DIFFUSION MAPS

MARIA CAMERON, SHASHANK SULE

Contents

While the PCA is a power tool when the data points lie near a d-dimensional hyperplane in $\mathbb{R}^D$, it might fail to give a nice embedding if the data are instead located near some d-dimensional curved manifold. To handle this case, Coifman and Lafon (Yale University, 2006) introduced the so-called *diffusion maps* [1]. The key idea of this approach is to devise a discrete-time Markov chain on the data points and define the distances between remote points using the stochastic matrix of this Markov chain. This approach is robust to noisy data and is capable of adequately representing complex geometries of data structures. This dimensional reduction technique has been successfully applied to problems arising in protein dynamics (e.g. [2, 3]). The diffusion map algorithm requires providing a bandwidth parameter ϵ whose choice is nontrivial and has been a subject of active research. One of the first approaches to tackle the problem of choosing ϵ was proposed by A. Little, M. Maggioni, and L. Rosasco. Later, simpler and more robust approaches were proposed by Lindenbaum et al. [4] and T. Berry, J. Harlim, and D. Giannakis [5, 6, 7].

1. Background: properties of stochastic matrices

An $n \times n$ matrix P is called *stochastic* if its entries are nonnegative and its row sums are equal to 1. The entries $P_{i,:}$ can be interpreted as the transition probabilities from state i: p_{ij} is the probability that the system currently at state i will go next to state j.

Definition 1. *We say that a sequence of random variables $(X_k)_{k\geq 0}$, $X_k : \Omega \to S \subset \mathbb{Z}$, is a Markov chain with initial distribution λ and stochastic matrix $P = (p_{ij})_{i,j\in S}$ if*

(1) X_0 *has distribution* $\lambda = \{\lambda_i \mid i \in S\}$ *and*
(2) *the Markov property holds:*

$$\mathbb{P}(X_{k+1} = i_{k+1} \mid X_k = i_k, \dots, X_0 = i_0) = \mathbb{P}(X_{k+1} = i_{k+1} \mid X_k = i_n) = p_{i_k i_{k+1}}.$$

We will write a probability distribution as a row vector. One can show that if λ is the initial probability distribution, then the probabiliby distribution after one step becomes λP, in two steps λP^2, in k steps λP^k, and so on. A probability distribution π is *invariant* if

$$\pi P = \pi \quad \text{and} \quad \sum_{i=1}^{n} \pi_i = 1.$$

If μ is a row vector with n entries such that $\mu P = \mu$, we say that μ is an *invariant probability measure*. Note that μ does not need to sum up to 1[1].

We will limit ourselves to a special kind of Markov chains arising in diffusion maps:

- The number of states is finite: $|S| = n$.
- The stochastic matrix P is irreducible and aperiodic. *Irreducibility* means that any state can be reached from any state in a finite number of jumps, i.e, for any pair i, j, $(P^t)_{ij} > 0$ for some $t \in \mathbb{N}$. *Aperiodicity* means that for any state i and for all sufficiently large t, there is a nonzero probability of returning to i in t steps, i.e, $(P^t)_{ii} > 0$ for all large enogh t and for all i. In this case, one can prove that there exists a *unique invariant probability distribution* π, and for any initial probability distribution λ we have
$$\lim_{k\to\infty} \lambda P^k = \pi.$$
- The Markov chain is time-reversible, or, equivalently, possesses the property of detailed balance: if π is the invariant probability distribution then
$$\pi_i P_{ij} = \pi_j P_{ji},$$
The detailed balance means that, on average, the number of transitions from i to j per time unit is the same as that from j to i.

1.0.1. *Spectral decomposition.* The detailed balance property can be written in the matrix form:

$$\Pi P = P^\top \Pi, \quad \text{where} \quad \Pi := \mathsf{diag}\{\pi_1, \dots, \pi_n\}.$$

Note that the detailed balance condition $\Pi P = P^\top \Pi$ implies that ΠP is symmetric. Indeed, its transpose is $P^\top \Pi$. Hence, the stochastic matrix P is decomposable as

$$P = \Pi^{-1} \tilde{K}, \quad \text{where } \tilde{K} \text{ is symmetric.}$$

Furthermore, P has one eigenvalue equal to 1. The corresponding right eigenvector is $r_0 = [1, \dots, 1]^\top$ (as row sums are all 1), while the corresponding left eigenvector is π, the invariant distribution (as $\pi P = \pi$). All other eigenvalues of P are less than 1 in absolute value. The fact that they do not exceed 1 in absolute value readily follows for Gershgorin

[1]Note that is the set of states is infinite, invariant measure may exist while invariant distribution does not exist. For example, consider a symmetric random walk on $\mathbb{Z}$.

circle theorem saying that the eigenvalues of a matrix A are located within the union of Gershgorin discs $D(a_{ii}, R_i) \subset \mathbb{C}$, where $R_i = \sum_{j \neq i} |a_{ij}|$. Each such disc is centered on the real axis in the interval $[0, 1]$ and has a radius at most 1. The fact that all other eigenvalues are less than 1 in absolute value follows from aperiodicity and irreducibility.

Exercise Prove this.

The detailed balance condition $\Pi P = P^\top \Pi$ implies that P is similar to a symmetric matrix

$$\Pi^{1/2} P \Pi^{-1/2} = \Pi^{-1/2} (\Pi P) \Pi^{-1/2}.$$

Hence all eigenvalues of P are real. Furthermore, let

$$V \Lambda V^\top$$

be the spectral decomposition of $\Pi^{1/2} P \Pi^{-1/2}$. Then

$$V^\top \Pi^{1/2} P \Pi^{-1/2} V = (\Pi^{-1/2} V)^{-1/2} P (\Pi^{-1/2} V) = \Lambda.$$

Hence

$$P = (\Pi^{-1/2} V) \Lambda (\Pi^{-1/2} V)^{-1}$$

is the eigendecomposition of P. Denoting the matrix $\Pi^{-1/2} V$ of right eigenvectors of P by R, we express $V = \Pi^{1/2} R$. Hence, the matrix $L = (\Pi^{-1/2} V)^{-1} = V^\top \Pi^{1/2}$ of left eigenvectors of P is expressed via R and Π as:

$$L = V^\top \Pi^{1/2} = R^\top \Pi.$$

Hence, the eigendecomposition of P is

$$P = R \Lambda R^\top \Pi. \tag{1}$$

Since $RL = LR = I$, we have

$$R^\top \Pi R = I. \tag{2}$$

2. A basic construction of a diffusion map

First, we present the most basic diffusion map algorithm corresponding to $\alpha = 0$ in [1]. This construction is very similar to the construction of Laplacian eigenmap by Belkin and Niyogi (2003) [8].

Let $X = (x_{ik})$ be an $n \times D$ matrix of data. The rows x_i, $i = 1, \ldots, n$, of X represent data points lying in $\mathbb{R}^D$.

- First, we compute the squared-distance matrix between the data points:

$$\Delta(i, j) = \sum_{k=1}^{D} (x_{ik} - x_{jk})^2.$$

- Next, we pick a scaling parameter ϵ and define the diffusion kernel, an $n \times n$ matrix $K = (k_{ij})$ where

$$k_{ij} = \exp\left(-\frac{\Delta(i, j)}{\epsilon} \right).$$

A good choice of ϵ is very important. ϵ should be comparable to squared distances from the data points to their neighbors. In practice, pick a reasonable initial guess for ϵ and then tune it experimentally. One way to pick an initial ϵ is the following. We find row minima among off-diagonal entries for the matrix Δ . Then we find the mean of these minima and set ϵ to be double this mean:

```
for i = 1 : N
    drowmin(i) = min(d(i,setdiff(1:N,i)));
end
epsilon = 2*mean(drowmin);
```

Then, if the result is not satisfactory, keep increasing the factor by which the mean of row minima is multiplied in the last line until the embedding starts making sense. This recipe is good for now. A detailed discussion on choosing ϵ is found in Section 7 below.

