CS6841: Advanced Algorithms **IIT Madras, Spring 2016**
Lecture #7: Graph Partitioning February 5, 2016
Lecturer: Ravishankar Krishnaswamy Scribe: Anju Das.B

1 Application of SVD: Graph Partitioning using Stochastic Block Models

The stochastic block model is a generative model for random graphs. This model tends to produce graphs containing communities, subsets characterized by being connected with one another with particular edge densities. For example, edges may be more common within communities than between communities. SVD or Eigen vectors can be used to recover such hidden communities. Some applications of stochastic block models are in giving suggestions to users of Social Networking Sites and in recovering the structure of a graph. The objective of this lecture is to see whether we can partition the graph well.

The graph partition problem is defined on data represented in the form of a graph G = (V,E), with V vertices and E edges, such that it is possible to partition G into smaller components with specific properties. Here ,we are interested to partition the graph into two components with fewest edges crossing. The problem is similar to min-cut problem which can be solved in polynomial time. But min-cut problem doesn't reveal much about the graph partitioning.

So, a reasonable goal would be to check whether we can partition the graph into two equal sized pieces or rephrased it in another way, given a graph G, can we bisect G with fewest edges crossing? This is clearly an NP-Hard problem. Although there exists a $O(\log n)$ approximation using LP Rounding. By $O(\log n)$ approximation we mean , for all input graphs G, algorithm finds a bisection with $O(\log n)$ times crossing edges as the best bisection for the graph.

Can we design algorithm which models real world instances?

So, our problem is to perform bisection , i.e., to partition the graph into two communities of size $\frac{n}{2}$ each, where n is the number of nodes in the graph. The two communities should be disjoint.
Let's consider the graph to be consisting of equal number of students from CS and EE departments of IITM as shown in figure 7.1. Our aim is to bisect the graph so that the students are grouped to their respective partition with very low error.

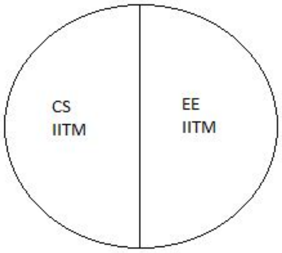


Figure 7.1:

1.1 Stochastic Block Model

In this model, we build a random graph that has an unknown community structure and generates the graph of friendship. The simplest model of this form is for the graph bisection problem. This is the problem of partitioning the vertices of a graph into two equal-sized sets while minimizing the number of edges bridging the sets. To create an instance of the planted bisection problem, we first choose a partition of the vertices into equal-sized sets X and Y . Then choose probabilities p ¿ q, and place edges between vertices with the following probabilities:

$$\mathbb{P}[(u,v)\epsilon E] = \begin{cases} p & \text{if } u \in X \text{ and } v \in X \\ p & \text{if } u \in Y \text{ and } v \in Y \\ q & \text{otherwise} \end{cases}$$

The expected number of edges crossing between X and Y will be $q|X||Y|$. If p is sufficiently larger than q, then every other bisection will have more crossing edges. If p is too close to q, then the partition given by X and Y will not be the smallest. In this lecture, we will show that this partition can be recovered from the second eigenvector of the adjacency matrix of the graph (See figure 7.2). Feed such an unlabelled graph to the algorithm and ask to recover the community upto permutation.

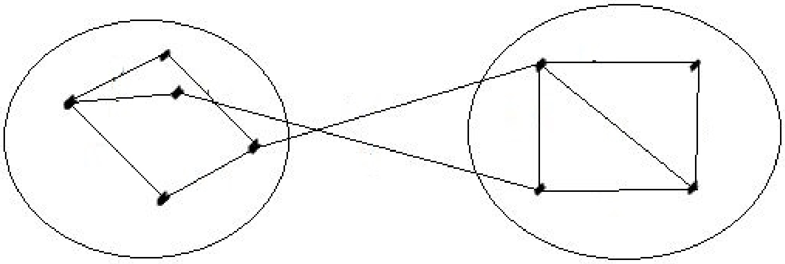


Figure 7.2:

1.1.1 Toy Case

We will consider the case $p = \frac{1}{2}$ and $q = \frac{1}{4}$ as the generation process.

$$\text{Expected degree of a vertex} = \frac{n}{2}.\frac{1}{2} + \frac{n}{2}.\frac{1}{4} = \frac{3n}{8} \tag{7.2}$$

$$\text{Variance of degree} = \frac{n}{2}p(1-p) + \frac{n}{2}q(1-q) \leq \frac{3n}{8} \qquad (\sigma \approx \sqrt{n}) \tag{7.3}$$

Using Bernstein's inequality,

$$\mathbb{P}[|Degree(u) - \mu| \geq t\sigma] \leq \exp(t^2) \tag{7.4}$$

From figure 7.3 it is clear that all the vertices have degree $\frac{3n}{8} \pm O(\sqrt{n \log n})$. $95\mathbb{P}[|X - \mu| \geq t\sigma] \leq$

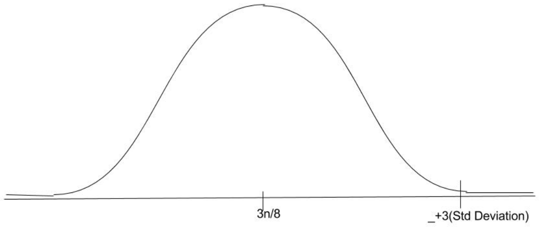


Figure 7.3:

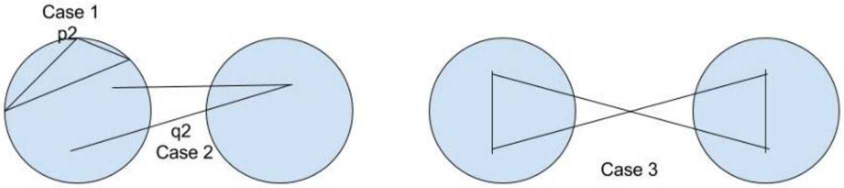


Figure 7.4:

$\exp(t^2)$(7.5)By applying union bound , set $t = 2\sqrt{\log n}$,

$$\mathbb{P}[|X - \mu| \geq t\sigma] \leq \exp(-4\log n) = \frac{1}{n^4} \tag{7.6}$$

Morally,

(i) if u and v were in the same block, then the expected number of common neighbours $= \frac{1}{2}np^2 + \frac{1}{2}nq^2$.

(ii) If u and v are in different blocks, then this value is npq.

$$E[\text{number of mutual friends}] = \{\text{u v are in same cluster}\} + \{\text{u v are in different clusters}\}$$

$$= \tfrac{n}{2}p^2 + \tfrac{n}{2}q^2 + npq$$

$$= \tfrac{5n}{32} + \tfrac{4n}{32}$$

$$= \tfrac{9n}{32}$$

(7.7)Let Var_1 and Var_2 be the variance of number of mutual friends when the nodes are in the same cluster and different clusters respectively.

$$Var_1 = (\tfrac{n}{2})^2p^2 + (\tfrac{n}{2})^2q^2 - (\tfrac{n}{2}p^2 + \tfrac{n}{2}q^2)^2$$

$$= \tfrac{n^2p^2}{4}(1-p^2) + \tfrac{n^2q^2}{4}(1-q^2) - \tfrac{n^2p^2q^2}{2}$$

$$Var_2 = n^2pq - n^2p^2q^2 = n^2pq(1-pq)$$

(7.8)We can approximate $(1-p^2) \approx 1$ and $(1-q^2) \approx 1$. Then Var_1 will become,

$$Var_1 = \frac{n^2}{4}(p^2+q^2)$$

$\therefore Var_1 > Var_2$(7.9)

