MATH 690: Topics in Probablity Theory	September 21, 2017

Dimension Reduction

Lecturer: Xiuyuan Cheng | *Scribe: Oscar Li*

1 Introduction

In this lecture, we will

- continue our discussion of Laplacian Eigenmap from last time;
- introduce a similar dimensionality reduction algorithm *Diffusion Map*;
- discuss some convergence results of eigenvalues and eigenvectors of the graph Laplacian L to the eigenvalues and eigenfunctions of the *Laplace-Beltrami operator* when the data points are sampled according to some distribution on the embedded manifold;
- introduce our last dimensionality reduction algorithm for this topic *tSNE*.

2 Laplacian Eigenmap

We recall from last time that given a point cloud $\{\mathbf{x}_i\}_{i=1}^n$, the Laplacian Eigenmap algorithm first constructs a heat kernel $W \in \mathbb{R}^{n\times n}$, where $W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\epsilon}} & \text{if } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise.} \end{cases}$. The connectedness between $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined on the knn graph or ϵ graph (notice here this ϵ need not be the same ϵ used in w_{ij}'s computation) induced by the pairwise distances between the points. A simpler alternative is to use the adjacency matrix A_{ij} of the knn or ϵ graph as the W_{ij}.

Then we create a diagonal "degree" matrix $D \in \mathbb{R}^{n\times n}$, where $D_{ii} = \sum_{j=1}^n W_{ij}$. For simplicity, we will use di in place of D_{ii} for the following discussion. For practical cases, we will assume D has full rank, i.e. no $d_i = 0$, from here.

The *graph Laplacian* is then defined as $L = D - W$, and the *normalized graph Laplacian* defined as $L_{rw} = D^{-1}L = D^{-1}(D-W) = I - D^{-1}W$. We define matrix $P = D^{-1}W$ and we can think of P as the probability transition matrix of a random walk on the graph $Pr[X_{t+1} = j | X_t = i] = P_{ij} = W_{ij}/d_i$, $\forall t$. So the transition probability is affected by the proximity of neighbors: the closer you are to your neighbor, the more probable you'll transition to it in the next state. We also see now $L_{rw} = I - P$.

Before we dive more into the dimensionality reduction algorithm, let's look into some properties and relationships of the matrices we just defined.

Proposition 1. L is positive semidefinite.

Proof.

$$\forall \mathbf{f} \in \mathbb{R}^n, \quad \mathbf{f}^T L \mathbf{f} = \frac{1}{2}\sum_{i,j} W_{ij}(f_i - f_j)^2 \geq 0, \text{ where } f_i \text{ is the ith coordinate of } \mathbf{f}.$$

, $\square$

Proposition 2. For any eigenvalue λ of P, λ is real and $|\lambda| \leq 1$.

Proof. Let $A_s = D^{-1/2}WD^{-1/2}$. We see A_s is a symmetric matrix because $D^{-1/2}$ and W are symmetric. Thus by spectral theorem, A_s has all real eigenvalues. We also observe that $P = D^{-1}W = D^{-1/2}D^{-1/2}WD^{-1/2}D^{1/2} = D^{-1/2}A_sD^{1/2}$, so P is similar to A_s. Because similar matrices have the exact same set of eigenvalues, all the eigenvalues of P must also be real.

Suppose $\boldsymbol{\psi}$ is an eigenvector of P with eigenvalue λ. We choose i s.t. $|\psi_i| \geq |\psi_j|, \quad \forall j = 1, ..., n$. Here ψ_i means the ith coordinate of vector $\boldsymbol{\psi}$.

$\lambda\psi_i = (P\psi)_i = \sum_{j=1}^{n} P_{ij}\psi_j$. Taking the absolute value of both sides and using triangle inequality, we have $|\lambda||\psi_i| = |\sum_j P_{ij}\psi_j| \leq \sum_j P_{ij}|\psi_j| \leq \sum_j P_{ij}|\psi_i| = |\psi_i|$. Since ψ_i is the largest in absolute value, $\psi_i \neq 0$. Thus $|\lambda| \leq 1$. $\square$

Remark 1. (by the Scriber) P is a right stochastic matrix. The right spectral radius of any right stochastic matrix is at most 1. But a matrix being right stochastic does not always imply all its eigenvalues are real. For example, $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. P has all eigenvalues real specifically because of the symmetricity of W.

