Prof. Dr. Michael Griebel
Prof. Dr. Jochen Garcke
Dr. Bastian Bohn
Lisa Beer
Jannik Schürg

4

DIFFUSION MAPS

Send your solutions to this chapter's tasks until December 15th.

In the last chapter we introduced PCA as a classic example of a linear dimensionality reduction method. PCA is based on the restrictive assumption, that the data is lying in an affine linear subspace. In contrast to that, nonlinear dimensionality reduction methods consider general manifolds instead, see e.g. [3]. In this chapter, we will deal with *diffusion maps*, which is a nonlinear dimensionality reduction method, introduced by Coifman and Lafon in 2004-2006, e.g. in [1]. In particular, we will apply the new method to real biological data.

4.1 INTRODUCTION

Let $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\} \subset \mathbb{R}^d$ be a given data set. Later on, in the biological applications, n will be the number of cells and d the number of measured genes. We assume that $\mathcal{X}$ is lying on a lower-dimensional manifold $\mathcal{M}$. To reveal the geometry of the data set on this manifold, we define a notion of affinity or similarity between points of $\mathcal{X}$ using a symmetric and positive-semidefinite kernel function $K\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. A common choice for K is the Gaussian kernel

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \tag{4.1}$$

for $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ with bandwidth $\sigma > 0$, which we have already seen on sheet 2.

The main idea of diffusion maps is to construct a random walk Markov chain on the data, where walking to a nearby data point is more likely than walking to another one that is far away. First, we perform a density normalization step by setting

$$q(\mathbf{x}) = \sum_{\mathbf{z} \in \mathcal{X}} K(\mathbf{x}, \mathbf{z})$$

and computing the new kernel

$$K^{(\alpha)}(\mathbf{x}, \mathbf{y}) = \frac{K(\mathbf{x}, \mathbf{y})}{q(\mathbf{x})^\alpha q(\mathbf{y})^\alpha}$$

for some $\alpha \in [0, 1]$. Choosing $\alpha = 1$ provides an embedding which is least affected by the data distribution.

From this, we construct a Markov chain as follows: Set

$$D^{(\alpha)}(\mathbf{x}) = \sum_{\mathbf{z}\in\mathcal{X}} K^{(\alpha)}(\mathbf{x},\mathbf{z})$$

and define the transition matrix

$$P(\mathbf{x},\mathbf{y}) = \frac{K^{(\alpha)}(\mathbf{x},\mathbf{y})}{D^{(\alpha)}(\mathbf{x})}.$$

Now, we have

$$\sum_{\mathbf{z}\in\mathcal{X}} P(\mathbf{x},\mathbf{z}) = 1 \tag{4.2}$$

for all $\mathbf{x}$. This means that the entry $P(\mathbf{x},\mathbf{y})$ can be viewed as the one-step transition probability from $\mathbf{x}$ to $\mathbf{y}$. For a time parameter $t \in \mathbb{N}$, the power P^t gives the t-step transition matrix, i.e. the entry $P^t(\mathbf{x},\mathbf{y})$ represents the transition probability from $\mathbf{x}$ to $\mathbf{y}$ after t time steps. Thus, running the chain forward in time describes the diffusion process of the data $\mathcal{X}$ at various scales.

The Markov chain now allows us to define a time-dependent distance measure on $\mathcal{X}$, the *diffusion distance* D_t by

$$D_t^2(\mathbf{x},\mathbf{y}) := \sum_{\mathbf{z}\in\mathcal{X}} (P^t(\mathbf{x},\mathbf{z}) - P^t(\mathbf{y},\mathbf{z}))^2 \frac{1}{\pi(\mathbf{z})},$$

where π denotes the stationary distribution of the Markov chain.

It is useful to rewrite the diffusion distance D_t by means of a spectral analysis of the Markov chain. Under mild assumptions on K, the transition matrix P has n real eigenvalues $\{\lambda_l\}$ and (right) eigenvectors $\{\psi_l\}$ such that $1 = \lambda_0 > \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1}$ and

$$P\psi_l = \lambda_l \psi_l.$$

The diffusion distance can then be expressed in terms of the eigenvalues and eigenvectors of P:

