Diffusion Maps: A manifold learning algorithm

John Harlim
Department of Mathematics
Department of Meteorology & Atmospheric Science
Institute of CyberScience.
The Pennsylvania State University

March 18, 2019

Plan of the talk:

- Principal Component Analysis
- Diffusion Maps
- Variable Bandwidth Diffusion Kernels
- Automatic estimation of manifold dimension and bandwidth parameter.

What is manifold learning?

Given data $\{x_i\}_{i=1,\ldots,N}$, the central task of unsupervised learning algorithm is to be able to characterize this data set.

Under an assumption that these data lie on (or close to) a manifold $\mathcal{M} \subseteq \mathbb{R}^n$, manifold learning algorithm seeks for a set of (basis) functions, $\Phi_k : \mathcal{M} \to \mathbb{R}$ to describe the data.

Principal Component Analysis (a linear manifold learning)

Given $x_i \in \mathbb{R}^n$ with zero empirical mean, define

$$X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{n \times N}.$$

Let (λ_k, w_k) be defined as,

$$\frac{1}{N} X X^\top w_k = \lambda_k w_k$$

The kth principal component is defined as $\Phi_k(x) = w_k^\top x$.

Principal Component Analysis (a linear manifold learning)

Example: Uniformly distributed data on a unit circle.

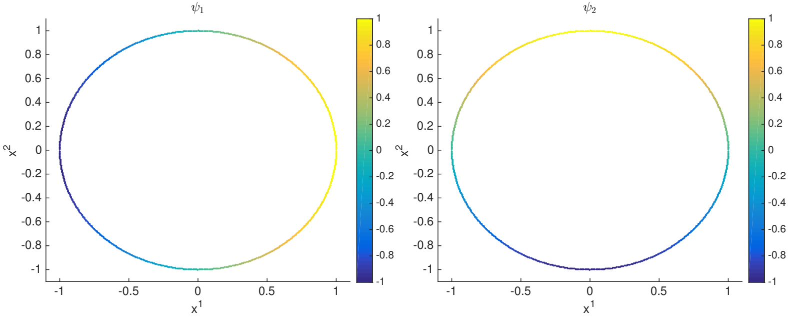


Figure : The principal components (color) as functions of the data.

Principal Component Analysis (a linear manifold learning)

Example: Gaussian invariant density of a two-dimensional SDE's

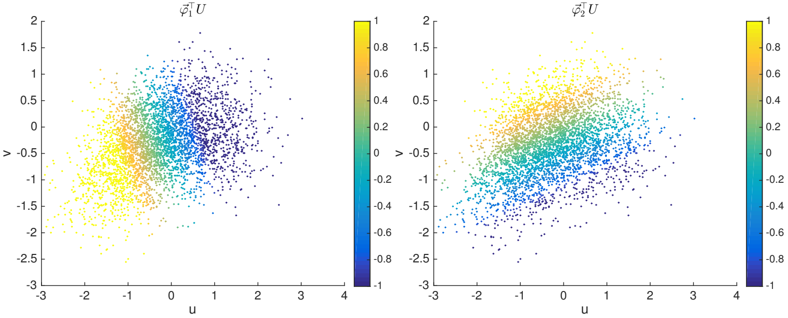


Figure : Principal components of the Gaussian data.

Diffusion maps (a nonlinear manifold learning)[1]

Given $\{x_i\} \in \mathcal{M} \subseteq \mathbb{R}^n$ with a sampling density q, the **diffusion maps** algorithm is a kernel based method that produces orthonormal basis functions $\varphi_k \in L^2(\mathcal{M}, q)$.

[1] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

Diffusion maps (a nonlinear manifold learning)[1]

Given $\{x_i\} \in \mathcal{M} \subseteq \mathbb{R}^n$ with a sampling density q, the **diffusion maps** algorithm is a kernel based method that produces orthonormal basis functions $\varphi_k \in L^2(\mathcal{M}, q)$.

These basis functions are solutions of an eigenvalue problem,

$$\mathcal{L}\varphi_k(x) = q^{-1}\text{div}\Big(q\nabla\varphi_k(x)\Big) = \lambda_k\varphi_k(x),$$

with Neumann BC (if the manifold has a boundary).

