Math 690: Topics in Data Analysis and Computation
Lecture notes for September 29, 2017

Scribed by Dev Dabke and Andrew Cho

1 Introduction

The lecture covered the following:

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

2 Wrap up Dimension Reduction

Recall the convergence of eigenmap

$$L_{n,\epsilon} \xrightarrow{n\to\infty} L_\epsilon \xrightarrow{\Sigma\to 0} L$$

$$L_\epsilon f \xrightarrow{\epsilon\to 0} Lf \text{ , for each } f.$$

This is a point-wise convergence operator and doesn't necessarily mean everything converges. Rather, what we need is a convergence of the spectrum $\operatorname{eig} L_\Sigma \to \operatorname{eig} L$. In essence, we seek $\sup \|L_\Sigma f - Lf\| \to 0$ 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.

Ref (Belkin-Niyogi 2006)
$t = \frac{1}{2}\epsilon$ with

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

defined as

$$\begin{cases} \partial_t u(x,t) & = -\Delta_M u(x,t) \\ u(x,0) & = f(x) \end{cases}$$

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

As an aside, note that for $y'(t) = -at \implies y = e^{-at} y(0)$.

Additionally

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

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 $k = 1, \ldots, d$

Remark that when $p(x)$ is not uniform then

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

where $L_{FK} f = \Delta_M f - \nabla u \dot{\nabla} f$ and noting that $p(x) = e^{-\frac{1}{2}u(x)}$ (Fokker-Plank) that $u(x) = -2 \log p(x)$

We have to perform a "correction" of density

$$W_{ij} = k(x_i, x_j)$$

and let

$$d_i = \sum_j k(x_i, x_j)$$

where we perform the correction by defining

$$\widetilde{k}(x, y) = \frac{k(x, y)}{\sqrt{d(x)}\sqrt{d(y)}}$$

and

$$d(x) = \int_M k(x, y) p(y) dy$$

is the degree function. However, we can never take a continuous integral in practice, so we instead compute

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

and we let

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

and we should consider the eigenmap from $\widetilde{W}$ instead of W.

We consider the matrix $\widetilde{L}_{rw} = I - \widetilde{D}^{-1}\widetilde{W}$ where

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

and we have that

$$\widetilde{L}_{n,\epsilon} \xrightarrow{n\to\infty\, \epsilon\to\infty} \Delta_M$$

The proof is omitted, but hint: as $\epsilon \to 0$, $d_\epsilon(x) \approx p(x) \cdot$constant. Additionally, we can generalize this to a graph Laplacian with any $0 < \alpha < 1$. The corrected kernel $\widetilde{k}$ above uses $\alpha = \frac{1}{2}$.

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

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."

Given $\{x_i\}_{i=1}^n$, find a partition of the vertices $\mathcal{V} = \{1, \ldots, n\}$ into disjoint subsets

$$\mathcal{C} = \{C_1, \ldots, C_k\}$$

i.e. $C_l \cap C_{l'} = \{\emptyset\} \iff l \neq l'$ such that

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

and we say that each C_i gives the i^{th} cluster.

Remark: 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 $\forall i,\ \sum_{l=1}^k p_{il} = 1$

At any rate, we could start with k-means. The process is as follows (Llyod's Algorithm 1957)

1. Randomly generate "centroids" $\{\mu_1, \ldots, \mu_k\} = \mu$
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 $l = 1, \ldots, k$ 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.

This process gives us the objective function we are ultimately solving for.

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

Remark. 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.