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

Note that the term for $l = 0$ is omitted because the eigenvector ψ_0 with eigenvalue $\lambda_0 = 1$ is constant.[1] Since the eigenvalues $\{\lambda_l\}$ become smaller and smaller (they tend to zero with bigger n), the diffusion distance can be approximated by the first terms of the sum. For $s < n$, we therefore introduce the family of *diffusion maps* $\{\Psi_t : \mathcal{X} \to \mathbb{R}^s\}_{t\in\mathbb{N}}$ given by

$$\Psi_t(\mathbf{x}) := \begin{pmatrix} \lambda_1^t \psi_1(\mathbf{x}) \\ \lambda_2^t \psi_2(\mathbf{x}) \\ \vdots \\ \lambda_s^t \psi_s(\mathbf{x}) \end{pmatrix}.$$

1 By eq. (4.2) and $\lambda_0 > \lambda_1$ follows that all entries of ψ_0 must be identical.

Each component $\lambda_l^t \psi_l$ is called *diffusion coordinate*. We now can connect the diffusion distance with the diffusion map.

Theorem 4.1. *The diffusion distance D_t is equal to the Euclidean distance in the diffusion map space (up to a relative accuracy depending on s):*

$$D_t(\mathbf{x}, \mathbf{y}) = \|\Psi_t(\mathbf{x}) - \Psi_t(\mathbf{y})\|$$

Thus, the diffusion map Ψ_t is an embedding of the data into the Euclidean space $\mathbb{R}^s$. Note that the k-th entry of the l-th diffusion coordinate, i.e. $\lambda_l^t \psi_l(\mathbf{x}_k)$, is the l-th coordinate of the k-th data point in the embedding space.

The complete diffusion maps algorithm is given below. Note, that for later tasks, one achieves better results, if we set the diagonal of $K^{(\alpha)}$ to zero. In fact, for recovering the structure of the data set, based on relations between data points, it could be disturbing to have nonzero entries on the diagonal of the transition matrix. In this way, the relations between data points are better weighted in the embedding.

Algorithm 4.4 Diffusion maps algorithm

Input: data $\mathcal{X}$, $\alpha \in [0, 1]$, $s < n$
Output: embedded data $\mathcal{Y}$
$K \leftarrow [K(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^n$ with kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
$Q \leftarrow \text{diag}(K\mathbb{1})$ with $\mathbb{1} = (1, \ldots, 1)$
$K^{(\alpha)} \leftarrow Q^{-\alpha} K Q^{-\alpha}$
$K_{i,i}^{(\alpha)} \leftarrow 0$ for all $i = 1, \ldots, n$
$D^{(\alpha)} \leftarrow \text{diag}(K^{(\alpha)}\mathbb{1})$
$P \leftarrow (D^{(\alpha)})^{-1} K^{(\alpha)}$
Compute the first $s + 1$ eigenvalues $\{\lambda_l\}_{l=0}^s$ and the corresponding eigenvectors $\{\psi_l\}_{l=0}^s$ of P
$\mathcal{Y} \leftarrow \{\lambda_l \psi_l\}_{l=1}^s$

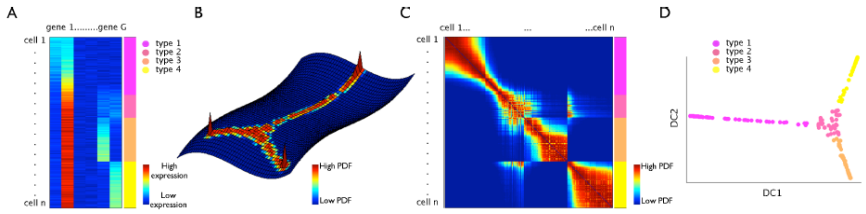

Figure 4.1: Schematic overview of the diffusion maps embedding for single-cell data. (A) The $n \times G$ matrix representation of a single-cell data set consisting of four different cell types. (B) Representation of each cell in the G-dimensional gene space. (C) The $n \times n$ transition matrix. (D) Data embedding on the first two non-trivial eigenvectors of the transition matrix (the diffusion coordinates DC1 and DC2). The embedding shows the flow of cells across the four cell types.

A schematic overview of the diffusion maps embedding, in particular for biological data, is given in fig. 4.1.

Task 4.1. *Implement the diffusion maps algorithm. For this and the next tasks, use the JUPYTER notebook template, which is provided on our website.*

To study the performance of diffusion maps, we start with a data set from [4]. It contains 182 data points, which are subdivided into three different groups. Each data point has a dimension of 8989. For the following tasks, the biological meaning of this data set is not relevant.