- Convert the diffusion kernel K into a stochastic matrix $P = (p_{ij})$ by dividing each row of K by the corresponding row sum:

$$P = Q^{-1}K \quad \text{where} \quad Q := \mathsf{diag}\left\{\sum_{j=1}^{n} k_{1j}, \ldots, \sum_{j=1}^{n} k_{nj}\right\} := \mathrm{diag}\{q_1, \ldots, q_n\}.$$

Indeed, all entries of the resulting matrix P are nonnegative, and its row sums are one.

Note that the diagonal entries of Q constitute an invariant probability measure. Indeed:

$$[q_1, \ldots, q_n]Q^{-1}K = [1, \ldots, 1]K = [q_1, \ldots, q_n]$$

as $K = K^\top$ and both ith row and ith column sum of K is q_i. To obtain the invariant probability distribution, we normalize $[q_1, \ldots, q_n]$ so that it sums up to one:

$$\pi = \frac{q}{\sum_{i=1}^{n} q_i} \quad \text{where} \quad q := [q_1, \ldots, q_n].$$

- Let us take tth power of the matrix P and denote its entries by $p_{ij}^t \equiv (P^t)_{ij}$. The entry p_{ij}^t is the probability to transition from i to j in t steps, $t \in \mathbb{N}$. A family of *diffusion distances* indexed by $t \in \mathbb{N}$ is defined by

$$\boxed{D_t(x_i, x_j)^2 := \sum_{m=1}^{n} \frac{1}{\pi_m} \left|p_{im}^t - p_{jm}^t\right|^2.} \tag{3}$$

Hence, the diffusion distance is a weighted l_2 distance between rows i and j of the matrix P^t. Note that $D_t(x_i, x_j)$ will be small if there is a large number of short paths connecting x_i and x_j, which makes the transition for either of them to any state x_m approximately equally likely. The power t plays the role of a scale parameter. Let us list interesting features of the diffusion distance:

  - Since it reflects the connectivity of the data at a given scale, points are closer if they are highly connected in the graph. Therefore, this distance emphasizes the notion of a cluster.
  - The quantity $D_t(x_i, x_j)$ involves summing over all paths of length t connecting x_i and x_j. This number is very robust to noise perturbation, unlike the geodesic distance.
- The family of *diffusion maps* $\Psi_t : \mathbb{R}^D \to \mathbb{R}^{n-1}$ indexed by $t \in \mathbb{N}$ from the data space $\mathbb{R}^D$ to the diffusion space $\mathbb{R}^{n-1}$ is defined so that the Euclidean distances $\|\Psi_t(x_i) - \Psi_t(x_j)\|$ in the diffusion space are equal to the diffusion distances $D_t(x_i, x_j)$.

  Let
$$P = R\Lambda L \equiv R\Lambda R^\top \Pi$$
be the spectral decomposition of P with ordered eigenvalues:
$$1 = \lambda_0 > |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_{n-1}|.$$
  The diffusion map Ψ_t is defined by:
$$\Psi_t(x_i) := \begin{bmatrix} \lambda_1^t r_1(i) \\ \vdots \\ \lambda_{n-1}^t r_{n-1}(i) \end{bmatrix}, \tag{4}$$
  where $R := [r_0, r_1, \ldots, r_{N-1}]$ is the matrix of right eigenvectors of P normalized so that $R^\top \Pi R = I$. Respectively, λ_m^t, is the tth power of λ_m, $m = 1, \ldots, n-1$. Note that since P is irreducible and aperiodic (as $P_{ii} > 0$, $i = 1, \ldots, n$) by construction, $\lambda_0 = 1$ and $r_0 = [1, \ldots, 1]^\top$. In Eq. (4), $r_m(i)$ denotes the ith entry of the vector r_k. In other words, $\Psi_t(x_i)$ is the transposed ith row of the matrix
$$(R\Lambda^t)^\top \equiv [\lambda_1^t r_1, \lambda_2^t r_2, \ldots, \lambda_{n-1}^t r_{n-1}]^\top.$$
  Note that we remove the first column of R because it consists of all ones and hence is not informative.

Proposition 1.
$$D_t(x_i, x_j)^2 = \sum_{m=1}^{n-1} \lambda_m^{2t} |r_m(i) - r_m(j)|^2, \tag{5}$$
i.e., the diffusion distance in the data space equals the Euclidean distance in the diffusion space.

We will prove this proposition after we finish the description of the construction.

- The diffusion maps allow us to do dimensional reduction by keeping only the first few components of $\Psi_t(\cdot)$. Often it is desirable to keep only the first two or three entries of $\Psi_t(\cdot)$ as then the diffusion map is readily visualizable. To make the dimension of the embedding space justified, we introduce an accuracy parameter $\delta \in (0, 1)$ and define the number of terms to keep:
$$s(\delta, t) = \max\{m \in \mathbb{N} \text{ such that } |\lambda_m|^t > \delta |\lambda_1|^t\}. \tag{6}$$

Then, up to relative precision δ, we have:

$$D_t(x_i, x_j)^2 = \sum_{m=1}^{s(\delta,t)} \lambda_m^{2t} |r_m(i) - r_m(j)|^2, \tag{7}$$

and

$$\Psi_t(x_i) = \begin{bmatrix} \lambda_1^t r_1(i) \\ \vdots \\ \lambda_{s(\delta,t)}^t r_{s(\delta,t)}(i) \end{bmatrix}. \tag{8}$$

This allows us to determine the power t for embedding into $\mathbb{R}^d$ as follows. We pick $\delta \in (0,1)$, for example, $\delta = 0.2$, and then define t so that t is the smallest integer such that

$$\left(\frac{|\lambda_d|}{|\lambda_1|}\right)^t \leq \delta \; \mathbb{R}ightarrow t = \mathsf{ceil}\left[\frac{\log(1/\delta)}{\log\left(|\lambda_1|/|\lambda_d|\right)}\right]. \tag{9}$$

Once the appropriate power for the desired dimension of embedding space (2 or 3) is found, we can define diffusion maps (abusing the term) to 2D or 3D diffusion spaces by

$$\Psi_t(x_i) = \begin{bmatrix} \lambda_1^t r_1(i) \\ \lambda_2^t r_2(i) \end{bmatrix} \quad \text{and} \quad \Psi_t(x_i) = \begin{bmatrix} \lambda_1^t r_1(i) \\ \lambda_2^t r_2(i) \\ \lambda_3^t r_3(i) \end{bmatrix}. \tag{10}$$

Now let us prove Proposition 1.

Proof. Let us redefine the diffusion kernel K as

$$K \to \left(\sum_{i=1}^N q_i\right)^{-1} K.$$

Then the stochastic matrix P with the new K can be decomposed as

$$P = \Pi^{-1} K.$$

P is similar to the symmetric matrix

$$A := \Pi^{1/2} P \Pi^{-1/2} = \Pi^{1/2} \Pi^{-1} K \Pi^{-1/2} = \Pi^{-1/2} K \Pi^{-1/2}.$$

Hence, the eigenvalues of A coincide with those of P. Let

$$A = \Phi \Lambda \Phi^\top = \sum_{k=0}^{n-1} \lambda_k \phi_k \phi_k^\top.$$

be an eigendecomposition of A where Φ is orthogonal, and the diagonal entries of Λ, the eigenvalues, are ordered in the decreasing order. Then the desired eigendecomposition of

