

Dimension Reduction

Lecturer: Xiuyuan Cheng

Scribes: Dev Dabke and Andrew Cho

1 Introduction

The lecture covered

- A wrap up of *Topic 2: Dimension Reduction* by discussing convergence of eigenmaps when $p(x)$ is not uniform
- The start of *Topic 3: Clustering*

2 Dimension Reduction

Recall the convergence of eigenmap

$$\begin{aligned} L_{n,\epsilon} &\xrightarrow{n \rightarrow \infty} L_\epsilon \xrightarrow{\Sigma \rightarrow 0} L \\ L_\epsilon f &\xrightarrow{\epsilon \rightarrow 0} Lf \end{aligned} \quad \text{for each } f$$

This is a point-wise convergence operator and doesn't necessarily mean uniform convergence. Rather, what we need is a convergence of the spectrum $\text{eig } L_\Sigma \rightarrow \text{eig } L$. In essence, we seek $\sup \|L_\Sigma f - Lf\| \rightarrow 0$ where $f \in C^2(M)$, $\|f\|_2^2 = 1$, and $\int f(x)^2 dp(x) = 1$. Unfortunately, these last two conditions are not always true. [BN03]

Definition 2.1: Heat Kernel

Where $t = \frac{1}{2}\epsilon$

$$\begin{aligned} L_t &= \frac{I_\alpha - H_t}{t} + R_t \\ H_t f(x) &= u(x, t) \end{aligned}$$

where the function u has constraints

$$\begin{aligned} u_t &= -\Delta_M u \\ u(x, 0) &= f(x) \end{aligned}$$

As a result, we have that the residual $\|R_t\|$ can be controlled properly which implies that $\text{eig } L_t = \text{eig}(\frac{I_d - H_t}{t})$ and $H_t f = e^{-t\Delta_M} f$

Remark 2.1: Exponential ODE

$$y'(t) = -at \implies y = e^{-at}y(0).$$

Additionally

$$\frac{1 - e^{-t\lambda_k}}{t} \xrightarrow{t \rightarrow 0} \lambda_k$$

Anyways, note that

$$H_t f = e^{-t\Delta_M} f$$

such that $\Delta_M : \{\lambda_k, \psi_k\}_k$ and that $H_t : \{e^{-t\lambda_k}, \psi_k\}_k$ such that $k = 1, \dots, d$.

Remark 2.2: $p(x)$ Uniformity

When $p(x)$ is not uniform, then

$$L_{n,\epsilon} \rightarrow L_{FK}$$

where $L_{FK} f = \Delta_M f - \nabla u \dot{\nabla} f$. and that

$$\begin{aligned} p(x) &= e^{-\frac{1}{2}u(x)} \\ u(x) &= -2 \log p(x) \end{aligned}$$

by Fokker-Planck.

We have to perform a “correction” of density by defining a Weight Matrix W such that $W_{ij} = k(x_i, x_j)$.

Definition 2.2: Affinity Matrix

Let

$$\begin{aligned} d_i &= \sum_j k(x_i, x_j) \\ \tilde{k}(x, y) &= \frac{k(x, y)}{\sqrt{d(x)} \sqrt{d(y)}} \\ d(x) &= \int_M k(x, y) p(y) dy \end{aligned}$$

where d is the degree function.

In practice, we cannot take a continuous integral, so instead we compute

$$d_R(x) = \frac{1}{n} \sum_{j=1}^n k(x, x_j) \xrightarrow{n \rightarrow \infty} d(x)$$

and we let

$$\widetilde{W}_{ij} = \frac{W_{ij}}{\sqrt{d(x)} \sqrt{d(y)}}$$

and so consider the eigenmap from \widetilde{W} instead of W .

Theorem 2.1: Convergence of L

Given the matrix $\tilde{L}_{rw} = I - \tilde{D}^{-1}\tilde{W}$ where

$$\tilde{D}_{ij} = \sum_j \tilde{W}_{ij}$$

then

$$\tilde{L}_{n,\epsilon} \xrightarrow{n \rightarrow \infty, \epsilon \rightarrow \infty} \Delta_M$$

Proof 2.1: Convergence of L , Theorem 2.1

The proof is omitted, but as hint, note that $\epsilon \rightarrow 0$, $d_\epsilon(x) \approx p(x) \cdot \text{constant}$.

Additionally, we can generalize this to a graph Laplacian with any $0 < \alpha < 1$. The corrected kernel \tilde{k} above uses $\alpha = \frac{1}{2}$. Therefore, we write

$$\tilde{L}_\alpha = \frac{W_{ij}}{d_i^\alpha d_j^\alpha}$$

Recall that $k_\epsilon(x, y) = e^{-\frac{\|x-y\|^2}{2\epsilon}}$ and $d_\epsilon(x) = \int_M k_\epsilon(x, y)p(y)dy \approx p(x)$.

3 Topic 3: Clustering

We start the discussion of our third topic on clustering by defining what the problem of clustering is.

Problem: given $\{x_i\}_{i=1}^n$, find clusters. These clusters may or may not have labels (*supervised* vs. *unsupervised* learning). There are many possible definitions and models of clusters. For example, we will consider two possible cases:

1. given data points
2. given graph, affinity matrix W is $n \times n$ where W_{ij} is the similarity of node i and j

3.1 Case 1: With Data Points

We will consider a better and precise formulation of “clusters” using a scheme of “hard membership.”

Definition 3.1: Cluster

Given $\{x_i\}_{i=1}^n$, find a partition of the vertices $\mathcal{V} = \{1, \dots, n\}$ into disjoint subsets $\mathcal{C} = \{C_1, \dots, C_k\}$ such that

$$\mathcal{V} = \bigcup_{C \in \mathcal{C}} C$$

where “disjoint” means $C_l \cap C_{l'} = \{\emptyset\} \iff l \neq l'$.

We say that each C_i is the i^{th} cluster.

Remark 3.1: Soft Membership

We can also consider some idea of “soft membership.” In this case, we have some probability profile over each node such that $\mathbb{P}(\text{node } i \in C_l) = p_{i,l}$ with the constraint that $\forall i, \sum_{l=1}^k p_{il} = 1$

Definition 3.2: k -means

We use the following algorithm [Llo82]

1. Seeding: Randomly generate “centroids” $\{\mu_1, \dots, \mu_k\} = \mu$
2. Assignment: $\forall i$ assign x_i to the closest centroid in μ and this gives a partition \mathcal{C}
3. Update of μ : for $l = 1, \dots, k$ we compute an updated μ'_l where we let

$$\mu'_l = \frac{1}{|C_l|} \sum_{i \in C_l} x_i$$

and $|C_l|$ is “the cardinal number of the set C_l .”

After step 3, we repeat step 2 – 3 until we reach the stopping condition: $\|\mu_{\text{NEW}} - \mu_{\text{OLD}}\| < \delta$ for some tolerance level δ .

Theorem 3.1: Optimality of k -means

The process in Definition 3.2 solves the objective function

$$\operatorname{argmin}_{\mu, \mathcal{C}} \sum_{l=1}^k \sum_{i \in C_l} \|x_i - \mu_l\|^2$$

Remark 3.2: k -means and k -medians

The squared L^2 norm $\|x_i - \mu_l\|_2^2$ gives the formulation of k -means. If using the (unsquared) L^1 norm $\|x_i - \mu_l\|_1$, it leads to the objective function of k -medians. One can also remove the square, that is, using $\|x_i - \mu_l\|_2$ instead of $\|x_i - \mu_l\|_2^2$, which is a mixed L^2 - L^1 norm.

References

- [BN03] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003.
- [Llo82] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982.