Task 4.2. *Embed the data set in a 3-dimensional space by using diffusion maps. Use the kernel in* (4.1) *with parameters* $\sigma = 20$ *and* $\alpha = 1$ *and plot the result in a 3-dimensional scatter plot* ($s = 3$)*, i.e. plot the second, third and fourth eigenvector against each other. Do not forget to label your resulting points in the plot according to the group assignments.*

Task 4.3. *Embed the data set in a 3-dimensional space by using principal component analysis (PCA). You can use your own implementation from the last chapter or from SCIKIT-LEARN. Compare the PCA embedding with the results achieved with diffusion maps in task 4.2.*

4.2 SINGLE-CELL DATA ANALYSIS

In recent years, dimensionality reduction methods have become popular to extract valuable information from high-dimensional biological data. Biologists aim to discover how single cells (e.g. stem cells) differentiate over time and which developmental stages they pass. For this, cell data are collected from different developmental time points and are then united into a single data set. For each cell, gene expression analysis is done by measuring a certain number of genes. However, the high amount of measured genes for each cell often makes it difficult for biologists to detect cell differentiation progressions. Dimensionality reduction methods can help to extract information by embedding the data in a lower-dimensional space. If the embedding space is 2- or 3-dimensional, the data can be visualized. Afterwards, it is possible to discover different cell groups in the data as clusters in the embedding space.

In the following, we will apply diffusion maps to the "Guo" data [5]. The single-cell qPCR data set contains Ct values for 48 genes of 442 mouse embryonic stem cells at seven different developmental time points, from zygote to blastocyst. Starting at the 1-cell stage, cells transition smoothly either towards the trophectoderm (TE) lineage or the inner cell mass (ICM). Subsequently, cells transition from the inner cell mass either towards the primitive endoderm (PE) or the epiblast (EPI) lineage. In table 4.1, you can see how the Excel file for the Guo data looks like. In the first row, you find the names of the measured genes. The naming annotation in the first column refers to the embryonic

stage, embryo number and individual cell number, thus 64C 2.7 refers to the 7th cell harvested from the second embryo collected from the 64-cell stage. However, we are only interested in the embryonic stage of the cells, which is given by the first number (e.g. 64C).

For this, a qPCR (real-time quantitative polymerase chain reaction) is conducted, which consists of several cycles. At each cycle, the amount of fluorescence is measured. A Ct-value (abbreviation for threshold cycle values) is then defined as the cycle number at which the fluorescence significantly exceeds the background-fluorescence, i.e. at which a clear fluorescence signal is first detected. Thus, a higher Ct value means a lower DNA or gene concentration.

4.2.1 *Preprocessing*

To ensure accurate and meaningful analysis, data sets often require preprocessing techniques, such as data cleaning, normalization and handling missing or uncertain values. In the following, we will learn how to preprocess the Guo data. Denote the raw data set as

$$\mathcal{X}_{\text{raw}} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^G,$$

where n is the number of cells and G the number of genes, i.e. $\mathbf{x}_{ij}$ is the expression value of the j-th gene of the i-th cell. For the Guo data, we know the following information:

- cells from the 1-cell stage embryos were treated differently in the experimental procedure and
- entries bigger than the baseline 28 point out undetectable data.

Thus, cells from the 1-cell stage and cells with at least one entry bigger than the baseline have to be excluded from analysis. The resulting cleaned data is given by

$$\mathcal{X} = \mathcal{X}_{\text{raw}} \setminus \Big(\mathcal{X}_{1C} \cup \big\{\mathbf{x} \in \mathcal{X}_{\text{raw}} \,\big|\, \text{exists } j \in \{1, \dots, G\} \text{ s.t. } \mathbf{x}_j > 28\big\}\Big),$$

where $\mathcal{X}_{1C}$ denotes the set of cells from the 1-cell stage.

Afterwards, we need to normalize the data, in order to obtain more accurate results. A common strategy in biology is the normalization via reference genes. In our case, we subtract for each cell the mean

Cell	Actb	Ahcy	Aqp3	⋯	Gapdh	⋯	Tspan8
1C 1	14.01	19.28	23.89	⋯	16.21	⋯	18.53
1C 2	13.68	18.56	28.00	⋯	15.69	⋯	18.29
⋮	⋮	⋮	⋮	⋱	⋮	⋱	⋮
64C 7.14	13.78	25.46	20.79	⋯	17.43	⋯	18.47

Table 4.1: Table of the raw Guo data.

expression of the endogenous control genes Actb and Gapdh apart from the ones with baseline value 28:

$$\mathbf{x}_{ij} \leftarrow \mathbf{x}_{ij} - \frac{1}{2}(\mathbf{x}_{ig_{\text{Actb}}} + \mathbf{x}_{ig_{\text{Gapdh}}}),$$

where $\mathbf{x}_{ij} \neq 28$ and g_{Actb} and g_{Gapdh} are the indices of the genes Actb and Gapdh, respectively. Subsequently, we need to set the entries with baseline 28 to a new threshold. We define this threshold as the smallest integer value greater than or equal to the maximum value of the normalized data set, i.e. $\lceil \max_{i,j}\{\mathbf{x}_{ij} \mid \mathbf{x}_{ij} \neq 28\} \rceil$.

