by the definition of the transition matrix in (15). This shows that $\boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1}$ is an eigenvector of $\boldsymbol{P'}$ that corresponds to the eigenvalue $\lambda_0 = 1$. Consequently, $\boldsymbol{\phi}_0 = \boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1}$.

From (29) and using (30) we can write $\boldsymbol{P}$ as

$$\boldsymbol{P} = \boldsymbol{D}^{-1/2}\boldsymbol{P'}\boldsymbol{D}^{1/2} = \boldsymbol{D}^{-1/2}\left(\sum_{l=0}^{n-1}\lambda_l\boldsymbol{\phi}_l\boldsymbol{\phi}_l^\top\right)\boldsymbol{D}^{1/2} = \sum_{l=0}^{n-1}\lambda_l\boldsymbol{D}^{-1/2}\boldsymbol{\phi}_l\boldsymbol{\phi}_l^\top\boldsymbol{D}^{1/2}$$

By letting

$$\boldsymbol{\psi}_l = \boldsymbol{D}^{-1/2}\boldsymbol{\phi}_l \quad \text{and} \quad \boldsymbol{\varphi}_l = \left(\boldsymbol{\phi}_l^\top\boldsymbol{D}^{1/2}\right)^\top = \boldsymbol{D}^{1/2}\boldsymbol{\phi}_l \tag{31}$$

we get an eigen-decomposition of $\boldsymbol{P}$ as

$$\boldsymbol{P} = \sum_{l=0}^{n-1}\lambda_l\boldsymbol{\psi}_l\boldsymbol{\varphi}_l^\top$$

where $\boldsymbol{\psi}_l$ is the right and $\boldsymbol{\varphi}_l$ is the left eigenvectors. In addition, $\boldsymbol{\phi}_0 = \boldsymbol{D}^{1/2}\mathbf{1}_{n\times 1}$ implies that $\boldsymbol{\psi}_0 = \mathbf{1}_{n\times 1}$ and $\boldsymbol{\varphi}_0 = \boldsymbol{D}1_{n\times 1}$. Moreover, the stationary probability $\boldsymbol{\pi} = \frac{1}{\sum_{i=1}^{n} D_i}\boldsymbol{D}1_{n\times 1}$ is the normalization of vector $\boldsymbol{D}1_{n\times 1}$, that implies $\boldsymbol{\pi}$ is also a left eigenvector corresponding to $\lambda_0 = 1$.

Furthermore, since $\{\boldsymbol{\phi}_l\}_{l=0,...,n-1}$ are orthonormal, $\{\boldsymbol{D}^{-1/2}\boldsymbol{\varphi}_l\}_{l=0,...,n-1}$ are also orthonormal. Therefore, $\{\boldsymbol{\varphi}_l\}_{l=0,...,n-1}$ sets an an orthonormal basis of $l_2(\mathbb{R}^n, \boldsymbol{D}^{-1})$, where the new norm is defined as

$$\|\boldsymbol{x}\|^2_{l_2(\mathbb{R}^n,\boldsymbol{D}^{-1})} = \boldsymbol{x}^\top\boldsymbol{D}^{-1}\boldsymbol{x}, \qquad \boldsymbol{x}\in\mathbb{R}^n.$$

Appendix B Spectrum of the transition matrix

We consider a transition matrix $\boldsymbol{P}$ of size $n \times n$ in which each entry is non-negative, $P_{ij} \geq 0$, and each row sums to 1. As proving above, there is an eigen-decomposition of $\boldsymbol{P}$. It can be checked that

$$\boldsymbol{P}\mathbf{1}_{n\times 1} = \mathbf{1}_{n\times 1}.$$

Therefore, 1 is an eigenvalue of $\boldsymbol{P}$ and $\mathbf{1}_{n\times 1}$ is a right eigenvector corresponding to the eigenvalue 1. Moreover, we will prove that all other eigenvalues of $\boldsymbol{P}$ in absolute value are smaller or equal to 1.

Now, let λ be an arbitrary eigenvalue of $\boldsymbol{P}$ and $\boldsymbol{v}$ is a right eigenvector that satisfy

$$\boldsymbol{P}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

Comparing the ith row of both sides we see that

$$\sum_{j=1}^{n} P_{ij} v_j = \lambda v_i \tag{32}$$

for all $i = 1, ..., n$. Consider v_k is the maximal absolute value entry of $\boldsymbol{v}$ such that

$$|v_k| = \max\{|v_1|, |v_2|, \ldots, |v_n|\}.$$

Note that $|v_k| > 0$ because the eigenvector is non-zero. From (32) we have

$$|\lambda||v_k| = |\lambda v_k| = \left|\sum_{j=1}^{n} P_{kj} v_j\right| \leq \sum_{j=1}^{n} |P_{kj} v_j| \leq \sum_{j=1}^{n} P_{kj}|v_k| = |v_k|,$$

where the first inequality comes from the triangle inequality, the second is because P_{ij} is non-negative, and the last equation is because row of $\boldsymbol{P}$ sum to 1. As a consequence, $|\lambda| \leq 1$.

