Dimension Reduction for Data Science

Lin Bo-Ru
2019/12/26

0. Dimension Reduction

1. Dimension reduction seeks to produce a low dimensional representation of high dimensional data that preserves relevant structure.

2. Dimension reduction plays an important role in data science, being a fundamental technique in both visualization and as pre-processing for machine learning.

0.1 Geometric approach [4]

The geometric approach adopts manifold embedding technique that embeds a highdimensional data set into a low-dimensioal manifold, by applying eigenvector-optimization on a dimension reduction (DR) kernel. Hence, it is also called **spectral method**. A manifold embedding technique usually consists of the following steps:

1. For a given observed high-dimensional data, a neighborhood system is first designed on it to describe the similarity between the points (i.e., the digitized objects) of the data set.

2. The neighborhood system is formulated by the weighted graph mathematically.

3. This formulation equips the data with a manifold structure, presenting the data geometry. Each manifold has a coordinate system, which provides a low-dimensional representation of the original data.

4. The spectral decomposition of the weight matrix of the graph yields the coordinate representation, called the DR data. The weight matrix is called DR kernel.

Examples of the geometric approach

1. Linear DR methods: (That seek to preserve the distance structure within the data)
   Principal Components Analysis (PCA)
   Multidimensional Scaling (MDS)
   Random Projection

2. Nonlinear DR methods: (That favor the preservation of local distances over global distance)
   Isometric Feature Mapping (Isomap)
   Maximum Variance Unfolding (MVU)
   Locally Linear Embedding (LLE)
   Local Tangent Space Alignment (LTSA)
   Laplacian Eigenmaps
   Hessian Locally Linear Embedding (HLLE)
   Diffusion maps

1. Diffusion Maps

The Gaussian-type diffusion kernels were widely used in machine learning and data clusters before 2004. The mathematics of diffusion maps was first studied by Coifman and Lafon [1, 2].

1.1 Ideas of Diffusion Maps [1, 4, 5]

Diffusion maps exploit the relationship between heat diffusion and random walk Markov chain. The basic observation is that if we take a random walk on the data, walking to a nearby data-point is more likely than walking to another that is far away. Eigenfunctions of Markov matrices can be used to construct coordinates (diffusion maps) that generate efficient representations of complex geometric structures. The associated family of diffusion distances, obtained by iterating the Markov matrix, defines multiscale geometries that prove to be useful in the context of data parametrization and dimensionality reduction. Each of diffusion maps embeds the data set into a Euclidean space so that the Euclidean distance in the space is equal to the diffusion distance on the data.

1.2 Diffusion Maps Algorithms [1, 4]

Let given data $X \subset \mathbb{R}^D$ be the input of a diffusion maps algorithm and $(X, \mathcal{A}, \mu)$ be a measure space. Assume that the dimension of the dimension reduction (DR) data is d. Then the output of the algorithm is the DR data $Y \subset R^d$ in the matrix form. A diffusion maps algorithm consists of the following steps.

Step 1. Data graph construction.

Either k-neighborhood or ϵ-neighborhood can be used for defining data graph.

★ This step is same as other nonlinear DR methods.

Step 2. Weight matrix creation.

Assume that a date graph $[X, \mathbf{A}]$ is defined on the data set X with the adjacency matrix $\mathbf{A}$ to represent the edge set in a graph, which determines a neighborhood system on X. Let O_i denote the neighborhood of x_i and $N(i)$ denote the corresponding subscript set, which is read from the ith row of $\mathbf{A}$. For example, we can create the weight matrix $\mathbf{W} = [w_{ij}]_{i,j=1}^n$ by setting

$$w_{ij} := \begin{cases} \exp(-\dfrac{\|x_i - x_j\|^2}{\epsilon}) & , \quad j \in N(i) \cup \{i\} \\ 0 & , \quad otherwise \end{cases}$$

where $\epsilon > 0$ is a scaling parameter in the weight matrix and impacts the accuracy of DR since it dominates the measurement of data similarities.

★ Zelnik-Manor and Perona [7] proposed a variable local scaling parameter ϵ_i in the computation of the weight w_{ij}:

$$w_{ij} := \exp(-\frac{\|x_i - x_j\|^2}{\epsilon_i \epsilon_j})$$

Suggested in [7], for a fixed m, ϵ_i is computed by $\epsilon_i = d_2(x_i, x_{i_m})$ (the Euclidean distance), where x_{i_m} is the mth neighbor of x_i. The number m may be dependent on the dimension of the embedding space.