[1] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

Diffusion maps (a nonlinear manifold learning)[1]

Given $\{x_i\} \in \mathcal{M} \subseteq \mathbb{R}^n$ with a sampling density q, the **diffusion maps** algorithm is a kernel based method that produces orthonormal basis functions $\varphi_k \in L^2(\mathcal{M}, q)$.

These basis functions are solutions of an eigenvalue problem,

$$\mathcal{L}\varphi_k(x) = q^{-1}\text{div}\Big(q\nabla\varphi_k(x)\Big) = \lambda_k\varphi_k(x),$$

with Neumann BC (if the manifold has a boundary).

Remarks:

- If $q = 1$, then $\mathcal{L} = \Delta$.
- Diffusion maps approximates $\mathcal{L}$ with an exponentially decaying function function $K_\epsilon(x, y) = h\big(\frac{\|x-y\|^2}{4\epsilon}\big)$.

[1] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

A review on diffusion maps algorithm

The key idea of diffusion maps stimulated by the following asymptotic expansion[2]. For $x \in \mathcal{M} \subseteq \mathbb{R}^n$ away from the boundary and $f \in C^3(\mathcal{M})$

$$
\begin{aligned}
G_\epsilon f(x) &:= \epsilon^{-d/2} \int_{\mathcal{M}} K_\epsilon(x, y) f(y) dV(y) \\
&= m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2).
\end{aligned}
$$

where $m_0 = \int_{\mathbb{R}^d} h(||z||^2)dz$ and $m_2 = \frac{1}{2}\int_{\mathbb{R}^d} y_1^2 h(||z||^2)dz$ are constants determined by h, and ω depends on the induced geometry of $\mathcal{M}$.

[2] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

A review on diffusion maps algorithm

The key idea of diffusion maps stimulated by the following asymptotic expansion[2]. For $x \in \mathcal{M} \subseteq \mathbb{R}^n$ away from the boundary and $f \in C^3(\mathcal{M})$

$$\begin{aligned} G_\epsilon f(x) &:= \epsilon^{-d/2} \int_{\mathcal{M}} K_\epsilon(x,y) f(y) dV(y) \\ &= m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2). \end{aligned}$$

where $m_0 = \int_{\mathbb{R}^d} h(||z||^2)dz$ and $m_2 = \frac{1}{2}\int_{\mathbb{R}^d} y_1^2 h(||z||^2)dz$ are constants determined by h, and ω depends on the induced geometry of $\mathcal{M}$.

Diffusion maps is a discretization of the following algebraic manipulation:

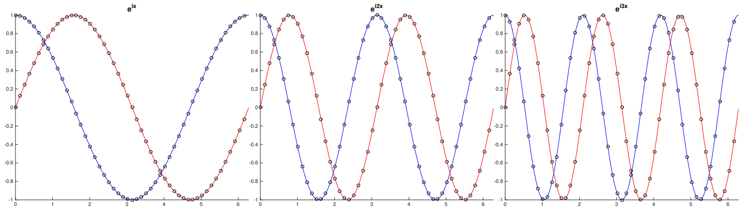
$$L_\epsilon f(x) := \frac{1}{\epsilon m_2 m_0^{-1}} (G_\epsilon 1(x))^{-1} G_\epsilon f(x) - f(x) = \Delta f(x) + \mathcal{O}(\epsilon)$$

[2] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

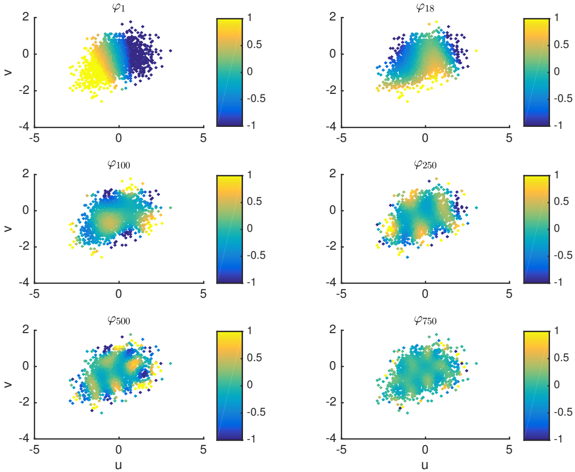
Examples: Uniformly distributed data on a circle

Analytically, DM solves $\Delta\varphi_k(x) = \lambda_k\varphi_k(x)$, which solutions are:

$$\lambda_k = -k^2, \qquad \varphi_k(x) = e^{\mathrm{i}kx}.$$



Example: Gaussian invariant density of a two-dimensional SDE's



Essentially, we view the DM as a method to construct generalized Fourier basis on the manifold.