From that, the spectrum of the transition matrix $\boldsymbol{P}$ of size $n \times n$ satisfies

$$1 = \lambda_0 \geq |\lambda_1| \geq \ldots \geq |\lambda_{n-1}|.$$

Appendix C Diffusion distance in the term of eigenvalues and eigenvectors of Markov matrix

Consider the diffusion distance of $\boldsymbol{x}_i, \boldsymbol{x}_j$ in the dataset $\boldsymbol{X}$ defined in (24) as

$$D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = (\boldsymbol{P}_i^t - \boldsymbol{P}_j^t)\boldsymbol{D}^{-1}(\boldsymbol{P}_i^t - \boldsymbol{P}_j^t)^\top, \tag{33}$$

where $\boldsymbol{P}_i^t = p_t(. \mid \boldsymbol{x}_i)$ is a vector of the probability of moving from $\boldsymbol{x}_i$ to other nodes on the graph in t steps, and $\boldsymbol{D}$ is a diagonal matrix defined in (14).

Additionally, from the eigen-decomposition of the matrix $\boldsymbol{P}^t$ in (20) we have

$$\boldsymbol{P}_i^t = \sum_{l=0}^{n-1} \lambda_l^t \psi_l(\boldsymbol{x}_i)\boldsymbol{\varphi}_l^\top.$$

That gives

$$\boldsymbol{P}_i^t - \boldsymbol{P}_j^t = \sum_{l=0}^{n-1} \lambda_l^t \psi_l(\boldsymbol{x}_i)\boldsymbol{\varphi}_l^\top - \sum_{l=0}^{n-1} \lambda_l^t \psi_l(\boldsymbol{x}_j)\boldsymbol{\varphi}_l^\top = \sum_{l=0}^{n-1} \lambda_l^t (\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))\boldsymbol{\varphi}_l^\top.$$

The distance in (33) then becomes

$$D_t(x_i, x_j)^2 = \left(\sum_{l=0}^{n-1} \lambda_l^t (\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))\boldsymbol{\varphi}_l^\top\right) \boldsymbol{D}^{-1} \left(\sum_{m=0}^{n-1} \lambda_m^t (\psi_m(\boldsymbol{x}_i) - \psi_m(\boldsymbol{x}_j))\boldsymbol{\varphi}_m^\top\right)^\top$$

$$
\begin{aligned}
&= \left(\sum_{l=0}^{n-1} \lambda_l^t(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))\boldsymbol{\varphi}_l^\top \boldsymbol{D}^{-1/2}\right)\left(\sum_{m=0}^{n-1} \lambda_m^t(\psi_m(\boldsymbol{x}_i) - \psi_m(\boldsymbol{x}_j))\boldsymbol{D}^{-1/2}\boldsymbol{\varphi}_m\right) \\
&= \left(\sum_{l=0}^{n-1} \lambda_l^t(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))\boldsymbol{\phi}_l^\top\right)\left(\sum_{m=0}^{n-1} \lambda_m^t(\psi_m(\boldsymbol{x}_i) - \psi_m(\boldsymbol{x}_j))\boldsymbol{\phi}_m\right) \\
&= \sum_{l=0}^{n-1}\sum_{m=0}^{n-1} \lambda_l^t\lambda_m^t(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))(\psi_m(\boldsymbol{x}_i) - \psi_m(\boldsymbol{x}_j))\boldsymbol{\phi}_l^\top\boldsymbol{\phi}_m
\end{aligned}
$$

where the third equation comes from (31). Using the fact that, $\{\boldsymbol{\phi}_l\}_{l=0,\ldots,n-1}$ is an orthonormal eigenvector set of the matrix $\boldsymbol{P'}$ (see Appendix A), that means

$$
\boldsymbol{\phi}_l^\top\boldsymbol{\phi}_m = \delta_{lm} = \begin{cases} 1 & \text{if } l = m \\ 0 & \text{if } l \neq m, \end{cases}
$$

we get

$$
D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=0}^{n-1} \lambda_l^{2t}(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))^2.
$$

However $\boldsymbol{\psi}_0 = \mathbf{1}_{n\times 1}$, the first term in the summation is omitted

$$
D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)^2 = \sum_{l=1}^{n-1} \lambda_l^{2t}(\psi_l(\boldsymbol{x}_i) - \psi_l(\boldsymbol{x}_j))^2.
$$