A Mathematical Introduction to Data Science **Apr. 15, 2020**

Homework 7. Markov Chains on Graphs and Spectral Theory

Instructor: Yuan Yao *Due: 1.5 weeks later*

The problem below marked by * is optional with bonus credits.

1. *PageRank*: The following dataset contains Chinese (mainland) University Weblink during 12/2001-1/2002,

   `https://github.com/yao-lab/yao-lab.github.io/blob/master/data/univ_cn.mat`

   where `rank_cn` is the research ranking of universities in that year, `univ_cn` contains the webpages of universities, and `W_cn` is the link matrix from university i to j.

   (a) Compute PageRank with Google's hyperparameter $\alpha = 0.85$;

   (b) Compute HITS authority and hub ranking using SVD of the link matrix;

   (c) Compare these rankings against the research ranking (you may consider Kendall's τ distance – as the number of pairwise mismatches between two orders – to compare different rankings);

   (d) Compute extended PageRank with various hyperparameters $\alpha \in (0, 1)$, investigate its effect on ranking stability.

   For your reference, an implementation of PageRank and HITs can be found at

   `https://github.com/yao-lab/yao-lab.github.io/blob/master/data/pagerank.m`

2. *Perron Theorem:* Assume that $A > 0$. Consider the following optimization problem:

$$\begin{aligned} &\max \delta \\ s.t.\quad &Ax \geq \delta x \\ &x \geq 0 \\ &x \neq 0. \end{aligned}$$

   Let λ^* be optimal value with $\nu^* \geq 0, \quad 1^T \nu^* = 1$, and $A\nu^* \geq \lambda^* \nu^*$. Show that

   (a) $A\nu^* = \lambda^* \nu^*$, i.e. (λ^*, ν^*) is an eigenvalue-eigenvector pair of A;

   (b) $\nu^* > 0$;

   *(c) λ^* is unique and ν^* is unique;

   *(d) For other eigenvalue $\lambda \quad (\lambda z = Az \quad when \quad z \neq 0)$, $|\lambda| < \lambda^*$.

3. **Absorbing Markov Chain:*

   Let P be a row Markov matrix on $n+1$ states with non-absorbing state $\{1,\ldots,n\}$ and absorbing state $n+1$. Then P can be partitioned into

$$P = \begin{bmatrix} Q & R \\ 0 & 1 \end{bmatrix}$$

   Assume that Q is primitive. Let $N(i,j)$ be the expected number of jumps starting from nonabsorbent state i and hitting state j, before reaching the absorbing state $n+1$. Show that

   (a) $N(i,i) = 1 + \sum_k N(i,k)Q(k,i)$, for $i = 1,\ldots,n$;

   (b) $N(i,j) = \sum_k N(i,k)Q(k,j)$, for $i \neq j$;

   (c) These identities together imply that $N = (I-Q)^{-1}$, called the fundamental matrix;

   (d) Show that the probability of absorption from state i, $B(i)$ $(i = 1\ldots,n)$, is given by $B = NR$.

4. *Spectral Bipartition*: Consider the 374-by-475 matrix X of character-event for A Dream of Red Mansions, e.g. in the Matlab format

   `https://github.com/yuany-pku/dream-of-the-red-chamber/blob/master/HongLouMeng374.txt`

   with a readme file:

   `https://github.com/yuany-pku/dream-of-the-red-chamber/blob/master/README.md`

   Construct a weighted adjacency matrix for character-cooccurance network $A = XX^T$. Define the degree matrix $D = \mathrm{diag}(\sum_j A_{ij})$. Check if the graph is connected. If you are not familiar with this novel and would like to work on a different network, you may consider the Karate Club Network:

   `https://github.com/yao-lab/yao-lab.github.io/blob/master/data/karate.mat`

   that contains a 34-by-34 adjacency matrix.

   (a) Find the second smallest generalized eigenvector of $L = D - A$, i.e. $(D-A)f = \lambda_2 f$ where $\lambda_2 > 0$;

   (b) Sort the nodes (characters) according to the ascending order of f, such that $f_1 \leq f_2 \leq \ldots \leq f_n$, and construct the subset $S_i = \{1,\ldots,i\}$;

   (c) Find an optimal subset S^* such that the following is minimized

$$\alpha_f = \min_{S_i} \left\{ \frac{|\partial S_i|}{\min(|S_i|, |\bar{S}_i|)} \right\}$$

   where $|\partial S_i| = \sum_{x\sim y, x\in S_i, y\in \bar{S}_i} A_{xy}$ and $|S_i| = \sum_{x\in S_i} d_x = \sum_{x\in S_i, y} A_{xy}$.

   (d) Check if $\lambda_2 > \alpha_f$;

(e) Quite often people find a suboptimal cut by $S^+ = \{i : f_i \geq 0\}$ and $S^- = \{i : f_i < 0\}$. Compute its Cheeger ratio

$$h_{S^+} = \frac{|\partial S^+|}{\min(|S^+|, |S^-|)}$$

and compare it with α_f, λ_2.

(f) You may further recursively bipartite the subgraphs into two groups, which gives a recursive spectral bipartition.