Same concentration lets you know that for all pairs, the real count differs from estimate say e,

$$e \leq \pm C\sqrt{n \log n}$$

```
Algorithm

Step 1: Set cutoff = $\frac{5n}{32} - 6\sqrt{n \log n} = \zeta$
Step 2: If (u,v) have more than $\zeta$ mutual friends, then,
        they are in the same cluster
Step 3: else,
        u and v are in different clusters
```

The two drawbacks are:

1. Need large gap between p and q. ($Gap \geq \sigma^2$)
2. Algorithm is very tailored to given model.(Not robust)

1.2 Applying SVD for partitioning

To explain the algorithm, let us choose X = 1, . . . , n/2 and Y = n/2 + 1, . . . , n. Lets do this for simplicity.

Define the matrix

$$\mathrm{M} = \mathrm{E[A]} = \begin{bmatrix} p & . & . & . & p & q & . & . & . & q \\ . & & & & & . & & & & \\ . & & & & & . & & & & \\ . & & & & & . & & & & \\ p & . & . & . & p & q & . & . & . & q \\ q & . & . & . & q & p & . & . & . & p \\ . & & & & & . & & & & \\ . & & & & & . & & & & \\ . & & & & & . & & & & \\ q & . & . & . & q & p & . & . & . & p \end{bmatrix}$$

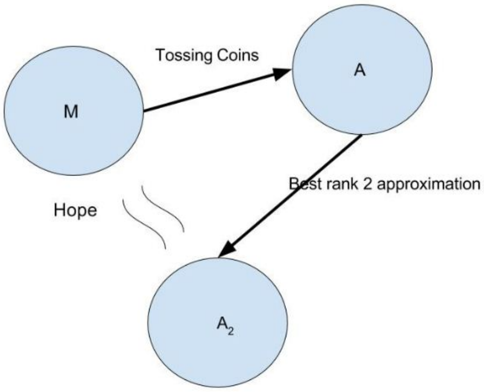
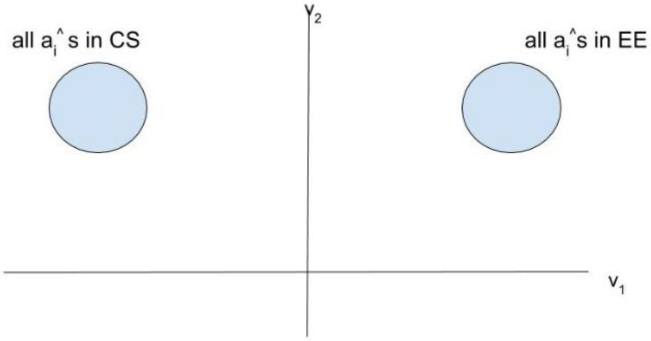


Figure 7.5:



The rank of matrix, M is 2 as there are only two different kinds of rows. G is a random term-by-term sample accounting M. SVD of A upto rank 2 is the best low-rank approximation of A. Intuitively, the reason this works is that A is a slight perturbation of M , and so the eigenvectors of A should look like the eigenvectors of M . See figure 7.5 which depicts the intuition of approximating A to A_2. Use A_2 and cluster these two-dimensional vectors.

$$\begin{bmatrix} ... & a_1^T & ... \\ ... & a_2^T & ... \\ ... & ... & ... \\ ... & ... & ... \\ ... & a_n^T & ... \end{bmatrix} \begin{bmatrix} . \\ v_1 \\ . \\ . \\ . \end{bmatrix} = \begin{bmatrix} < a_1, v_1 > \\ < a_2, v_1 > \\ . \\ . \\ < a_n, v_1 > \end{bmatrix} = < A, v_1 > \tag{7.10}$$

Similarly,

$$< A, v_2 >= \begin{bmatrix} < a_1, v_1 > \\ < a_2, v_1 > \\ . \\ . \\ < a_n, v_1 > \end{bmatrix} \tag{7.11}$$

Hence we can view $\hat{a_i}$ as,

$$\hat{a_i} \approx < a_i, v_1 > \hat{v_1} + < a_i, v_2 > \hat{v_2} \tag{7.12}$$

Cluster $\hat{a_i}s$ into two clusters.