Diffusion Maps Algorithm

Using this asymptotic expansion,

$$G_\epsilon f(x) = m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2),$$

given data $x_i \in \mathcal{M}$ with sampling density $q(x)$, we can estimate the Laplacian through the following procedure:

Diffusion Maps Algorithm

Using this asymptotic expansion,

$$G_\epsilon f(x) = m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2),$$

given data $x_i \in \mathcal{M}$ with sampling density $q(x)$, we can estimate the Laplacian through the following procedure:

- Compute $q_\epsilon = G_\epsilon(q)$.

Diffusion Maps Algorithm

Using this asymptotic expansion,

$$G_\epsilon f(x) = m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2),$$

given data $x_i \in \mathcal{M}$ with sampling density $q(x)$, we can estimate the Laplacian through the following procedure:

- Compute $q_\epsilon = G_\epsilon(q)$.
- Compute $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$ for some parameter α.

$$\hat{G}_{\epsilon,\alpha,q}(f) = m_0^{1-\alpha} f q^{1-\alpha}\left(1 + \epsilon m\omega(1-\alpha) - \epsilon m\alpha\frac{\Delta q}{q} + \epsilon m\frac{\Delta(fq^{1-\alpha})}{fq^{1-\alpha}} + \mathcal{O}(\epsilon^2)\right),$$

where $m = m_2/m_0$.

Diffusion Maps Algorithm

Using this asymptotic expansion,

$$G_\epsilon f(x) = m_0 f(x) + \epsilon m_2(\omega(x) f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2),$$

given data $x_i \in \mathcal{M}$ with sampling density $q(x)$, we can estimate the Laplacian through the following procedure:

- Compute $q_\epsilon = G_\epsilon(q)$.
- Compute $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$ for some parameter α.

$$\hat{G}_{\epsilon,\alpha,q}(f) = m_0^{1-\alpha} f q^{1-\alpha}\left(1 + \epsilon m\omega(1-\alpha) - \epsilon m\alpha\frac{\Delta q}{q} + \epsilon m\frac{\Delta(fq^{1-\alpha})}{fq^{1-\alpha}} + \mathcal{O}(\epsilon^2)\right),$$

  where $m = m_2/m_0$.
- Compute $\hat{q}_\epsilon := \hat{G}_{\epsilon,\alpha,q}(1)$.

Diffusion Maps Algorithm

Using this asymptotic expansion,

$$G_\epsilon f(x) = m_0 f(x) + \epsilon m_2(\omega(x)f(x) + \Delta f(x)) + \mathcal{O}(\epsilon^2),$$

given data $x_i \in \mathcal{M}$ with sampling density $q(x)$, we can estimate the Laplacian through the following procedure:

- Compute $q_\epsilon = G_\epsilon(q)$.
- Compute $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$ for some parameter α.

$$\hat{G}_{\epsilon,\alpha,q}(f) = m_0^{1-\alpha} fq^{1-\alpha}\left(1 + \epsilon m\omega(1-\alpha) - \epsilon m\alpha\frac{\Delta q}{q} + \epsilon m\frac{\Delta(fq^{1-\alpha})}{fq^{1-\alpha}} + \mathcal{O}(\epsilon^2)\right),$$

  where $m = m_2/m_0$.
- Compute $\hat{q}_\epsilon := \hat{G}_{\epsilon,\alpha,q}(1)$.
- Finally,

$$\mathcal{L}_{\epsilon,\alpha} f := \frac{\hat{q}_\epsilon^{-1}\hat{G}_{\epsilon,\alpha,q}(f) - f}{m\epsilon} = (2-2\alpha)\nabla\log q \cdot \nabla f + \Delta f + \mathcal{O}(\epsilon).$$

Diffusion Maps Algorithm

Numerically, we can repeat this procedure as follows.
Given $\{x_i\}_{i=1}^N \sim q(x)$ that lie on $\mathcal{M} \in \mathbb{R}^n$, choose a Gaussian kernel,

$$K_\epsilon(x, y) = \exp\Big(-\frac{\|x - y\|^2}{4\epsilon}\Big),$$

such that $m = m_2/m_0 = 1$.