5. *Degree Corrected Stochastic Block Model (DCSBM)*: A random graph is generated from a DCSBM with respect to partition $\Omega = \{\Omega_k : k = 1, \ldots, K\}$ if its adjacency matrix $A \in \{0,1\}^{N \times N}$ has the following expectation

$$\mathbb{E}[A] = \mathcal{A} = \Theta Z B Z^T \Theta$$

where $Z^{N\times k}$ has row vectors $\in \{0,1\}^K$ as the block membership function $z : V \to \Omega$,

$$z_{ik} = \begin{cases} 1, & i \in \Omega_k, \\ 0, & otherwise. \end{cases}$$

and $\Theta = \mathrm{diag}(\theta_i)$ is the expected degree satisfying,

$$\sum_{i \in \Omega_k} \theta_i = 1, \quad \forall k = 1, \ldots, K.$$

The following matlab codes simulate a DCSBM of nK nodes, written by Kaizheng Wang,

`https://github.com/yao-lab/yao-lab.github.io/blob/master/data/DCSBM.m`

Construct a DCSBM yourself, and simulate random graphs of 10 times. Then try to compare the following two spectral clustering methods in finding the K blocks (communities).

Alg. A [1] Compute the *top* K generalized eigenvector

$$(D - A)\phi_i = \lambda_i D \phi_i,$$

construct a K-dimensional embedding of V using $\Phi^{N\times K} = [\phi_1, \ldots, \phi_K]$;

[2] Run k-means algorithm (call `kmeans` in matlab) on Φ to find K clusters.

Alg. B [1] Compute the *bottom* K eigenvector of

$$\mathcal{L} = D^{-1/2}(D - A)D^{-1/2} = U \Lambda U^T,$$

construct an embedding of V using $U^{N\times K}$;

[2] Normalized the row vectors u_{i*} on to the sphere: $\hat{u}_{i*} = u_{i*}/\|u_{i*}\|$;

[3] Run k-means algorithm (call `kmeans` in matlab) on $\hat{U}$ to find K clusters.

You may run it multiple times with a stabler clustering. Suppose the estimated membership function is $\hat{z} : V \to \{1, \ldots, K\}$ in either methods. Compare the performance using mutual information between membership function z and estimate $\hat{z}$,

$$I(z, \hat{z}) = \sum_{s,t=1}^{K} Prob(z_i = s, \hat{z}_i = t) \log \frac{Prob(z_i = s, \hat{z}_i = t)}{Prob(z_i = s)Prob(\hat{z}_i = t)}. \quad (1)$$

For example,

`https://github.com/yao-lab/yao-lab.github.io/blob/master/data/NormalizedMI.m`

6. **Directed Graph Laplacian*: Consider the following dataset with Chinese (mainland) University Weblink during 12/2001-1/2002,

   `https://github.com/yao-lab/yao-lab.github.io/blob/master/data/univ_cn.mat`

   where `rank_cn` is the research ranking of universities in that year, `univ_cn` contains the webpages of universities, and `W_cn` is the link matrix from university i to j.

   Define a PageRank Markov Chain

   $$P = \alpha P_0 + (1 - \alpha)\frac{1}{n} ee^T, \quad \alpha = 0.85$$

   where $P_0 = D_{out}^{-1} A$. Let $\phi \in \mathbb{R}_+^n$ be the stationary distribution of P, i.e. PageRank vector. Define $\Phi = \mathrm{diag}(\phi_i) \in \mathbb{R}^{n \times n}$.

   (a) Construct the normalized directed Laplacian

   $$\vec{\mathcal{L}} = I - \frac{1}{2}(\Phi^{1/2} P \Phi^{-1/2} + \Phi^{-1/2} P^T \Phi^{1/2})$$

   (b) Use the second eigenvector of $\vec{\mathcal{L}}$ to bipartite the universities into two groups, and describe your algorithm in detail;

   (c) Try to explain your observation through directed graph Cheeger inequality.

7. **Chung's Short Proof of Cheeger's Inequality*:

   Chung's short proof is based on the fact that

   $$h_G = \inf_{f \neq 0} \sup_{c \in \mathbb{R}} \frac{\sum_{x \sim y} |f(x) - f(y)|}{\sum_x |f(x) - c| d_x} \quad (2)$$

   where the supreme over c is reached at $c^* \in median(f(x) : x \in V)$. Such a claim can be found in Theorem 2.9 in Chung's monograph, Spectral Graph Theory. In fact, Theorem 2.9

implies that the infimum above is reached at certain function f. From here,

$$\begin{aligned}
\lambda_1 &= R(f) = \sup_c \frac{\sum_{x\sim y}(f(x)-f(y))^2}{\sum_x (f(x)-c)^2 d_x}, && (3)\\
&\geq \frac{\sum_{x\sim y}(g(x)-g(y))^2}{\sum_x g(x)^2 d_x}, \quad g(x) = f(x) - c && (4)\\
&= \frac{(\sum_{x\sim y}(g(x)-g(y))^2)(\sum_{x\sim y}(g(x)+g(y))^2)}{(\sum_{x\in V} g^2(x)d_x)((\sum_{x\sim y}(g(x)+g(y))^2)} && (5)\\
&\geq \frac{(\sum_{x\sim y}|g^2(x)-g^2(y)|)^2}{(\sum_{x\in V} g^2(x)d_x)((\sum_{x\sim y}(g(x)+g(y))^2)}, \quad \text{Cauchy-Schwartz Inequality} && (6)\\
&\geq \frac{(\sum_{x\sim y}|g^2(x)-g^2(y)|)^2}{2(\sum_{x\in V} g^2(x)d_x)^2}, \quad (g(x)+g(y))^2 \leq 2(g^2(x)+g^2(y)) && (7)\\
&\geq \frac{h_G^2}{2}. && (8)
\end{aligned}$$

Is there any step wrong in the reasoning above? If yes, can you remedy it/them?