After obtaining the normalized graph Laplacian L_{rw}, we compute its eigendecomposition. This is achievable because we have just proved that P is similar to a symmetric matrix A_s that is eigendecomposable by the spectral theorem. Since $L_{rw} = I - P$, we know L_{rw} is decomposable. Let $\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_n$ be the n eigenvectors of L_{rw} with their corresponding eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$. Then to compress the point clouds $\{\mathbf{x}_i\}_{i=1}^{n}$ into d-dimension, the eigenmap is defined as $\Psi(\boldsymbol{x}_i) = (\boldsymbol{\psi}_2(i), \boldsymbol{\psi}_3(i), ..., \boldsymbol{\psi}_{d+1}(i)) \in \mathbb{R}^d$, where $\boldsymbol{\psi}_j(i)$ means the ith coordinate of the vector $\boldsymbol{\psi}_j$.

One way to find the eigenvectors for L_{rw} is to find the eigenvectors of P (P and L_{rw} have the same eigenvectors) through the eigendecomposition of A_s. Because A_s is symmetric and have the same eigenvalues as P due to similarity, we can find an orthonormal basis of eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n$ with corresponding eigenvalues $1 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq -1$. We write this succinctly as $A_s = U\Lambda U^T$, where U is an orthogonal square matrix whose ith column is $\boldsymbol{u}_i$ and Λ is a diagonal matrix with decreasing eigenvalues on the diagonal. We let matrix $\Psi = D^{-1/2}U$ and see that

$$P\Psi = D^{-1/2}A_sD^{1/2}D^{-1/2}U = D^{-1/2}U\Lambda U^TU = D^{-1/2}U\Lambda = \Psi\Lambda \tag{1}$$

We see from Equation 1 that every column of Ψ must be an eigenvector of P and the set of all column vectors forms an eigenbasis of P.

$$L_{rw}\Psi = (I - P)\Psi = (I - \Lambda)\Psi \tag{2}$$

shows that the columns of Ψ are already sorted increasingly according to their corresponding eigenvalues in L_{rw}. So Ψ gives us the eigenvectors we want in the algorithm.

Remark 2. We can similarly define a matrix $\Phi = D^{1/2}U$ and see that the following is true:

$$\Phi = D\Psi \tag{3}$$
$$\Phi^T\Psi = I \tag{4}$$
$$\Phi\Psi^T = I \tag{5}$$
$$P = \Psi\Lambda\Phi^T \tag{6}$$
$$\Psi^T D\Psi = I \tag{7}$$
$$P^T\Phi = \Phi\Lambda \tag{8}$$

We see from Equation 8 that the columns of Φ are the eigenvectors of P^T. Then the first column of Φ, $\boldsymbol{\phi}_1$, is an eigenvector of P^T with eigenvalue 1. From Equation 3 we see that $\boldsymbol{\phi}_1 = D\boldsymbol{\psi}_1$. Because P is a right stochastic matrix, the first eigenvector of P, $\boldsymbol{\psi}_1$, must have all the coordinates the same, i.e. $\psi = \begin{bmatrix} c \\ . \\ . \\ . \\ c \end{bmatrix}$. As a result, $\boldsymbol{\phi}_1 = c\begin{bmatrix} d_1 \\ . \\ . \\ . \\ d_n \end{bmatrix}$. This implies that the invariant measure (stationary distribution) $\boldsymbol{\pi}$ of the random walk specified by P satisfies that $\pi_j = \frac{d_j}{\sum_{i=1}^n d_i}$ due to a normalization of $\boldsymbol{\phi}_1$.