P can be obtained as follows:

$$P = \Pi^{-1/2} A \Pi^{1/2} = \Pi^{-1/2} \Phi \Lambda \Phi^\top \Pi^{1/2} =: R \Lambda L = \sum_{k=0}^{n-1} \lambda_k r_k l_k, \tag{11}$$

where $r_k := \Pi^{-1/2}\phi_k$, the columns of R, are the right eigenvectors of P, and $l_k := \phi_k^\top \Pi^{1/2}$, the rows of L, are the left eigenvectors of P. It can be readily verified that the left and right eigenvectors satisfy the following conjugacy relationships:

$$\sum_{m=1}^{n} \pi_m r_i(m) r_j(m) = r_i^\top \Pi r_j = \phi_i^\top \Pi^{-1/2} \Pi \Pi^{-1/2} \phi_j = \phi_i^\top \phi_j = \delta_{i,j}, \tag{12}$$

$$\sum_{m=1}^{n} \frac{l_i(m) l_j(m)}{\pi_m} = l_i \Pi^{-1} l_j^\top = \phi_i^\top \Pi^{1/2} \Pi^{-1} \Pi^{1/2} \phi = \phi_i^\top \phi_j = \delta_{i,j}. \tag{13}$$

Eq. (11) allows us to write entries of P^t as

$$p_{im}^t = \sum_{k=0}^{n-1} \lambda_k^t r_k(i) l_k(m). \tag{14}$$

Plugging p_{im}^t and p_{jm}^t into the definition of $D_t(i,j)$ (equation (3)), we get:

$$\begin{aligned} D_t(x_i, x_j)^2 &= \sum_{m=1}^{n} \frac{1}{\pi_m} \left[\sum_{k=0}^{n-1} \lambda_k^t r_k(i) l_k(m) - \lambda_k^t r_k(j) l_k(m) \right]^2 \\ &= \sum_{m=1}^{n} \frac{1}{\pi_m} \sum_{k=0}^{n-1} \left[\lambda_k^t r_k(i) l_k(m) - \lambda_k^t r_k(j) l_k(m) \right]^2 \\ &+ \sum_{m=1}^{n} \sum_{k=0}^{n-1} \sum_{s \neq k} \frac{l_k(m) l_s(m)}{\pi_m} \lambda_k^t \lambda_s^t [r_k(i) - r_k(j)][r_s(i) - r_s(j)]. \end{aligned}$$

Let us show that the second term in this sum is zero. Rearranging the order of summation and using (13) we get

$$\sum_{k=0}^{n-1} \sum_{s \neq k} \lambda_k^t \lambda_s^t [r_k(i) - r_k(j)][r_s(i) - r_s(j)] \sum_{m=1}^{n} \frac{l_k(m) l_s(m)}{\pi_m} = 0.$$

Returning to the first term, we calculate:

$$\begin{aligned} D_t(x_i, x_j)^2 &= \sum_{m=1}^{n} \frac{1}{\pi_m} \sum_{k=0}^{n-1} \left[\lambda_k^t r_k(i) l_k(m) - \lambda_k^t r_k(j) l_k(m)\right]^2 \\ &= \sum_{m=1}^{n} \sum_{k=0}^{n-1} \frac{[l_k(m)]^2}{\pi_m} \lambda_k^{2t} \left[r_k(i) - r_k(j)\right]^2 \\ &= \sum_{k=0}^{n-1} \lambda_k^{2t} \left[r_k(i) - r_k(j)\right]^2 \sum_{m=1}^{n} \frac{[l_k(m)]^2}{\pi_m} \\ &= \sum_{n=0}^{n-1} \lambda_n^{2t} \left[r_n(i) - r_n(j)\right]^2 . \end{aligned}$$

Finally, we take into account that since $r_0 = [1, \ldots, 1]^\top$, $r_0(i) - r_0(j) = 0$. Therefore,

$$D_t(x_i, x_j) = \sum_{k=1}^{n-1} \lambda_k^{2t} \left[r_k(i) - r_k(j)\right]^2$$

as desired. □

Algorithm 1: Diffusion map with uniform sampling

Input: Data $X \in \mathbb{R}^{n \times D}, x_i \in \mathcal{M} \subset \mathbb{R}^D$, kernel bandwidth $\epsilon > 0$, time parameter $t \geq 0$

Do:

Step 1: Form kernel matrix

$$k_{ij} = \exp\left(-\frac{\Delta(i,j)}{\epsilon}\right)$$

Step 2: Renormalize kernel using invariant distribution:

$$A = Q^{-1/2} K Q^{-1/2} \quad \text{where} \quad Q := \mathsf{diag}\left\{\sum_{j=1}^{n} k_{1j}, \ldots, \sum_{j=1}^{n} k_{nj}\right\} := \mathrm{diag}\{q_1, \ldots, q_n\}.$$

Step 3: Get spectral decomposition of renormalized kernel:

$$A = \Phi \Lambda \Phi^\top$$

Step 4: Construct right eigenvectors of the transition matrix $P = Q^{-1}K$:

$$R = Q^{-1}\Phi$$

Output: Diffusion map at time t: $R\Lambda^t$

3. Relation to Laplacian eigenmap

As we have mentioned, the presented construction of the diffusion map is the most basic one and is very similar to the construction of Laplacian eigenmap [8]. Step 1 of Laplacian eigenmap is the construction of a graph with vertices at the data points which is done in one of the two following ways. Vertices x_i and x_j are connected by an edge

- if $\|x_i - x_j\| < \epsilon$, or
- if x_i is among k nearest neighbors of x_j or x_j is among k nearest neighbors of x_i.

Then, either way, the kernel matrix K is defined by

$$k_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{t}\right), & \|x_i - x_j\| < \epsilon, \\ 0, & \text{otherwise.} \end{cases},$$

where t is interpreted as time for the heat equation on the manifold occupied with the data. Note that $t = \infty$ corresponds to $k_{ij} = 1$ if x_i and x_j are connected and zero otherwise, i.e., K is merely the adjacency matrix.

Next, we set up the matrix called the *graph Laplacian*:

$$L := Q - K, \quad \text{where} \quad Q = \mathsf{diag}\left\{\sum_j k_{1j}, \ldots, \sum_j k_{nj}\right\}. \tag{15}$$

Then the following generalized eigenvalue problem is solved:

$$LR = QRM, \quad M = \mathsf{diag}\{\mu_0, \mu_1, \ldots, \mu_{n-1}\}, \tag{16}$$

where

$$0 = \mu_0 \leq \mu_1 \leq \ldots \leq \mu_{n-1}. \tag{17}$$

Note that the matrix of the right eigenvectors R coincides with that of P. The matrix M relates to Λ via

$$\Lambda = I - M. \tag{18}$$

Finally, the *Laplacian eigenmap* to $\mathbb{R}^m$, $m \leq n$, is defined by

$$x_i \mapsto (r_1(i), \ldots, r_m(i)). \tag{19}$$

Exercise Show that the Laplacian eigenmap to $\mathbb{R}^m$ is the solution to the following optimization problem:

$$\sum_{i,j} k_{ij}\|y_i - y_j\|_2^2 = \mathsf{tr}\left(Y^\top L Y\right) \to \min \quad \text{subject to} \quad Y^\top Q Y = I, \quad Y^\top Q 1_{n\times 1} = 0. \tag{20}$$

Here, y_i's are columns of Y, and Y is $n \times m$.

Remark It is shown in [8] that LLE and Laplacian eigenmap are closely related. The minimization problem for LLE involves graph Laplacian squared.

4. Illustrative examples

4.0.1. *Swiss Roll.* First we make an approximately uniform mesh of points on the Swiss Roll as shown in Fig. 1(a). The number of points is $n = 1060$. We set $\delta = 0.2$ and find the values for ϵ and t as described above:

$$\epsilon = 0.7717928, \quad t = 147.$$

The points are sorted and colored according to the approximate geodesic distance to the data point closest to the origin. The matrix P^t is displayed in Fig. 1(b). The absolute eigenvalues of P^t starting from $|\lambda_1|$ are shown in Fig. 1(c). The embedding to 3D is in Fig. 1(d). The Swiss Roll has been mostly unrolled.