1.3 Eigenvalue/Eigenvector Approach

To make the approach simpler, the matrix under consideration, M is a square ,symmetric matrix. The relation between M and its eigenvector(v) and eigenvalue(λ) is,

$$Mv = \lambda v$$

Compute eigenvectors and eigenvalues of M and let's hope that eigenvectors of A are close to those of M. Cluster the vectors using eigenvectors. Let's consider a 4X4 matrix, M. The rank of M is 2. Hence it has two λ,v pairs and they are:

$$M = \begin{bmatrix} p & p & q & q \\ p & p & q & q \\ q & q & p & p \\ q & q & p & p \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 2(p+q) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{7.13}$$

$$M = \begin{bmatrix} p & p & q & q \\ p & p & q & q \\ q & q & p & p \\ q & q & p & p \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} = 2(p-q) \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \tag{7.14}$$

Generally, the eigenvectors w_1, w_2 and their corresponding eigenvalues λ_1, λ_2 are:

$$w_1 = \begin{bmatrix} \frac{1}{\sqrt{n}} \\ \frac{1}{\sqrt{n}} \\ . \\ . \\ . \\ \frac{1}{\sqrt{n}} \end{bmatrix} ; \lambda_1 = \frac{n}{2}(p+q) \tag{7.15}$$

$$w_2 = \begin{bmatrix} \frac{1}{\sqrt{n}} \\ . \\ . \\ \frac{1}{\sqrt{n}} \\ -\frac{1}{\sqrt{n}} \\ . \\ . \\ -\frac{1}{\sqrt{n}} \end{bmatrix} ; \lambda_2 = \frac{n}{2}(p-q) \tag{7.16}$$

$$M = \lambda_1 w_1 w_1^T + \lambda_2 w_2 w_2^T \tag{7.17}$$

All other eigenvalues (viz., λ_3 and λ_4) are zeros. But, we don't know M, so we use matrix perturbation theorem. We compute eigenvectors of A in the hope that they also reveal similar information.

$$Av = \lambda v \tag{7.18}$$

$$A \begin{bmatrix} \frac{v_1}{\sqrt{n}} \\ \frac{v_2}{\sqrt{n}} \\ . \\ . \\ \frac{v_n}{\sqrt{n}} \end{bmatrix} = \lambda \begin{bmatrix} \frac{v_1}{\sqrt{n}} \\ \frac{v_2}{\sqrt{n}} \\ . \\ . \\ \frac{v_n}{\sqrt{n}} \end{bmatrix} \tag{7.19}$$

1.4 Matrix Perturbation Theory

Let M' be the perturbation of M. **Can relate eigenvectors of M and M'!**

Algorithm 1
Receive graph G, adjacency matrix A.
Compute w_1, w_2 of A.
Look at w_2:

- All coordinates > 0 corresponds to cluster CS.
- All coordinates ≤ 0 corresponds to the cluster EE.

Rank 2 approximation is mostly enough for $k = 2$ clusters.

Takeaway

1. For block models with 2 clusters, M has rank 2, which implies 2-SVD is sufficient for partitioning.
2. For block models with k clusters, M has rank k, which implies k-SVD is sufficient for partitioning.