Step 3. Diffusion kernel construction.

We use the weight matrix to construct the diffusion maps kernel $k : X \times X \to \mathbb{R}$. The kernel function has the following properties:

1. k is symmetric: $k(x, y) = k(y, x), \forall x, y$
2. k is positivity preserving: $k(x, y) \geqslant 0, \forall x, y$

Then apply the **graph Laplacian normalization** to this kernel:

$$d(x) := \int_X k(x, y) d\mu(y)$$

to be a local measure of the volume (or degree in a graph) and define a new function

$$p(x, y) := \frac{k(x, y)}{d(x)}$$

Although p inherits the positivity-preserving property, it is no longer symmetric. However, we have gained a conservation property:

$$\int_X p(x, y) d\mu(y) = 1$$

This means that p can be viewed as the transition kernel of a Markov chain on X, or, equivalently, the operator $\mathbf{P}$ defined by

$$\mathbf{P}f(x) := \int_X p(x, y) f(y) d\mu(y)$$

preserves constant functions (it is an averaging or diffusion operator).

★ From a data analysis point of view, the reason for studying this Markov chain is that the matrix $\mathbf{P}$ contains geometric information about the data set X. Indeed, the transitions that it defines directly reflect the local geometry defined by the immediate neighbors of each node in the graph of the data. In other words, $p(x, y)$ represents the probability of transition in one time step from node x to node y and it is proportional to the edge-weight $k(x, y)$.

★ For $t \geqslant 0$, the probability of transition from x to y in t time steps is given by $p_t(x, y)$, the function of the t th power $\mathbf{P}^t$ of $\mathbf{P}$. Taking larger powers of $\mathbf{P}$, will allow us to integrate the local geometry and therefore will reveal relevant geometric structures of X at different scales. In addition to being the time parameter, t plays the role of a scale parameter.

Step 4. Use diffusion map to get the embedding and DR kernel decomposition.

Running the chain forward is equivalent to computing the powers of the operator $\mathbf{P}$. For this computation, we could, in theory, use the eigenvectors and eigenvalues of $\mathbf{P}$. Let $\psi_0, \psi_1, \ldots, \psi_d$ be the eigenvectors of $\mathbf{P}$ achieving the $(d+1)$ largest eigenvalues $1 = \lambda_0 > \lambda_1 \geqslant \lambda_2 \cdots \geqslant \lambda_d > 0$.

Instead, we are going to directly employ these objects in order to characterize the geometry of the data set X.

Definition

The family of **diffusion distances** $\{D_t\}_{t\in\mathbb{N}}$ given by

$$D_t(x,y) := \|p_t(x,\cdot) - p_t(y,\cdot)\|^2_{L^2(X,d\mu/\pi)} = \int_X [p_t(x,u) - p_t(y,u)]^2 \frac{d\mu(u)}{\pi(u)}$$

where

$$\pi(y) := \frac{d(y)}{\sum_{z\in X} d(z)}$$

is the stationary distribution of the Markov chain.

★ 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 $D_t(x,y)$ involves summing over all paths of length t connecting x to y and y to x.

★ As a consequence, this number is very robust to noise perturbation, unlike the geodesic distance. From a machine learning point of view, the same observation allows us to conclude that this distance is appropriate for the design of inference algorithms based on the majority of preponderance: this distance takes into account all evidences relating x and y.

Proposition $D_t(x, y)$ can be computed using the eigenvectors and eigenvalues of $\mathbf{P}$:

$$D_t(x, y) = \{\sum_{l \geqslant 1} \lambda_l^{2t} [\psi_l(x) - \psi_l(y)]^2\}^{\frac{1}{2}}$$

Definition

The family of **diffusion maps** $\{\Psi_t\}_{t \in \mathbb{N}}$ given by

$$\Psi_t(x) := \begin{bmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_d^t \psi_d(x) \end{bmatrix}.$$

For a given $\delta > 0$, we can choose $d(\delta, t) := \max\{l \in \mathbb{N} | \mid \lambda_l \mid^t > \delta \mid \lambda_1 \mid^t\}$. Then we have the family of **truncated diffusion maps** $\{\Psi_t\}_{t \in \mathbb{N}}$ given by

$$\Psi_t^\delta(x) := \begin{bmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{d(\delta,t)}^t \psi_{d(\delta,t)}(x) \end{bmatrix}.$$

and the family of **truncated diffusion distances** $\{D_t\}_{t \in \mathbb{N}}$ given by

$$D_t^\delta(x, y) = \{\sum_{1 \leqslant l \leqslant d(\delta,t)} \lambda_l^{2t} [\psi_l(x) - \psi_l(y)]^2\}^{\frac{1}{2}}.$$