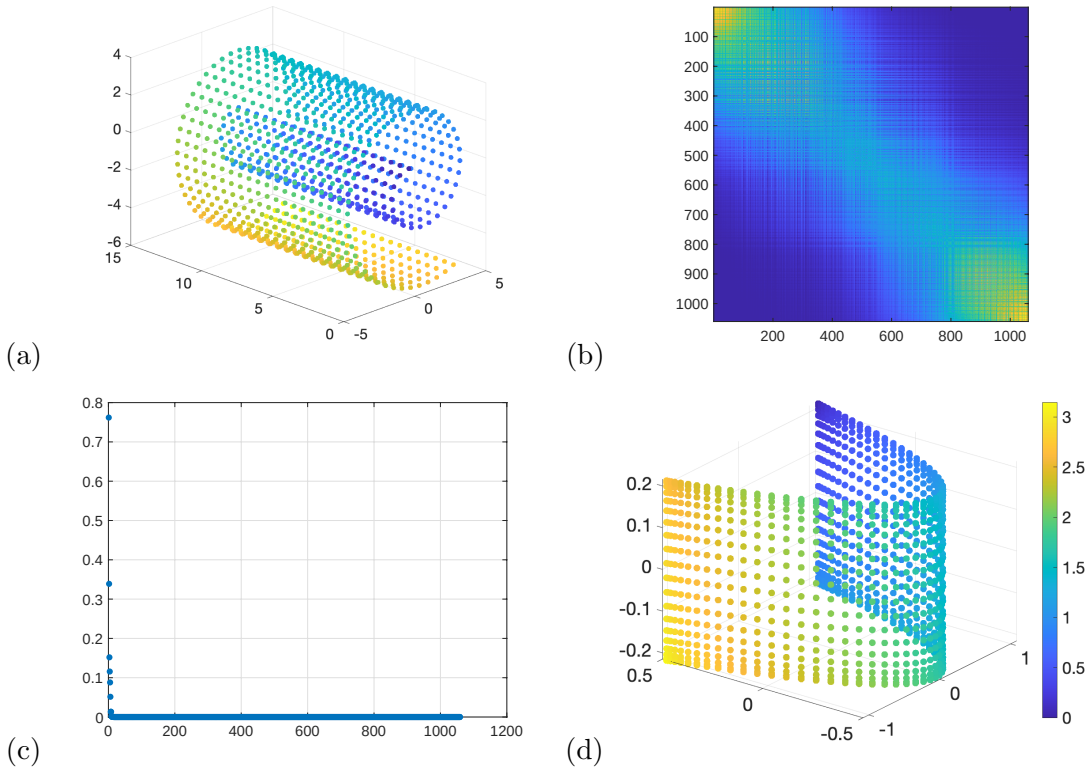


Figure 1. (a): The Swiss Roll dataset with points arranged into a quasi-uniform mesh. (b): The matrix P^t for $\epsilon = 0.7717928$, $t = 147$. (c): The absolute eigenvalues of the eigenvalues of P^t starting from $|\lambda_1|$. (d): Diffusion map to 3D.

Next, we repeat this experiment by adding noise to the data:

```
noisestd = 0.4;
X = X + noisestd*randn(size(X)); % perturb by Gaussian noise
```

Setting $\delta = 0.2$ as before, we find:

$$\epsilon = 0.6108029, \quad t = 300.$$

The results are shown in Fig. 2.

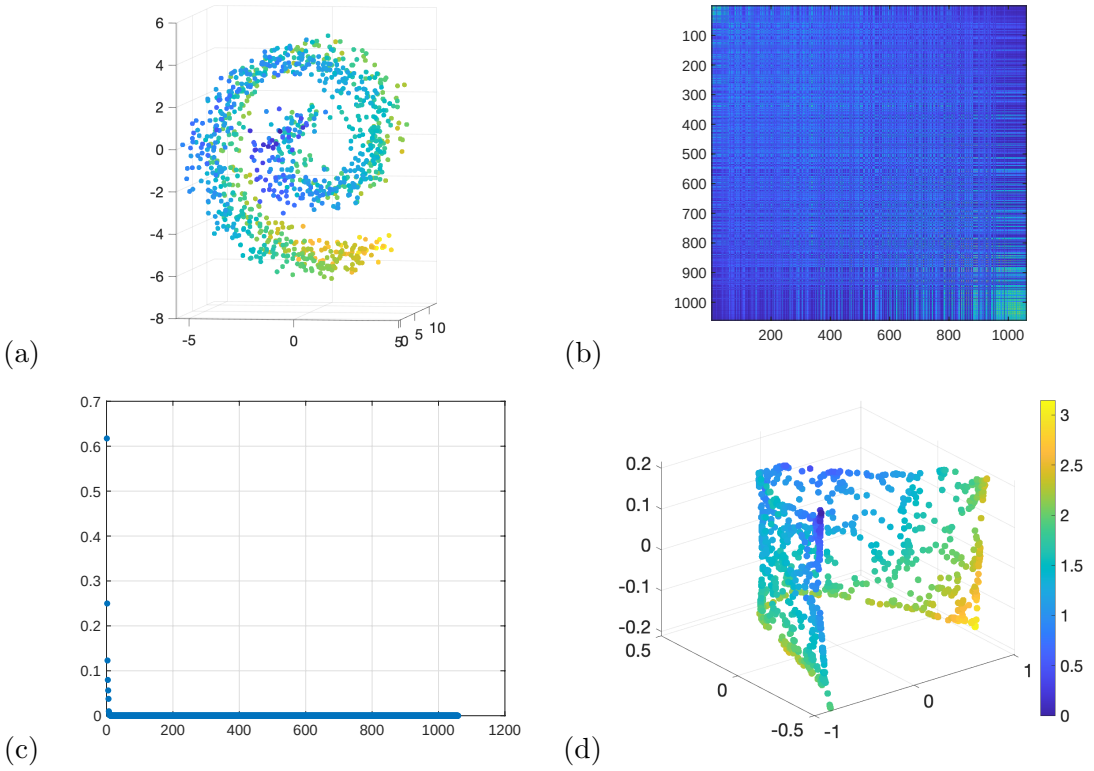


FIGURE 2. (a): The Swiss Roll dataset with points arranged into a quasi-uniform mesh and perturbed by Gaussian noise with standard deviation 0.4. (b): The matrix P^t for $\epsilon = 0.6108029$, $t = 300$. (c): The absolute eigenvalues of the eigenvalues of P^t starting from $|\lambda_1|$. (d): Diffusion map to 3D.

Finally, we take the same Swiss Roll data that we used for the isomap experiments with Gaussian noise of standard deviation 0.8. With $\delta = 0.2$, we found

$$\epsilon = 2.104531, \quad t = 1705.$$

The results are shown in Fig. 3.