3 Diffusion Map

Diffusion Map [CLL+05][CL06] is a similar dimensionality reduction algorithm as Laplacian Eigenmap. For some fixed $t > 0$, the diffusion map $\Psi_t^d(\boldsymbol{x}_i) = ({\lambda_2}^t\boldsymbol{\phi}_2(i), ..., (\lambda_{d+1})^t\boldsymbol{\phi}_{d+1}(i))$ compresses every $\boldsymbol{x}_i$ to a d-dimensional vector. $\boldsymbol{\phi}_j$ denotes the same vector as in the previous section, and λ_j is the jth largest eigenvalues of P, i.e. $1 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$.

Definition 1. The diffusion distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is defined as $D_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||\Psi_t^{n-1}(\boldsymbol{x}_i) - \Psi_t^{n-1}(\boldsymbol{x}_j)||^2$

Remark 3. Suppose $|\lambda_k| \searrow 0$ as $k \uparrow$, then when t is large, $D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \approx ||\Psi_t^d(\boldsymbol{x}_i) - \Psi_t^d(\boldsymbol{x}_j)||^2$

Proposition 3. If we think of P as the transition probability matrix for the random walk $\{X_t\}_{t=1}^\infty$ on the graph indices, meaning $(P^t)_{ij} = Pr[X_{s+t} = j | X_s = i]$, $\forall s$, then $D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) = ||(P^t)_{i,:} - (P^t)_{j,:}||_w^2$, where $(P^t)_{i,:}$ means the ith row vector of P^t and the squared norm on the right hand side is a weighted L^2 norm where $w_k = \frac{\sum_{i=1}^n d_i}{d_k}$, $\forall k$.

Remark 4. When the graph is connected, $\lambda = 1$ is of multiplicity 1. From Equation 6 and 4, we know $P^t = \Psi\Lambda^t\Phi^T$. As $t \to \infty$, only the first eigenvalue which is equal to 1 survives in Λ^t. As a result, $\lim_{t\to\infty} P^t = \boldsymbol{\psi}_1\boldsymbol{\phi}_1^T$. Because every coordinate in $\boldsymbol{\psi}_1$ is the same, by the Proposition above, the diffusion distance would become zero for any pair of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as $t \to \infty$.

Proof of Proposition 3.

$$(P^t)_{il} = \sum_{k=1}^n \lambda_k^t \boldsymbol{\psi}_k(i)\boldsymbol{\phi}_k(l) = \sum_{k=1}^n \lambda_k^t \boldsymbol{\psi}_k(i)\boldsymbol{\psi}_k(l)d_l \tag{9}$$

$$\sum_{l=1}^{n}((P^t)_{il}-(P^t)_{jl})^2 w_l = \ldots = \sum_{l,l'}[\lambda_l^t \lambda_{l'}^t (\boldsymbol{\psi}_l(i)-\boldsymbol{\psi}_l(j))(\boldsymbol{\psi}_{l'}(i)-\boldsymbol{\psi}_{l'}(j)) \sum_k \boldsymbol{\psi}_l(k)\boldsymbol{\psi}_{l'}(k) d_k] \tag{10}$$

We see from Equation 7 that $\boldsymbol{\psi}_l^T D \boldsymbol{\psi}_{l'} = \delta_{ll'}$. Therefore,

$$\sum_{l=1}^{n}((P^t)_{il}-(P^t)_{jl})^2 w_l = \sum_{l=1}^{n} \lambda_l^{2t}(\boldsymbol{\psi}_l(is)-\boldsymbol{\psi}_l(j))^2 = D_t^2(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{11}$$

□

4 Convergence of Eigenmap

Definition 2. Let (M, g) be a compact Riemannian manifold with no boundary, the *Laplace-Beltrami operator* on M is defined as: $\Delta_M : C^2(M) \to L^2(M), \quad \Delta_M(f) = -div(\nabla f)$.

Remark 5. We state some properties of the Laplace-Beltrami operator:

- Δ_M is a linear operator that is positive semidefinite with a discrete spectrum of eigenvalues $\{\lambda_k\}, 0 \le \lambda_1 \le \lambda_2 \le \ldots$;
- all eigenfunctions of Δ_M are in $C^\infty(M)$.
- the operator is "intrinsic" – it only sees the Riemannian metric tensor g but not the specific embedding in the high-dimensional ambient space.