We can approximate the operator $G_\epsilon f$ as a discrete sum,

$$\begin{aligned}\epsilon^{d/2} G_\epsilon(fq)(x) &= \int_{\mathcal{M}} K_\epsilon(x, y) f(y) q(y) dV(y) \\ &= \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} K_\epsilon(x, x_i) f(x_i).\end{aligned}$$

Diffusion Maps Algorithm

With the Monte-Carlo discretization, the numerical approximation of operator $\mathcal{L}_{\epsilon,\alpha}$ is given as:

Diffusion Maps Algorithm

With the Monte-Carlo discretization, the numerical approximation of operator $\mathcal{L}_{\epsilon,\alpha}$ is given as:

- Compute $q_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} K_\epsilon(x_i, x_j)$.

Diffusion Maps Algorithm

With the Monte-Carlo discretization, the numerical approximation of operator $\mathcal{L}_{\epsilon,\alpha}$ is given as:

- Compute $q_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} K_\epsilon(x_i, x_j)$.
- Construct the kernel of $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$:

$$\hat{K}_\epsilon(x_i, x_j) = \frac{K_\epsilon(x_i, x_j)}{q_\epsilon(x_i)^\alpha q_\epsilon(x_j)^\alpha}.$$

  This is called “right” normalization

Diffusion Maps Algorithm

With the Monte-Carlo discretization, the numerical approximation of operator $\mathcal{L}_{\epsilon,\alpha}$ is given as:

- Compute $q_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} K_\epsilon(x_i, x_j)$.
- Construct the kernel of $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$:

$$\hat{K}_\epsilon(x_i, x_j) = \frac{K_\epsilon(x_i, x_j)}{q_\epsilon(x_i)^\alpha q_\epsilon(x_j)^\alpha}.$$

  This is called "right" normalization
- Compute $\hat{q}_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} \hat{K}_\epsilon(x_i, x_j)$.

Diffusion Maps Algorithm

With the Monte-Carlo discretization, the numerical approximation of operator $\mathcal{L}_{\epsilon,\alpha}$ is given as:

- Compute $q_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} K_\epsilon(x_i, x_j)$.
- Construct the kernel of $\hat{G}_{\epsilon,\alpha,q}(f) := G_\epsilon\left(\frac{fq}{q_\epsilon^\alpha}\right)$:

$$\hat{K}_\epsilon(x_i, x_j) \quad = \quad \frac{K_\epsilon(x_i, x_j)}{q_\epsilon(x_i)^\alpha q_\epsilon(x_j)^\alpha}.$$

  This is called "right" normalization

- Compute $\hat{q}_\epsilon(x_i) = \frac{1}{N}\sum_{j=1}^{N} \hat{K}_\epsilon(x_i, x_j)$.
- Then matrix representation of $\mathcal{L}_{\epsilon,\alpha}$ is given as,

$$\left[L_{\epsilon,\alpha}\right]_{i,j} = \frac{1}{\epsilon}\left(\frac{\hat{K}_\epsilon(x_i, x_j)}{\hat{q}_\epsilon(x_i)} - \delta_{i,j}\right).$$

  The first term on the RHS is called "left" normalization.

Remarks:

Recall that $\mathcal{L}_{\epsilon,\alpha} f = (2-2\alpha)\nabla \log q \cdot \nabla f + \Delta f + \mathcal{O}(\epsilon)$.

- If $\alpha = 0$ and $q(x) = 1/Vol(\mathcal{M})$ is uniform, then we approximate the Laplace-Beltrami on $\mathcal{M}$; this is the "Laplacian eigenmaps" introduced by Belkin and Niyogi 2003.
- If $\alpha = 1$, we also get Laplace-Beltrami on $\mathcal{M}$ even if the sampling measure is non-uniform.
- If $\alpha = 1/2$, we approximate,

$$\mathcal{L}_{\epsilon,1/2} = \nabla \log q \cdot \nabla + \Delta + \mathcal{O}(\epsilon) = q^{-1}\mathrm{div}\Big(q\nabla \quad\Big) + \mathcal{O}(\epsilon),$$

which is the generator of a gradient system with an *isotropic* diffusion:

$$dx = -\nabla U(x)dt + \sqrt{2}\, dW_t,$$

where $x \in \mathcal{M}$ and the equilibrium measure is $q(x) = e^{-U(x)}$.