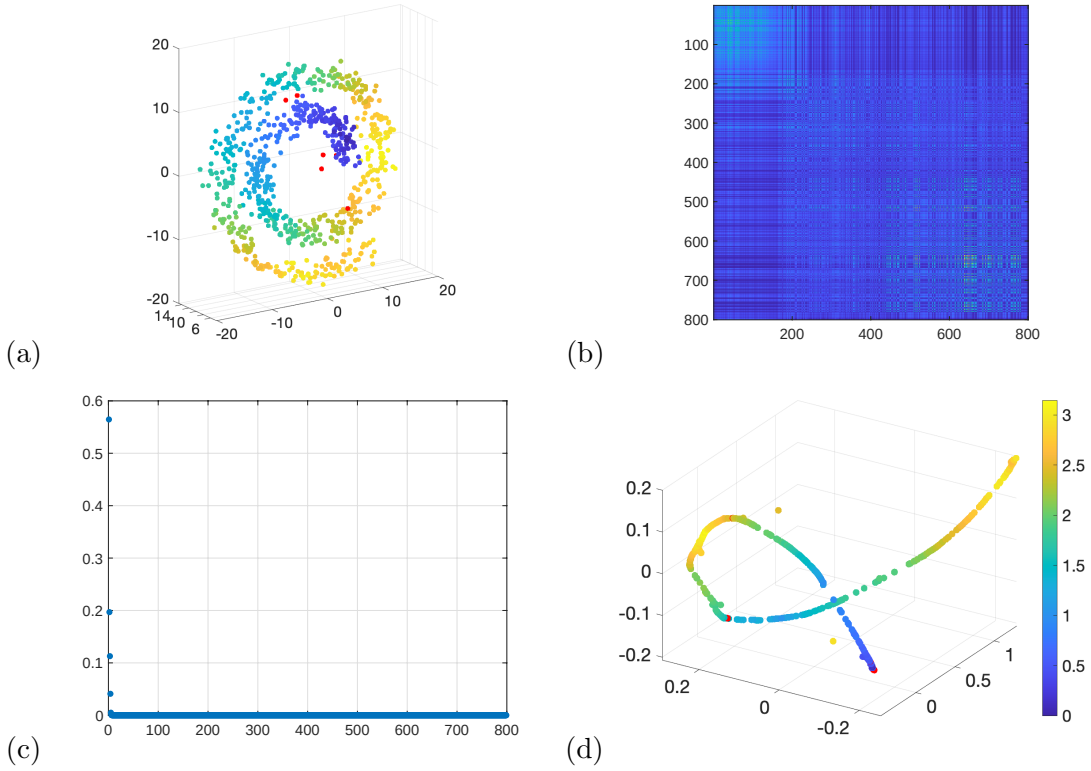


FIGURE 3. (a): The Swiss Roll dataset used for the experiments with isomap perturbed by Gaussian noise with standard deviation 0.8. (b): The matrix P^t for $\epsilon = 2.104531$, $t = 1705$. (c): The absolute eigenvalues of the eigenvalues of P^t starting from $|\lambda_1|$. (d): Diffusion map to 3D.

4.0.2. *Pacman.* Let us consider a data set consisting of 200 images depicting the Pacman. This example is similar to the one in this article[1]. Each image is 65×65 pixels either black (color = 0) or white (color = 255). The images differ from each other by the angle of rotation of the Pacman around the center of the image. The angles of rotation are

$$\alpha_i = \frac{2\pi i}{200}.$$

A sample of 20 such images is shown in Fig. 4(a). This dataset is naturally embedded into $\mathbb{R}^{65^2} = \mathbb{R}^{4225}$ space. Note that $D > N$ in this case. The PCA mapping into 3D applied to this dataset is shown in Fig. 4(b). The absolute eigenvalues and the embedding into 3D

[1]While this article offers a nice exposition, I do not recommend to rely on it as it contains a number of errors in important formulas. For example, Eqs. (6) and (7) contain errors, the comments following Eq. (9) are misleading, etc.

are shown in Figs. 4(c) and (d) respectively. Both the PCA and the diffusion map show that the set of images is well-approximated by a 1D manifold as we would expect.

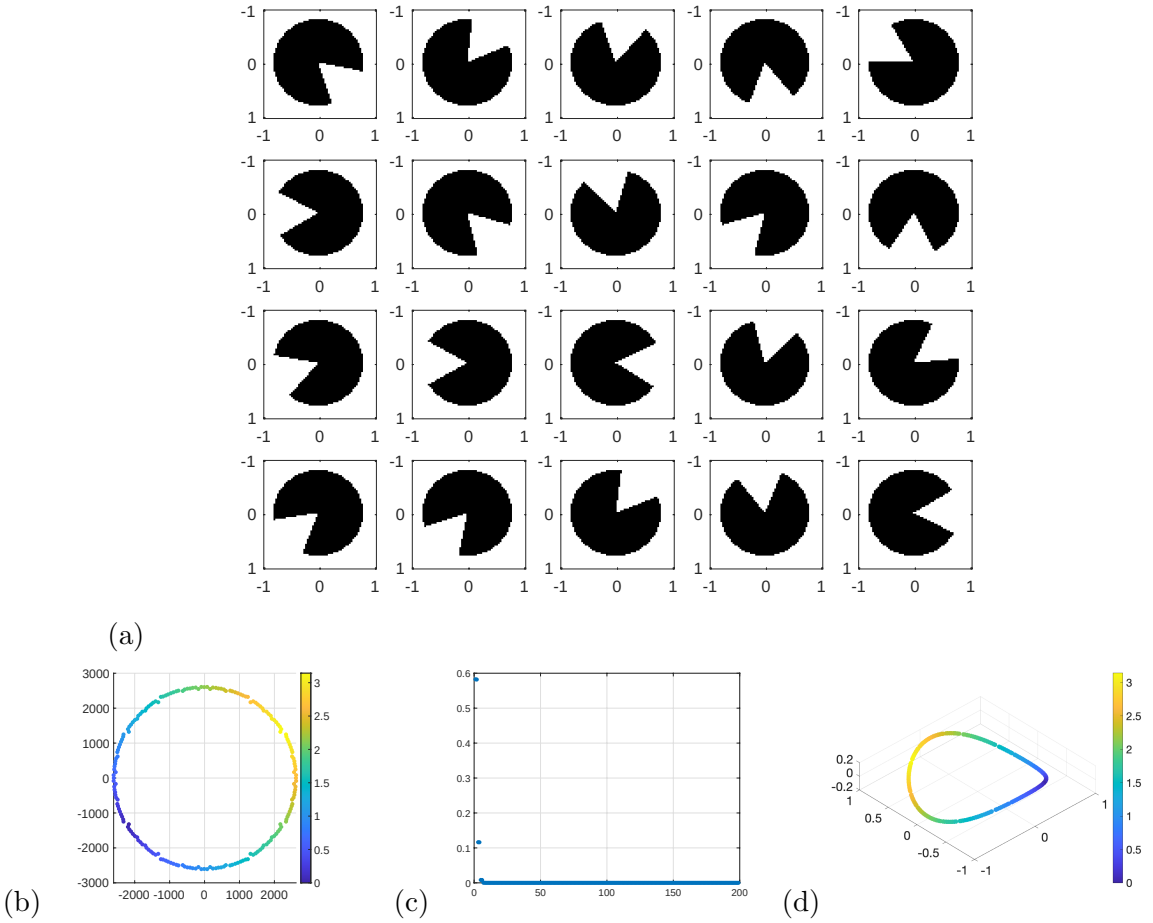
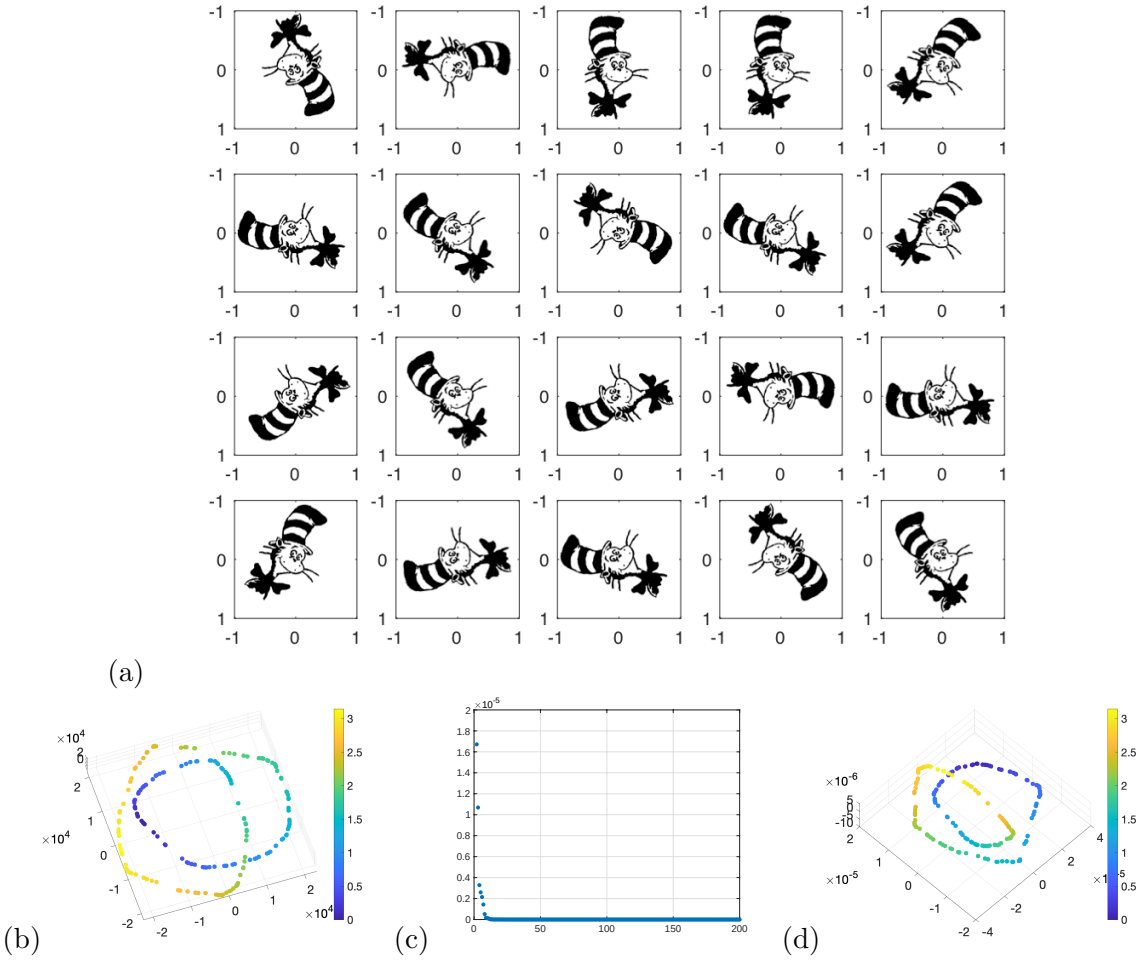