★ The connection between diffusion maps and diffusion distances can be summarized as follows:

Proposition The diffusion map Ψ_t embeds the data into the Euclidean space $\mathbb{R}^{d(\delta,t)}$ so that in this space, the Euclidean distance is equal to the diffusion distance (up to δ), or equivalently,

$$\|\Psi_t^\delta(x) - \Psi_t^\delta(y)\| = D_t(x, y).$$

★ The reduced data have a twofold interpretation: Each column of the data matrix provides the feature coordinates of the objective vector in the original data, and each row represents a feature function on the data, arranged according to the degree of significance.

★ Diffusion maps are reduced to Laplacian eigenmaps when the eigenvalues are discarded from the mapping (using merely the eigenvectors).

1.3 Anisotropic Diffusions [1]

Let $\mathcal{M}$ be a compact C^∞ submanifold of $\mathbb{R}^n$. The heat diffusion on $\mathcal{M}$ is the diffusion process whose infinitesimal generator is the Laplace-Beltrami operator Δ. The Neumann heat diffusion operator $e^{-t\Delta}$ has the same eigenfunction set as the Laplace-Beltrami operator, but it is a bounded operator on $L^2(\mathcal{M})$ with the operator norm 1. Besides, all of its eigenvalues are nonnegative so that it is also a positive self-adjoint operator [6]. These important properties make diffusion operators widely used in many applications.

The operator Δ has eigenvalues and eigenfunctions on $\mathcal{M}$: $\Delta\phi_l = v_l\phi_l$, where $\partial\phi_l = 0$ at $\partial\mathcal{M}$ (Neumann condition). Let $E_K := Span\{\phi_l, 0 \leqslant l \leqslant K\}$ be the linear span of the first K + 1 Neumann eigenfunctions. Diffusion maps method uses the eigen decomposition of $e^{-t\Delta}$ to find the DR data set.

We assume that the data set X is the entire manifold. Let $q(x)$ be the density of the points on $\mathcal{M}$. Construction of the family of diffusions consists of the following steps.

1. Fix a number $\alpha \in \mathbb{R}$ and define a rotation-invariant kernel $k_\epsilon(x, y) := h(\frac{\|x-y\|^2}{\epsilon})$, where $h > 0$ is a decreasing function on $[0, \infty)$.
2. Let
$$q_\epsilon(x) := \int_X k_\epsilon(x, y)q(y)dy$$
and form the new kernel
$$k_\epsilon^{(\alpha)}(x, y) := \frac{k_\epsilon(x, y)}{q_\epsilon^\alpha(x)q_\epsilon^\alpha(y)}.$$
3. Apply the weighted graph Laplacian normalization to this kernel by setting
$$d_\epsilon^{(\alpha)}(x) := \int_X k_\epsilon^{(\alpha)}(x, y)q(y)dy$$
and by defining the anisotropic transition kernel
$$p_{\epsilon,\alpha}(x, y) := \frac{k_\epsilon^{(\alpha)}(x, y)}{d_\epsilon^{(\alpha)}(x)}.$$

Definition

Define
$$P_{\epsilon,\alpha} := \int_X p_{\epsilon,\alpha}(x, y)f(y)q(y)dy$$
and let $L_{\epsilon,\alpha} = \dfrac{I - P_{\epsilon,\alpha}}{\epsilon}$ be the infinitesimal generator of the Markov chain.

Theorem For a fixed $K > 0$, we have on E_K,

$$\lim_{\epsilon \to 0} L_{\epsilon,\alpha} f = \frac{\Delta(f q^{1-\alpha})}{q^{1-\alpha}} - \frac{\Delta(q^{1-\alpha})}{q^{1-\alpha}} f.$$

In particular, the eigenfunctions of $P_{\epsilon,\alpha}$ can be used to approximate those of the following symmetric Schrödinger operator:

$$\Delta\phi - \frac{\Delta(q^{1-\alpha})}{q^{1-\alpha}}\phi,$$

where $\phi = f q^{1-\alpha}$.

★ When $\alpha = 0$, the diffusion reduces to that of the classical normalized graph Laplacian normalization applied to the graph with isotropic weights. The influence of the density is maximal in this case.