Remarks:

For the estimation of Δ, the eigenfunctions φ_k form an orthonormal basis of $L^2(\mathcal{M})$ correspond to eigenvalues $\lambda_k \geq 0$.

Definition

Let $S_\epsilon(x, y) = e^{\epsilon \Delta \delta_y(x)}$ be the heat kernel of Δ. The diffusion distance is defined as,

$$D_\epsilon(x, y)^2 := \|S_\epsilon(x, \cdot) - S_\epsilon(y, \cdot)\|^2_{L^2(\mathcal{M})}.$$

Representing the heat kernel with the basis functions, we have

$$D_\epsilon(x, y)^2 = \sum_{k=1}^{\infty} e^{2\lambda_k \epsilon} \big(\varphi_k(x) - \varphi_k(y)\big)^2.$$

Remarks:

Diffusion Maps[3] is defined as a map, $\Phi_{\epsilon,M} : \mathcal{M} \to \mathbb{R}^M$, as

$$\Phi_{\epsilon,M}(x) := (e^{\lambda_1 \epsilon}\varphi_1(x), \dots, e^{\lambda_M \epsilon}\varphi_M(x)).$$

Then for appropriate choices of ϵ and M, the map $\Phi_{\epsilon,M}$ is an isometric embedding, in the sense of:

$$D_\epsilon(x, y)^2 \approx \|\Phi_{\epsilon,M}(x) - \Phi_{\epsilon,M}(y)\|_{\mathbb{R}^M}$$

preserving the diffusion distance.

[3] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

Remarks:

Diffusion Maps[3] is defined as a map, $\Phi_{\epsilon,M} : \mathcal{M} \to \mathbb{R}^M$, as

$$\Phi_{\epsilon,M}(x) := (e^{\lambda_1 \epsilon}\varphi_1(x), \dots, e^{\lambda_M \epsilon}\varphi_M(x)).$$

Then for appropriate choices of ϵ and M, the map $\Phi_{\epsilon,M}$ is an isometric embedding, in the sense of:

$$D_\epsilon(x,y)^2 \approx \|\Phi_{\epsilon,M}(x) - \Phi_{\epsilon,M}(y)\|_{\mathbb{R}^M}$$

preserving the diffusion distance.

Compare to PCA, $\Phi_k(x) = w_k^\top x$.

[3] Coifman & Lafon, Appl. Comp. Harmon. Anal. 2006

Restriction on compact manifold

Consider estimating generator of Ornstein-Uhlenbeck process on a line $\mathcal{M} = \mathbb{R}$, which is a gradient flow with potential $U(x) = x^2/2$.

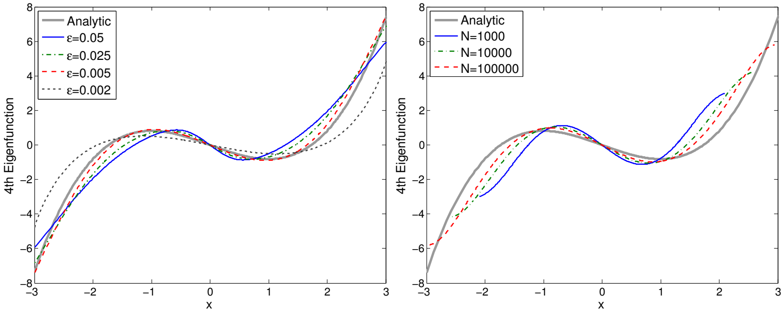


Figure : Left: Estimation of the third eigenfunction of the generator of the OU process with 2000 data points. Right: Various number of data points where $\sqrt{N}$ outliers are removed.