FIGURE 4. The dataset consists of 200 images of the Pacman rotated around the center of the image by angles $\alpha_i = 2\pi i/200$. (a): A sample of 20 data points. (b): The PCA mapping into 3D. (c): The absolute eigenvalues of the eigenvalues of P^t starting from $|\lambda_1|$. Here: $\delta = 0.5$, $\epsilon = 2335698$, $t = 187$. (d): Diffusion map to 3D.

4.0.3. *Cat-in-the-hat.* A similar example with a more complex image of the Cat-in-the-hat is shown in Fig. 5. Each image is 500×500. The double-loop formed by the mapped data is caused by the fact that the image rotated by π is closer to the original image than those rotated by an angle between $\pi/6$ and $5\pi/6$.



(a) (b) (c) (d)

FIGURE 5. The dataset consists of 200 images of the Cat-in-the-hat rotated around the center of the image by angles uniformly distributed in $(0, 2\pi)$. (a): A sample of 20 data points. (b): The PCA mapping into 2D. (c): The absolute eigenvalues of the eigenvalues of P^t starting from $|\lambda_1|$. Here: $\delta = 0.2$, $\epsilon = 7.318159 \cdot 10^7$, $t = t$. (d): Diffusion map to 3D.

5. The continuous counterpart of the diffusion map algorithm

Reference for this section: [9] arXiv:2208.13772. The basic diffusion map algorithm presented in Section 2 leaves us with two questions. What is a good choice of the bandwidth parameter ϵ? What is a good choice of power t? The first question will be answered in Section 7 below. The second question will become irrelevant as we will renormalize the kernel and eliminate the need for taking power t altogether. To understand to answers to

these questions, we will consider the continuous counterpart of the diffusion map algorithm also described in the paper by Coifman and Lafon (2006).

Why the diffusion map algorithm contains the word diffusion in its name? What is the underlying diffusion process? Looking at the basic construction we can think that the dynamics of the Markov chain with the constructed stochastic matrix P is, perhaps, a diffusion process. In fact, it is indeed a diffusion process discretized to a point cloud. To see it, let us start with a diffusion equation in $\mathbb{R}^d$:

$$u_t = \frac{1}{4}\Delta u, \quad x \in \mathbb{R}^d, \quad u(x,0) = f(x), \tag{21}$$

where $\Delta u = u_{x_1x_1} + u_{x_2x_2} + \ldots + u_{x_dx_d}$ is Laplace's operator applied to u. The solution to (21) at time $t = \epsilon$ given by

$$u(x,\epsilon) = \frac{1}{(\pi\epsilon)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{\epsilon}} f(x')dx'. \tag{22}$$

On the other hand, if ϵ is small, $u(x,\epsilon)$ can be found using a Taylor expansion:

$$u(x,\epsilon) = u(x,0) + \frac{\partial}{\partial t}u(x,0)\epsilon + O(\epsilon^2) = f(x) + \frac{\epsilon}{4}\Delta f(x) + O(\epsilon^2). \tag{23}$$

Matching (22) and (23), we obtain the following Taylor expansion for the integral operator with the Gaussian kernel $k_\epsilon(x,x') := \exp(-||x-x'||^2/\epsilon)$:

$$\frac{1}{(\pi\epsilon)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{\epsilon}} f(x')dx' = f(x) + \frac{\epsilon}{4}\Delta f(x) + O(\epsilon^2). \tag{24}$$

Now let us connect this integral operator with the matrix operators K and P constructed in Section 2. We observe that the normalizing factor $(\pi\epsilon)^{d/2}$ is obtained by integrating the kernel with respect to its second argument over $\mathbb{R}^d$:

$$(\pi\epsilon)^{d/2} = \int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{\epsilon}} dx'. \tag{25}$$

Let X be an $n \times d$ data matrix whose rows $x_i^\top \in \mathbb{R}^d$ are the data points forming a point cloud. Let f be a smooth function with compact support. We discretize f to the point cloud $\{x_i\}_{i=1}^n$ by setting $f_i = f(x_i)$, $1 \le i \le n$, $[f] = [f_1, \ldots, f_n]^\top$, and compute $P[f]$. We have:

$$[P[f]]_i = \frac{[P[f]]_i}{[K1_{n\times 1}]_i} = \frac{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}} f(x_j)}{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}}}.$$

Taking limit as $n \to \infty$ we get:

$$\lim_{n\to\infty} \frac{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}} f(x_j)}{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}}} = \frac{\int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{\epsilon}} f(x')\rho(x)dx'}{\int_{\mathbb{R}^d} e^{-\frac{||x-x'||^2}{\epsilon}} \rho(x)dx'},$$

where $\rho(x)$ is the probability density function that the point cloud is sampled from. Using the Taylor expansion (24) we calculate:

$$\lim_{n\to\infty} \frac{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}} f(x_j)}{\sum_j e^{-\frac{||x_i-x_j||^2}{\epsilon}}} = \frac{\int_{\mathbb{R}^d} e^{-\frac{||x_i-x'||^2}{\epsilon}} f(x')\rho(x)dx'}{\int_{\mathbb{R}^d} e^{-\frac{||x_i-x'||^2}{\epsilon}} \rho(x)dx'} \tag{26}$$

$$= \frac{(f\rho)(x_i) + \frac{\epsilon}{4}\Delta(f\rho)(x_i) + O(\epsilon^2)}{\rho(x_i) + \frac{\epsilon}{4}\Delta\rho(x_i) + O(\epsilon^2)}$$

$$= f(x_i) + \frac{\epsilon}{4}\left[\Delta f(x_i) + 2\nabla f(x_i)\cdot\frac{\nabla\rho(x_i)}{\rho(x_i)}\right] + O(\epsilon^2). \tag{27}$$