Task 4.4. *Preprocess the Guo data as described above. Round all entries to three decimal places.*

Now, we are able to apply the diffusion maps algorithm to the data set.

Task 4.5. *Perform a diffusion map analysis of the preprocessed Guo data for the kernel in (4.1) with parameters $\sigma = 10$ and $\alpha = 1$ and plot the embedding in a 2-dimensional scatter plot ($s = 2$), i.e. plot the second eigenvector against the third eigenvector. Interpret your result. Can you assign the branches revealed in the plot to the described lineages of the Guo data?*

Task 4.6. *Perform a diffusion map analysis of the Guo data with the same parameters as in task 4.5, but without preprocessing (remove only cells with undetectable data) and compare your result with the plot from task 4.5.*

4.2.2 *Comparison with other dimensionality reduction methods*

We have seen, that preprocessing is an important step in data analysis and from now on, we use the preprocessed Guo data. In the following, we want to compare the diffusion maps performance of the Guo data to other dimensionality reduction methods.

Task 4.7. *Embed the preprocessed Guo data by using principal component analysis (PCA) and another dimensionality reduction method (e.g. tSNE). You can use SCIKIT-LEARN. Compare the results with the diffusion maps embedding from task task 4.5. Compare the computation times of the dimensionality reduction methods, as well.*

4.2.3 *Parameter selection*

Up to now, we have used the bandwidth $\sigma = 10$ for doing the diffusion maps analysis which has given a reasonable result. However, parameter selection is a difficult task in machine learning algorithms.

Task 4.8. *Compare the diffusion maps embedding of the Guo data for several bandwidths σ. Describe the different behaviours.*

We propose a rule for σ, suggested by Lafon [2]:

$$\sigma = \sqrt{\frac{1}{2n} \sum_{i=1}^{n} \min_{j \neq i} \{\|\mathbf{x}_i - \mathbf{x}_j\|^2\}}. \tag{4.3}$$

The radicand indicates the half of the average of all nearest neighbor distances in the data set.

Task 4.9. *Implement the rule (4.3) for the bandwidth σ and plot the embedding for the Guo data set with the bandwidth chosen by this rule.*

4.2.4 *Cell group detection*

So far, we determined the cell groups and lineages by looking at the picture. We aim to identify the cell lineages by a learning method. Since diffusion maps is a spectral embedding method, we can use it to perform spectral clustering on the transition matrix P:

1. For some not too large M, compute the M largest eigenvalues $\{\lambda_l\}_{l=0}^{M}$ of P.
2. Identify Λ such that $\lambda_{\Lambda-1} - \lambda_\Lambda$ is large (spectral gap).
3. Compute the corresponding Λ eigenvectors $\{\psi_i\}_{i=0}^{\Lambda-1}$ of P
4. Extract Λ clusters from $\{\psi_i\}_{i=1}^{\Lambda-1}$ (e.g. with k-means).

Task 4.10. *Implement the spectral clustering algorithm, using k-means for clustering (from SCIKIT-LEARN) for a given number of clusters Λ in step 4.*

Task 4.11. *Plot the first 20 eigenvalues of the transition matrix P for the preprocessed Guo data and identify Λ by determining the biggest gap (use the parameters from task 4.5).*

Task 4.12. *Perform the spectral clustering algorithm for the Guo data with Λ from task 4.11 and plot the resulting points/clusters in 2D. Explain and interpret your result.*

REFERENCES

[1] R.R. Coifman and S. Lafon. "Diffusion Maps." In: *Applied and Computational Harmonic Analysis* 7 (2006), pp. 5–30. DOI: 10.1016/j.acha.2006.04.006.

[2] S. Lafon. "Diffusion maps and geometric harmonics." PhD thesis. Yale University, 2004.

[3] J.A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[4] F. Buettner et al. "Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells." In: *Applied and Computational Harmonic Analysis* 33 (2015), pp. 155–160.

[5] G. Guo et al. "Resolution of cell fate decisions revealed by single-cell gene expression analysis from zygote to blastocyst." In: *Developmental Cell* 18 (2010), pp. 675–685. DOI: 10.1016/j.devcel.2010.02.012.