Variable bandwidth diffusion kernels[4]

- We consider variable bandwidth diffusion kernels for data lie on non-compact domain without boundary of the following form,

$$K_\epsilon^S(x,y) = \exp\Big(-\frac{\|x-y\|^2}{4\epsilon\rho(x)\rho(y)}\Big).$$

- If we choose $\rho(x) = q(x)^\beta + \mathcal{O}(\epsilon)$ and $\beta = -1/2$, and apply DM with $\alpha = -d/4$, where $d = dim(\mathcal{M})$, then we can approximate the generator $\mathcal{L}_{\epsilon,1/2}$ that takes functions on $L^2(\mathcal{M}, q) \cap C^3(\mathcal{M})$.

[4] Berry and H, Appl. Comput. Harmon. Anal. 2016.

Back to the OU example

With the variable bandwidth kernel.

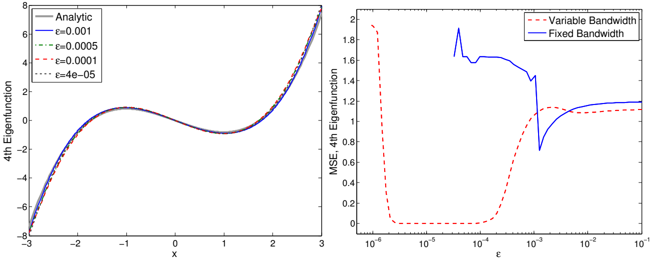


Figure : Left: VB estimation of the fourth eigenfunction of the generator of the OU process with 2000 data points. Right: The mean squared error between the analytic fourth eigenfunction and the kernel based approximations as a function of ϵ.

Variable Bandwidth Diffusion Kernels [5]

Given data $x_i \sim q(x)$,

$$K_\epsilon^S(x_i, x_j) = \exp\left\{\frac{-||x_i - x_j||^2}{4\epsilon\rho(x_i)\rho(x_j)}\right\} \qquad q_\epsilon^S(x_i) = \sum_{j=1}^{N} \frac{K_\epsilon(x_i, x_j)}{\rho(x_i)^d}$$

$$K_{\epsilon,\alpha}^S(x_i, x_j) = \frac{K_\epsilon^S(x_i, x_j)}{q_\epsilon^S(x_i)^\alpha q_\epsilon^S(x_j)^\alpha} \qquad q_{\epsilon,\alpha}^S(x_i) = \sum_{j=1}^{N} K_{\epsilon,\alpha}^S(x_i, x_j)$$

$$\hat{K}_{\epsilon,\alpha}^S(x_i, x_j) = \frac{K_{\epsilon,\alpha}^S(x_i, x_j)}{q_{\epsilon,\alpha}^S(x_i)} \qquad L_{\epsilon,\alpha}^S(x_i, x_j) = \frac{\hat{K}_{\epsilon,\alpha}^S(x_i, x_j) - \delta_{ij}}{\epsilon\rho(x_i)^2},$$

We proved that for each x,

$$L_{\epsilon,\alpha}^S f(x) \to \Delta f(x) + 2(1-\alpha)\nabla f(x) \cdot \frac{\nabla q(x)}{q(x)} + (d+2)\nabla f(x) \cdot \frac{\nabla \rho(x)}{\rho(x)}$$

in probability.

---

[5] Berry and H, Appl. Comput. Harmon. Anal. 2016

Variable Bandwidth Diffusion Kernels [6]

Choosing $\rho = q^{\beta} + \mathcal{O}(\epsilon)$, we have at each x_i,

$$L^{S}_{\epsilon,\alpha} f(x_i) = \Delta f(x_i) + c_1 \nabla f(x_i) \cdot \frac{\nabla q(x_i)}{q(x_i)} + \mathcal{O}\left(\epsilon, \frac{q(x_i)^{(1-d\beta)/2}}{\sqrt{N}\epsilon^{2+d/4}}, \frac{||\nabla f(x_i)|| q(x_i)^{-c_2}}{\sqrt{N}\epsilon^{1/2+d/4}}\right),$$

with $c_1 = 2 - 2\alpha + d\beta + 2\beta$ and $c_2 = 1/2 - 2\alpha + 2d\alpha + d\beta/2 + \beta$.

Remarks: A natural choice for $\beta = -1/2$.