For the intermediate case $\alpha = \frac{1}{2}$, the Markov chain is an approximation of the diffusion of a Fokker–Planck equation, allowing to approximate the long-time behavior or the point distribution of a system described by a certain stochastic differential equation.

When $\alpha = 1$, and if the points approximately lie on a submanifold of $\mathbb{R}^n$, one obtains an approximation of the Laplace-Beltrami operator. In this case, one is able to recover the Riemannian geometry of the data set, regardless of the distribution of the points. This case is particularly important in many applications.

★ When $\alpha = 1$, we obtain the following important result:

Proposition

$$\lim_{\epsilon \to 0} L_{\epsilon,1} = \Delta.$$

Furthermore, for any $t > 0$, the Neumann heat kernel $e^{-t\Delta}$ can be approximated on $L^2(\mathcal{M})$ by $P_{\epsilon,1}^{\frac{t}{\epsilon}}$

$$\lim_{\epsilon \to 0} P_{\epsilon,1}^{\frac{t}{\epsilon}} = e^{-t\Delta}.$$

1.4 Other Related Operators

Definitions [3]

Give two manifolds, denoted by $\mathcal{M}^{(1)}, \mathcal{M}^{(2)}$. $\phi^* : C^\infty(\mathcal{M}^{(2)}) \to C^\infty(\mathcal{M}^{(1)})$ denotes the operator corresponding to the pullback from $\mathcal{M}^{(2)}$ to $\mathcal{M}^{(1)}$, and $f \in C^\infty(\mathcal{M}^{(1)})$. V is the volume measure, $\mu(x')$ is the density function of the points on the manifold, ∇ denotes the covariant derivative on the manifold, and Δ denotes the Laplace-Beltrami operator.

1. **Backward diffusion operator** $P_\epsilon f(x) := \int p_\epsilon(x, x') f(x') \mu(x') dV(x')$
2. **Forward diffusion operator** $Q_\epsilon f(x) := \int q_\epsilon(x, x') f(x') \mu(x') dV(x')$
3. $G_{\epsilon_1,\epsilon_2} f(x) := \phi^* P^{(2)}_{\epsilon_2} (\phi^*)^{-1} Q^{(1)}_{\epsilon_1} f(x)$, where $\epsilon_1, \epsilon_2 > 0$
4. $H_{\epsilon_1,\epsilon_2} f(x) := P^{(1)}_{\epsilon_1} \phi^* Q^{(2)}_{\epsilon_2} (\phi^*)^{-1} f(x)$, where $\epsilon_1, \epsilon_2 > 0$
5. $S_{\epsilon_1,\epsilon_2} f(x) := \frac{1}{2}(G_{\epsilon_1,\epsilon_2} f(x) + H_{\epsilon_1,\epsilon_2} f(x))$, where $\epsilon_1, \epsilon_2 > 0$
6. $A_{\epsilon_1,\epsilon_2} f(x) := \frac{1}{2}(G_{\epsilon_1,\epsilon_2} f(x) - H_{\epsilon_1,\epsilon_2} f(x))$, where $\epsilon_1, \epsilon_2 > 0$
7. **Lie derivative** $\mathcal{L}_u f := u \cdot \nabla f$
8. **Liouvillian operator** $\mathcal{M}_u f := -\nabla \cdot f u$

References

[1] Ronald R. Coifman and Stéphane S. Lafon, *Diffusion maps*, Applied and Computational Harmonic Analysis 21, 5-30 2006.

[2] Stéphane S. Lafon, *Diffusion maps and geometric harmonics*, Ph.D. thesis, Yale University, 2004.

[3] Tal Shnitzer, Mirela Ben-Chen, Leonidas Guibas, Ronen Talmon and Hau-Tieng Wu, *Recovering hidden components in multimodal data with composite diffusion operators*, arXiv preprint arXiv:1808.07312., 2018.

[4] JianzhongWang, *Geometric structure of high-dimensional data and dimensionality reduction*, Heidelberg: Springer, 2012.

[5] Wikipedia contributors, *Diffusion map*, Wikipedia: The Free Encyclopedia. Wikimedia Foundation Inc. Updated 22 December 2019.

[6] Kosaku Yosida, *Functional Analysis*, springer, 1995.

[7] Lihi Zelnik-Manor and Pietro Perona, *Self-tuning spectral clustering*, Eighteenth Annual Conference on Neural Information Processing Systems (NIPS), 2004.