CS369M: Algorithms for Modern Massive Data Set Analysis **Lecture 9 - 10/21/2009**

Spectral & Kernel Methods for Nonlinear Dimensionality Reduction (2 of 2)

Lecturer: Michael Mahoney *Scribes: Yunting Sun and Meghana Vishvanath*

**Unedited Notes*

1 Laplacian Eigenmaps

Give a data set, we can construct a graph $G = (V, E)$. To reduce the dimensionality we want to map the graph to a line by minimizing $\sum_{i,j} w_{ij}(y_i - y_j)^2$ under appropriate constraints, where w_{ij} is the edge weight between vertex i and j. The objective function with our choice of weights w_{ij} incurs a heavy penalty if neighboring points are mapped far apart. Therefore, minimizing it is an attempt to ensure that if vertex i and j are close then y_i and y_j are close as well.

Claim: $\frac{1}{2}\sum_{i,j} w_{ij}(y_i - y_j)^2 = y^T L y$

Proof:

$$\begin{aligned}\frac{1}{2}\sum_{i,j} w_{ij}(y_i - y_j)^2 &= \frac{1}{2}\sum_{i,j} w_{ij}y_i^2 + w_{ij}y_j^2 - 2w_{ij}y_i y_j \\ &= \sum_i D_{ii}y_i^2 - \sum_{ij} w_{ij}y_i y_j \\ &= y^T L y\end{aligned}$$

where $L = D - W$, W is the weight matrix and L is Laplacian, a symmetric, positive semidefinite matrix that can be thought of as an operator on functions defined on vertices of G.

The solution to

$$\begin{aligned}&\arg\min y^T L y \\ \text{s.t.}\quad &yD\vec{1} = 0 \\ &y^T D y = 1\end{aligned}$$

turns out to be solving a generalized eigenvector problem

$$Ly = \lambda Dy$$

The condition $y^T D\vec{1} = 0$ can be interpreted as removing a translation invariance in y. The condition $y^T Dy = 1$ removes an arbitrary scaling factor in the embedding.

2 Random walk on the graph

Graph G defines a random walk. For some node i the probability going from i to j is $P'_{ij} = \frac{w_{ij}}{d_i}$, where $d_i = \sum_j w_{ij}$. Consider if you are at node i and you are move from i in the following way:

$$\begin{cases}\text{move to a neighbor chosen u.a.r } \frac{1}{d_i} & \text{w.p.}\frac{1}{2} \\ \text{stay at node i} & \text{w.p.}\frac{1}{2}\end{cases}$$

Then the transition matrix $P \in \mathcal{R}^{n\times m} = \frac{1}{2}I + \frac{1}{2}P'$

Fact if G is connected,for any measure initial v on the vertex, $\lim_{t\to\infty} P^t v = \frac{d_i}{\sum_j d_j} = \phi_0(i)$. This ϕ converges to the stationary distribution. P is related to the normalized Laplacian.

for $1 << t << \infty$, we could define similarity between vertex x and z in terms of the similarity between two density $P_t(x,\cdot)$ and $P_t(z,\cdot)$.

The L_2 distance is defined as $D_t^2(x,z) = ||P_t(x,\cdot) - P_t(z,\cdot)||^2 = \sum_y \frac{(P_t(x,y)-P_t(z,y))^2}{\phi_0(y)}$

Suppose the transition matrix P has q left and right eigenvectors such that $P_t(x,y) = \sum_j \lambda_j^t \psi_j(x)\phi_j(y)$. Then the L_2 distance turns out to be $D_t^2(x,z) = \sum_j \lambda_j^{2t}(\psi_j(x) - \psi_j(z))^2$.

$$\text{Map } \Psi_t : X \to \begin{pmatrix} \lambda_1^t\psi_1(x) \\ \vdots \\ \lambda_n^t\psi_n(x) \end{pmatrix}$$

$$\begin{array}{ccc} G & \leftrightarrow & R^n \\ | & & | \\ \text{diffusion} & \leftrightarrow & ||\cdot||_2 \end{array}$$

Fact: This defines a distance. If you think of a `diffusion` notion of distance in a graph G, this identically equals the Euclidean distance $||\cdot||_2$ between $\Psi(i)$ and $\Psi(j)$. The diffusion notion of distance is related to the commute time between node i and node j.