To obtain the last equality we used

$$\frac{1}{\rho + \frac{\epsilon}{4}\Delta\rho(x) + O(\epsilon^2)} = \frac{1}{\rho}\left(1 - \frac{\epsilon}{4}\frac{\Delta\rho}{\rho} + O(\epsilon^2)\right).$$

Therefore, subtracting $[f]$ from $P[f]$ and dividing the result by ϵ we obtain a point-wise approximation to the action of the differential operator

$$\frac{1}{4}\left[\Delta + 2\frac{\nabla\rho\cdot\nabla}{\rho}\right] \equiv \frac{1}{4}\left[\Delta + \nabla\log\rho^2\cdot\nabla\right] \tag{28}$$

on the function f:

$$\mathcal{L}_\epsilon f(x_i) := \lim_{\epsilon\to 0}\lim_{n\to\infty} \frac{[P[f]]_i - f_i}{\epsilon} = \frac{1}{4}\left[\Delta f(x_i) + \nabla\log\rho^2\cdot\nabla f(x_i)\right] \tag{29}$$

The operator $\frac{1}{\epsilon}(P-I)$ in the left-hand side of (29) is the generator of the Markov chain constructed in Section 2 for the discrete time steps of size ϵ. As we see, the generator approximates not the Laplacian but the differential operator (28). If the sampling density ρ were uniform in some region Ω, this operator would be proportional to Laplace's operator in Ω.

6. Removing the effect of nonuniform sampling

Usually, the sampling density of data points is nonuniform. In this case, it is advantageous to modulate the effect of nonuniform density by the right renormalization of the kernel function originally developed by Coifman and Lafon [1] and then simplified in later works [10, 11]. We define a family of right-renormalized kernels by

$$k_{\epsilon,\alpha}(x_i, x_j) = e^{-\frac{\|x_i-x_j\|^2}{\epsilon}} \rho_\epsilon^{-\alpha}(x_j), \tag{30}$$

where $\rho_\epsilon(x')$ is the estimate for the sampling density at x'. Note that typically the sampling density is not known but can be estimated using the fact that the Gaussian kernel with a proper normalization approximates the Dirac δ-function:

$$(\pi\epsilon)^{-d/2} e^{-\frac{\|x-x'\|}{\epsilon}} \approx \delta(x-x').$$

Indeed, it is easy to check using Taylor expansion in ϵ that

$$(\pi\epsilon)^{-d/2}\int_{\mathbb{R}^d} e^{-\frac{\|x-x'\|^2}{\epsilon}}\rho(x')dx' = \rho(x) + O(\epsilon). \tag{31}$$

Motivated by this, we define the following density estimate:

$$\rho_\epsilon(x_i) = (\pi\epsilon)^{-d/2}\frac{1}{n}\sum_j e^{-\frac{\|x_i-x_j\|^2}{\epsilon}}. \tag{32}$$

If the kernel $e^{-\frac{||x-x_j||^2}{\epsilon}}$ is replaced with the right-normalized kernel

$$e^{-\frac{||x-x_j||^2}{\epsilon}}\rho_\epsilon^{-\alpha}(x_j), \tag{33}$$

a calculation of a limit similar to the one in the left-hand side of (26) results in the following family of differential operators

$$\mathcal{L}_{\epsilon,\alpha} := \frac{1}{4}\left[\Delta f + \nabla f\cdot\nabla\left[\log\rho^{2(1-\alpha)}\right]\right], \tag{34}$$

For $\alpha = 1$, the resulting operator is the Laplacian. Another case of interest is $\alpha = 0.5$. In this case, the generator $\mathcal{L}_{\epsilon,0.5}$ is the generator for the overdamped Langevin dynamics.

Let us summarize the diffusion map algorithm with $\alpha \in [0,1]$ in Algorithm 2.

Algorithm 2: Diffusion map with nonuniform sampling

Input: Data X sampled i.i.d through non-uniform density ρ, kernel bandwidth ϵ, modulation parameter $\alpha \in [0,1]$

Do:

- **Step 1.** Set a rotation-invariant kernel
$$k_\epsilon(x,y) = \exp\left[-\|x-y\|^2/\epsilon\right].$$
To make the kernel matrix sparse, define
$$[K_\epsilon]_{ij} = \begin{cases} \exp\left[-\|x_i-x_j\|^2/\epsilon\right], & \|x_i-x_j\| < 3\sqrt{0.5\epsilon}, \\ 0, & \text{otherwise.} \end{cases}$$
- **Step 2.** Calculate row sums $q_\epsilon(x) = \sum_y k_\epsilon(x,y)$ and form the new kernel
$$k_\epsilon^{(\alpha)}(x_i,x_j) = \frac{k_\epsilon(x_i,x_j)}{q_\epsilon^\alpha(x_j)}, \quad \text{or} \quad K_{\epsilon,\alpha} = K_\epsilon Q^{-\alpha}. \tag{35}$$
- **Step 3.** Calculate row sums
$$d_\epsilon^{(\alpha)}(x_i) = \sum_j k_\epsilon^{(\alpha)}(x_i,x_j)$$
and define the stochastic matrix
$$P_{\epsilon,\alpha} = \left[D_\epsilon^{(\alpha)}\right]^{-1} K_{\epsilon,\alpha}, \tag{36}$$
where
$$D_\epsilon^{(\alpha)} = \mathsf{diag}\left\{d_\epsilon^{(\alpha)}(x_1),\ldots,d_\epsilon^{(\alpha)}(x_n)\right\}.$$
- **Step 4**: Finally, we construct the Laplacian
$$L_{\epsilon,\alpha} = \frac{1}{\epsilon}(P_{\epsilon,\alpha} - I).$$
- **Step 4**: Diagonalize $L_{\epsilon,\alpha}$:
$$L_{\epsilon,\alpha} = R\Lambda R^{-1} \iff P_{\epsilon,\alpha} = R(I+\epsilon\Lambda)R^{-1}$$
Note $P_{\epsilon,\alpha}$ and $L_{\epsilon,\alpha}$ get diagonalized by the same eigenbasis. The eigenvalues will differ, but we choose to remove their influence by setting $t = 0$:

Output: Diffusion map: $D_\alpha^{-1/2}R$

Note that the construction is the same the uniformly sampled diffusion map algorithm except that the formula for the embeddings to 2D or 3D are

$$\Psi_\alpha(x_i) = \begin{bmatrix} r_1(i) \\ r_2(i) \end{bmatrix}, \quad \Psi_\alpha(x_i) = \begin{bmatrix} r_1(i) \\ r_2(i) \\ r_3(i) \end{bmatrix}. \tag{37}$$

Note that in the above equation the parameter t has been chosen to be 0.

Remark The above algorithm was presented in class as diffusion map.

Theorem 1 ([12]). *Let the number of points $n \to 0$ and assume the data x_i has been sampled i.i.d from sampling density ρ. Then*

$$L_{\epsilon,\alpha} \to \frac{1}{4}\left(\Delta f + (2-2\alpha)\frac{\nabla\rho}{\rho}\cdot\nabla f\right) + O(\epsilon) \tag{38}$$

Thus, $4L_{\epsilon,\alpha}$ is a numerical approximation for $\mathcal{L}_\alpha := \Delta f + (2-2\alpha)\frac{\nabla\rho}{\rho}\cdot\nabla f$

7. Choosing ϵ

Reference for this section: [9] arXiv:2208.13772. In practice, the limit $\epsilon \to 0$ cannot be taken for a finite dataset. Instead, one generally tries to choose ϵ as small as possible without making the corresponding generator matrix $L_{\epsilon,\mu}$ reducible. Many heuristics exist for choosing the scaling parameter ϵ in diffusion maps, relating back to bandwidth selection in kernel density estimation [4]. Here, we present the method of Berry, Harlim and Giannakis [5, 6, 7, 13] which we refer to as the "Ksum test".