- For gradient flow, we want $c_1 = 1$ and $\alpha = -d/4$. In this case, $c_2 = d/2(1/2 - d) < 0$ for $d > 0$.
- In contrast, the fixed bandwidth with $\beta = 0$, we have $\alpha = 1/2$ and $c_2 = d - 1/2 > 0$ for $d > 0$.

[6] Berry and H, Appl. Comput. Harmon. Anal. 2016

Automatic estimation of ϵ and d

Note that

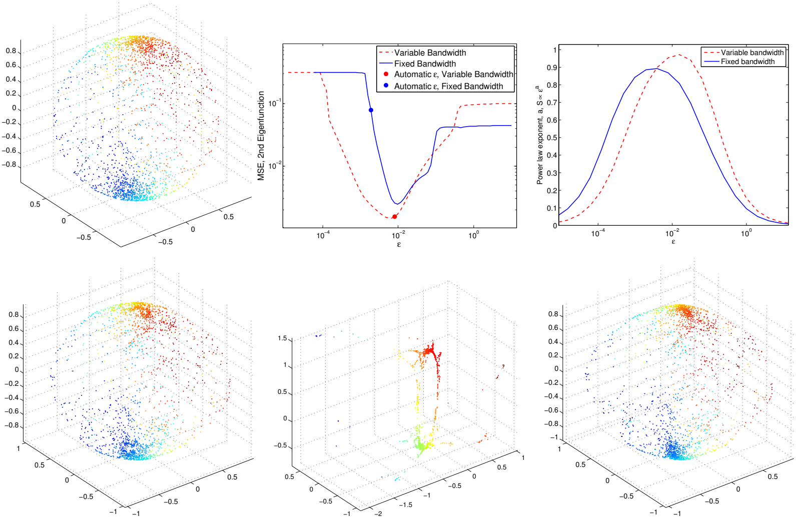
$$\begin{aligned} S(\epsilon) &\equiv \frac{1}{N^2}\sum_{i,j} K_\epsilon(x_i, x_j) \approx \frac{1}{Vol(\mathcal{M})}\int_{\mathcal{M}}\int_{T_{x_i}\mathcal{M}} K_\epsilon(x_i, y)\, dy\, dV(x) \\ &\approx \int_{\mathcal{M}} \frac{(4\pi\epsilon)^{d/2}}{Vol(\mathcal{M})} dV(x) = (4\pi\epsilon)^{d/2} \end{aligned}$$

such that,

$$\frac{dlogS}{dlog\epsilon} = d/2 \qquad (1)$$

Remark: As $\epsilon \to 0$, $S \to \frac{1}{N}$ and as $\epsilon \to \infty$, $S \to 1$ and in these extreme cases, the slopes of $\log S$ are zero. Our strategy is to determine ϵ and d that maximize (1).

Example: Estimation of Δ on $S^2 \in \mathbb{R}^3$ with $N = 3000$.



Other automatic estimation of ϵ and d

Let $X = [X_1, \ldots, X_N]$ and $x_i \in \mathcal{M} \subseteq \mathbb{R}^m$, where

$$\begin{aligned} X_j &= D(x)^{-1/2} \exp\Big(-\frac{\|x_j - x\|^2}{4\epsilon}\Big)(x_j - x) \\ D(x) &= \sum_{i=1}^{N} \exp\Big(-\frac{\|x_i - x\|^2}{2\epsilon}\Big) \end{aligned}$$

We showed[7] that

$$\lim_{N\to 0} \frac{1}{\epsilon} XX^\top = \mathcal{I}(x)^\top \mathcal{I}(x) + \mathcal{O}(\epsilon),$$

where $\mathcal{I} : \mathbb{R}^m \to T_x\mathcal{M}$ is a projection onto the tangent space.

Remarks: This means that for $\nu \in T_x\mathcal{M}$,

$$\lim_{N\to 0} \nu^\top XX^\top \nu = \epsilon\|\nu\|^2 + \mathcal{O}(\epsilon^2),$$

[7] Berry & H, Appl. Comput. Harmon. Anal., 2018

Other automatic estimation of ϵ and d

This means that, for $\nu \in T_x\mathcal{M}$, the singular value of X

$$\sigma_\nu := \lim_{N\to\infty} \frac{\sqrt{\nu^\top X X^\top \nu}}{\|\nu\|} = \sqrt{\epsilon} + \mathcal{O}(\epsilon).$$

and if $\nu \in T_x\mathcal{M}^\perp$, then $\sigma_\nu = \mathcal{O}(\epsilon)$.