Theorem 4. [BN07] If the point cloud $\{\mathbf{x}_i\}_{i=1}^n$ are i.i.d. sampled uniformly from manifold M and ϵ_n properly set so that $\epsilon_n \to 0$ as $n \to \infty$,

then for each k, $\widehat{\lambda_k} \to \lambda_k$ and $\widehat{\boldsymbol{\psi}_k} \to \boldsymbol{\psi}$ in probability as $n \to \infty$, where $\widehat{\lambda_k}$ and $\widehat{\boldsymbol{\psi}_k}$ are the kth eigenvalue-eigenvector pair of the normalized graph Laplacian, while λ_k and $\boldsymbol{\psi}_k$ are the kth eigenvalue-eigenfunction pair of the Laplace-Beltrami operator. More precisely, $\widehat{\boldsymbol{\psi}_k} \to \boldsymbol{\psi}$ means $\widehat{\boldsymbol{\psi}_k}(i) \to \boldsymbol{\psi}_k(\boldsymbol{x}_i), \ \forall i = 1, \ldots, n$.

Remark 6. When the point cloud is not uniformly sampled from M but instead sampled from a distribution p, the convergence result is as follows:

$$\frac{1}{\epsilon} L_{n,\epsilon} = -\frac{1}{2}(\Delta_M + 2\frac{\nabla p}{p} \cdot \nabla) + O(\epsilon) \tag{12}$$

, when $n \to \infty$ and $\epsilon \to 0$.

5 tSNE

t-Distributed Stochastic Neighbor Embedding (tSNE) [MH08] is the last dimensionality reduction technique we will introduce in this topic. It can achieve good results on real-life datasets such as MNIST. However, there is not a lot of theoretical result behind this method.

tSNE works as follows: for a point cloud $\{\mathbf{x}_i\}_{i=1}^n$, Step 1: For $i \neq j$, let $W_{ij} = e^{-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma_i}}$, where σ_i is tuned for each $\boldsymbol{x}_i$. For $i = j$, $W_{ii} = 0$.

Let P be such that $P_{ij} = \frac{W_{ij}}{\sum_{j'=1}^{n} W_{ij'}}$, and initialize $\overline{P} = \frac{P+P^T}{2}$. Now $\overline{P}$ is symmetric.

Step 2: We want to fit $Y \in \mathbb{R}^{d \times n}$, whose ith column is $\boldsymbol{y}_i$, to have the same transition probability matrix $Q_{ij}[Y]$ as $\overline{P}_{ij}$, where $Q_{ij}[Y] = \frac{k(||\boldsymbol{y}_i - \boldsymbol{y}_j||)}{\sum_{i' \neq j'} k(||\boldsymbol{y}_{i'} - \boldsymbol{y}_{j'}||)}$, $k(x) = \frac{1}{1+x^2}$.

The loss function $C(Y)$ is formulated using KL-divergence and the optimization for Y is as follows:

$$\min_{Y} C(Y) = \sum_{i \neq j} \overline{P}_{ij} log \frac{\overline{P}_{ij}}{Q_{ij}}$$

.

The gradient of $C(Y)$ with repect to each $\boldsymbol{y}_i$ is given by

$$\frac{\partial C}{\partial \boldsymbol{y}_i} = 4 \sum_{j=1}^{n} (\overline{P}_{ij} - Q_{ij})(\boldsymbol{y}_i - \boldsymbol{y}_j)(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2)^{-1}$$

The gradient descent update step introduced in [MH08] uses a momentum term to reduce the number of iterations required and works best if the momentum term is small until the map points have become moderately well organized:

$$Y^{(t+1)} = Y^{(t)} + \eta \frac{\partial C}{\partial Y} + \alpha(t)(Y^{(t)} - Y^{(t-1)})$$

, where η is the learning rate and $\alpha(t)$ is the momentum.

References

[BN07] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *Advances in Neural Information Processing Systems*, pages 129–136, 2007.

[CL06] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[CLL+05] Ronald R Coifman, Stephane Lafon, Ann B Lee, Mauro Maggioni, Boaz Nadler, Frederick Warner, and Steven W Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005.

[MH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.