The idea for the heuristic is to find the range of ϵ where the asymptotic results of diffusion maps hold true for the given dataset. We find this range by analyzing the double sum

$$S(\epsilon) := \frac{1}{N^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[K_\epsilon]_{ij}$$

over a range of ϵ values. Here, $[K_\epsilon]_{ij} = k_\epsilon(x_i, x_j)$ where $k_\epsilon(x,x') = \exp(-\|x-x'\|^2/\epsilon)$ is the Gaussian kernel. We assume that the point cloud is located in a finite region $\Omega \subset \mathbb{R}^d$. For large n, the intermediate asymptotic for $S(\epsilon)$ is [12, 6]

$$S(\epsilon) \approx \int_\Omega dx \int_\Omega dx' k_\epsilon(x,x') \approx \pi^{d/2}\epsilon^{d/2}\mathsf{vol}(\Omega) \equiv C\epsilon^{d/2}$$

where C is a constant independent of ϵ. Hence,

$$\log S(\epsilon) \approx \frac{d}{2}\log\epsilon + \log C, \tag{39}$$

i.e., $\log S$ is a linear function of $\log\epsilon$ if ϵ is not too large and not too small.

On the other hand, if ϵ is large, $[K_\epsilon]_{ij} \approx 1$ for all i,j, and hence $S(\epsilon) \to 1$ as $\epsilon \to \infty$. For small ϵ, $[K_\epsilon]_{ij} \approx 0$ for all i,j, $i \neq j$, and $[K_\epsilon]_{ii} = 1$. Therefore, $S(\epsilon) \to N^{-1}$ as $\epsilon \to 0$. Therefore, if we plot $\log\epsilon$ against $\log S(\epsilon)$, we expect to see a linear region of slope approximately $\frac{d}{2}$, where d is the dimension of the dataset.This region demarcates the range of suitable values of ϵ [5, 6, 7, 13]. On the other hand, the slope of this graph should tend to zero as $\epsilon \to 0$ and as $\epsilon \to \infty$.

In particular, we expect to have $\frac{\partial\log S(\epsilon)}{\partial\log\epsilon} \approx \frac{d}{2}$ where the slope is maximal. For a practical calculation, it is useful to note that the slope is given by

$$\frac{\partial\log S(\epsilon)}{\partial\log\epsilon} = -\frac{\sum_{i,j=1}^{N}[K_\epsilon]_{ij}\log[K_\epsilon]_{ij}}{\sum_{i,j=1}^{N}[K_\epsilon]_{ij}}. \tag{40}$$

8. Solving PDEs with diffusion maps

Recall that the d-dimensional diffusion map with parameter α is given by

$$\Psi_\alpha(x_i) := \begin{bmatrix} r_1(i) \\ \vdots \\ r_{n-1}(i) \end{bmatrix}, \tag{4}$$

Here r_j are the *right* eigenvectors of the transition matrix $P_{\epsilon,\alpha}$. Equivalently, they are the right eigenvectors of the generator matrix $L_{\epsilon,\alpha}$. Thus, computing the diffusion map amounts to solving

$$L_{\epsilon,\alpha} R = R\Lambda \tag{41}$$

In [12] it was shown that if the points x_i are drawn from a compact manifold $\mathcal{M}$ (such as a sphere or a torus) as $n \to \infty$ and $\epsilon \to 0$, the solutions r_i to the above problem approximate the Neumann boundary value problem:

$$\mathcal{L}_\alpha f(x) = \lambda f(x), x \in \mathcal{M} \quad \partial_n f(x) = 0, x \in \partial\mathcal{M} \tag{42}$$

This means that $L_{\epsilon,\alpha}$ is useful for another purpose: numerically solving elliptic boundary value problems on $\mathcal{M}$! A particularly familiar boundary value problem is the committor problem:

$$\mathcal{L}q(x) := \beta^{-1}\Delta q - \nabla V \cdot \nabla q = 0, x \in \Omega \setminus (A \cup B), \quad q(\partial A) = 0, \quad q(\partial B) = 1 \tag{43}$$

In fact, we can show, using (38) that when $\alpha = 1/2$, $\mathcal{L} = \beta^{-1}\mathcal{L}_{1/2}$. Since $\mathcal{L}_{1/2} \approx 4L_{\epsilon,\alpha}$, we have that $\mathcal{L} \approx 4\beta^{-1}L_{\epsilon,\alpha}$. Thus if $x_i \sim Z^{-1}\exp(-\beta V)$, the committor problem can now be discretized to

$$L_{\epsilon,\alpha}[q]_i = 0, x_i \in \Omega \setminus (A \cup B), \quad [q]_i = 0, x_i \in A, \quad [q]_i = 1, x_i \in B \tag{44}$$

This is a linear system with boundary constraints. The incorporation of these constraints into the linear system is detailed in Section 2.1 of [14].

References

[1] R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, pp. 5–30, 2006.

[2] A. L. Ferguson, A. Z. Panagiotopoulosa, P. G. Debenedettia, and K. I. G., "Systematic determination of order parameters for chain dynamics using diffusion maps," *PNAS*, vol. 107, no. 31, pp. 13597–13602, 2010.

[3] S. B. Kim, C. J. Dsilva, I. G. Kevrekidis, and P. G. Debenedetti, "Systematic characterization of protein folding pathways using diffusion maps: Application to trp-cage miniprotein," *J. Chem. Phys*, vol. 142, p. 085101, 2019.

[4] O. Lindenbaum, M. Salhov, A. Yeredor, and A. Averbuch, "Gaussian bandwidth selection for manifold learning and classification," *Data mining and knowledge discovery*, vol. 34, no. 6, pp. 1676–1712, 2020.

[5] T. Berry, D. Giannakis, and J. Harlim, "Nonparametric forecasting of low-dimensional dynamical systems," *Physical Review E*, vol. 91, no. 3, p. 032915, 2015.

[6] T. Berry and J. Harlim, "Variable bandwidth diffusion kernels," *Applied and Computational Harmonic Analysis*, vol. 40, no. 1, pp. 68–96, 2016.

[7] D. Giannakis, "Data-driven spectral decomposition and forecasting of ergodic dynamical systems," *Applied and Computational Harmonic Analysis*, vol. 47, no. 2, pp. 338–396, 2019.

[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput*, vol. 13, pp. 1373–1397, 2003.

[9] A. L. Evans, M. K. Cameron, and P. Tiwary, "Computing committors via mahalanobis diffusion maps with enhanced sampling data," *Journal of Chemical Physics*, 2022. arXiv:2208.13772.

[10] R. Banisch, Z. Trstanova, A. Bittracher, S. Klus, and P. Koltai, "Diffusion maps tailored to arbitrary non-degenerate itô processes," *Applied and Computational Harmonic Analysis*, vol. 48, no. 1, pp. 242–265, 2020.

[11] Z. Trstanova, B. Leimkuhler, and T. Lelièvre, "Local and global perspectives on diffusion maps in the analysis of molecular systems," *Proceedings of the Royal Society A*, vol. 476, no. 2233, p. 20190036, 2020.

[12] 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*, vol. 7, no. 2, pp. 842–864, 2008.

[13] A. D. Davis and D. Giannakis, "Graph-theoretic algorithms for kolmogorov operators: Approximating solutions and their gradients in elliptic and parabolic problems on manifolds," *arXiv preprint arXiv:2104.15124*, 2021.

[14] R. Lai and J. Lu, "Point cloud discretization of fokker–planck operators for committor functions," *Multiscale Modeling & Simulation*, vol. 16, no. 2, pp. 710–726, 2018.