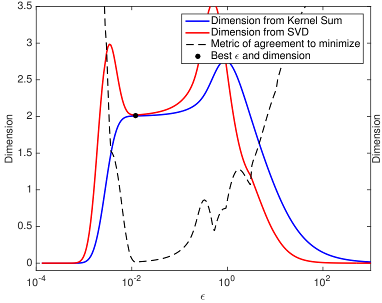
Thus, one can estimate the dimension using

$$d \approx \frac{1}{\epsilon} Trace(XX^\top),$$

or even using,

$$\left(det(XX^\top)\right) = \prod_{j=1}^{d} \sigma_j \approx \epsilon^d \Leftrightarrow d \approx \frac{d\left(det(XX^\top)\right)}{d\epsilon}$$

Example: 2D torus embedded in $\mathbb{R}^{30}$.



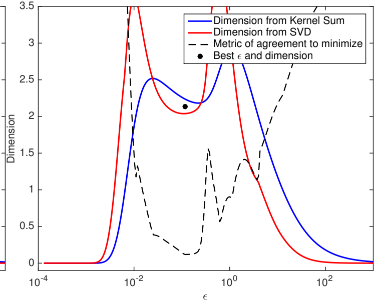


Figure : Dimension measures d_1 (blue) and d_2 (red) as functions of the bandwidth ϵ corresponding to the data set sampled from the torus embedded in 30-dim (left) and with 30-dim Gaussian noisy torus (right). The metric of agreement, $M(\epsilon)$, is shown as the dotted black curve. The solid black dot represents the bandwidth that minimizes the metric along with the average dimension at the optimal ϵ.

Discussion:

For junior participants:

- Convince yourself that the differential operator $\mathcal{L} = q^{-1}\text{div}(q\nabla \quad)$ that is being estimated is symmetric negative definite with respect to an appropriate Hilbert space.
- In the construction of matrix $L_{\epsilon,\alpha}$, notice that this $N \times N$ matrix is not symmetric. Can you find a similarity transformation to a symmetric matrix since we have a more stable algorithm for spd matrix.
- When N is large, you can store the matrix $L_{\epsilon,\alpha}$ and the entries of the matrix is mostly zero since the kernel is local with bandwidth ϵ. How do you get around of the storing and avoid computing zero entries.

Discussion:

A general research problem:

- Solving eigenvalue problem of such large system is very expensive. The amount of required data of any non-parametric method grows exponentially as a function of intrinsic dimension. Now, are there any computationally cheaper alternatives to get basis of the range of $L_{\epsilon,\alpha}$?
- I had explored one with QR decomposition[8] which is cheap but the problem is that QR basis does not reveal rank. Eigenbasis has a special properties since its corresponding eigenvalues $0 = \lambda_0 \geq \lambda_1 \geq \ldots$, and they satisfy

$$-\lambda_k = \arg \min_{f \in H^2(\mathcal{M},q) \cap \mathcal{H}_{k-1}^{\perp}} \|\nabla f\|_q$$

where $\mathcal{H}_{k-1} = \text{span}\{\varphi_0, \ldots, \varphi_{j-1}\}$.

[8] H & Yang, J. Nonlinear Science, 2018.

References:

- T. Berry and J. Harlim, *Variable bandwidth diffusion kernels*, Appl. Comput. Harmon. Anal. 40(1), 68-96, 2016.
- T. Berry and J. Harlim, *Iterated diffusion maps for feature identification*, Appl. Comput. Harmon. Anal. 45(1), 84-119, 2018.
- J. Harlim, *Data-driven computational methods: Parameter and operator estimations*, Cambridge University Press, UK, 2018. (with supplementary MATLAB codes for VBDM)
- J. Harlim and H. Yang, *Diffusion forecasting model with basis functions from QR decomposition*, J. Nonlinear Sci 28, 847-872, 2018.

Collaborators:

- Tyrus Berry, Assistant Professor at Department of Mathematical Sciences, George Mason University.
- Haizhao Yang, Assistant Professor at Department of Mathematics National University of Singapore.