In particular, **Laplacian Eigenmaps** (LE) chooses $k = k^*$ and sets $t = 0$. Under nice limits, this is close to the Laplacian-Beltrami operator on the manifold. More generally, this generalization will hold even if the graph is not nice.

These methods take a set of data points and embed them so that you can do something faster. The motivating question that arises is how do you deal with new data points?

3 Nyström method

You can use this method to get coordinates for new data points. It is similar to the Trapezoidal rule for integrals. We begin by choosing a smart set of points.

You have the eigenfunction problem

$$\int_D K(t,s))\phi(s)ds = \phi(t)$$

over some infinite domain D.

The domain is discretized, you solve the new matrix eigenvalue problem, and then extend your solution in $\mathcal{R}^n$ to D. You break up $\mathcal{R}^n$ into $\{y_i\}_{i=1}^n$ and your D becomes R.

Say $D = [a,b]$. **Quadrature Rule** is $\int_a^b y(s)ds = \sum_{i=1}^n w_i y(s_i)$.
Now,

$$\int_a^b K(x,s)\phi(s)ds \approx \sum_{j=1}^n w_j K(x,s_j)\hat{\phi}(s_j)$$

Nyström gives you a technique for solving this. You first choose a set of Nyström points (they are typically the same as quadrature points). This gives

$$\sum_{j=1}^{n} N_j K(x_i, s_j)\hat{\phi}(s_j) = \hat{\lambda}\hat{\phi}(x_i) \quad \forall i$$

where $N \in \mathcal{R}^{n\times n}$ is the eigenvalue problem. After solving this, we can extend back up to the eigenfunctions (in order to get $\hat{\lambda}_i, \hat{\phi}_i$) by

$$\hat{\phi}_m(x) = \frac{1}{\hat{\lambda}_m} \sum_{j=1}^{n} w_j K(x, s_j \hat{\phi}_m(s_j)$$

x_{n+1} uses $\hat{\phi}_m(x)$ to embed a new point (inference).

Side Note: Consider the problem of sampling columns from a matrix $A \in \mathcal{R}^{n\times n}$. We randomly choose a $c \in \mathcal{R}$ such that $C \in \mathcal{R}^{n\times c}$ and such that $|ACC^TA|$ is good.
From the SVD of C, we have $C = H\Sigma_c Z^T$ and $H = C_{(n\times c)} Z_{(c\times c)} \Sigma^{-1}_{(c\times c)}$

Side Note 2: Two ways to view random sampling:

(1) Algorithmically.
Given a matrix A, output a good matrix C.

(2) Statisically or Machine Learning/Inference.
Say A is the infinite dimensional real world and $C \in \mathcal{R}^{n\times c}$ is the single sample you're given. Look at $C^TC \in \mathcal{R}^{c\times c}$ and consider SVD of C to get an expansion of $C^TC = Z\Sigma^2 Z^T$. This is akin to kernal methods (which work in lower dimensions and make statements about higher dimension). Similarly, if we're trying to say something about A, we can say $U_A \approx U_C = H = CZ\Sigma^{-1}$

4 Expander Graphs

Assume a graph that is sparse and well connected. Say you have a grid with a node at every intersection that has dimension $(\sqrt{n} \times \sqrt{n})$. This graph has n nodes and n edges.

Now consider a complete graph that is well connected and dense, i.e., has n nodes and n^2 edges.

Side Note: Also consider G_{np} like the Erdos-Renyi random graphs. If $p = \frac{1}{2}$, there are $\frac{n^2}{2}$ edges. If you get a graph this dense, you get good qualities like measure concentration.

This brings us back to the question first introduced in Lecture 3. Which space (or class of graphs) is least like the Euclidean space? The answer is an expander, and they are usually studied in derandomization.

Wigner: $\lambda_{\max} \sim \sqrt{n}$ and that the singular values are nice and mass is spread out.
Consider G_{np} where $\frac{1}{n} < p < \frac{\log(n)}{n}$. Typically, you get localization since $G_{np} \neq G_{nm}$ where m is the expected value.