Let us discuss the matrix perturbation approach. Recall the matrix M. Let A be a random element wise sample of M. Consider another matrix $\hat{A}$ for analysis purpose. $\hat{A}$ is defined as:

$$\hat{A} = A + pI$$

Eigen vectors of A and $\hat{A}$ are same.

$$A\vartheta = \lambda\vartheta \rightleftharpoons \hat{A}\vartheta = (\lambda + p)\vartheta$$
$$\text{Error matrix, } E = \hat{A} - M$$

The entries in E are defined as follows:
For all $a, b \in X$ or $a, b \in Y$,

$$E[a,b] = \begin{cases} 1-p, & \text{with probability } p \\ -p, & \text{with probability } 1-p \end{cases}$$

For a, b in different clusters, i.e., $a, b \in$ XxY

$$E[a,b] = \begin{cases} 1-q, & \text{with probability } q \\ -q & \text{with probability } 1-q \end{cases}$$

The matrix, E has zero mean. Random matrix concentrates well, like how the sum of independent random variables concentrate around mean. The norm of random matrix concentrates around the mean.

Matrix Concentration Theorem[1]

Theorem 7.1. $\|E\|^2 = max\ \ eigenvalue \le O(\sqrt{pn}) \qquad whp \to 1$
$p \ge \Omega(\frac{\log(n^4)}{n})$

Using the spectral norm given by,

$$\lambda_{max}(A) =_{\vartheta,\|\vartheta\|=1}^{max} V^T AV$$

and Theorem 7.1 ,

$$x^T Ex \le O(\sqrt{pn}) \quad \text{for all} x \in \mathbb{R}^n \quad \text{unit vector} \tag{7.20}$$

Applying **Chernoff bound**,
For fixed X,

$$x^T Ex = \sum E_{ij} X_i X_j$$
$$E[x^T Ex] = 0$$

We can use Chernoff bound whp $x^T Ex \le SMALL$.

To move from fixed x to all x cleverly union bound over small representative set.

The main result is,

$$\|E\| \le O(\sqrt{pn})$$

The High Level Idea:

We have to prove that,

$$w_2(\hat{A}) \approx w_2(A)$$

where w_2 is the second eigen vector of a matrix.

Matrix Perturbation Theorem[2](Davis-Kahan's theorem)

Theorem 7.2. *Let $\hat{A}$ and M be the symmetric matrices. Let $E = M - \hat{A}$. Let $\alpha_1 \geq ..... \geq \alpha_n$ be the eigenvalues of $\hat{A}$ with corresponding eigenvectors $v_1, ..., v_n$ and let ${}_{1n}$ be the eigenvalues of M with corresponding eigenvectors $w_1, ..., w_n$. Let ${}_i$ be the angle between v_i and w_i. Then,*

$$\sin\theta_i \leq \frac{2\|E\|}{\underset{j\neq i}{min}|\mu_i - \mu_j|}$$

The denominator of the inequality shows how isolated is the i^{th} eigenvalue in $\hat{A}$. We have to show that if θ is small, then the classifier using second eigenvector will not mis-classify too many points. we can rewrite the **Davis-Kahan's** theorem as follows:

Theorem 7.3. *Let θ_i be the angle between u_i and v_i.*
$\sin\theta_i \leq \frac{2\|A-M\|_2}{min(\mu_{i-1}-\mu_i, \mu_i-\mu_{i+1})}$

Definition:

$$\|A - M\|_2 = |\text{Largest eigen value of } A - M|$$

Why is the gap important?

If $\mu_i = \mu_{i-1}$, i.e., two eigenvectors with same eigenvalue, all eigenvectors in space is also eigenvector of same value. With high probability we can write,

Theorem 7.4. $\|E\| = \|A - M\|_2 \leq O(\sqrt{np}) \Leftrightarrow$ *for all* $x \in \mathbb{R}^n, \|x\| = 1$
$x^T E x \leq \sqrt{np}$

The proof needs union bound over all $x \in \mathbb{R}^n$. For fixed x, use union bound.

$$\begin{aligned}\sin\theta_2 &\leq \frac{2O(\sqrt{np})}{\Omega(n(p-q))} \\ &= O(\frac{\sqrt{p}}{\sqrt{n}(p-q)})\end{aligned}$$

very small for suitable p and q. Approximate $\sin\theta$ by θ.

$$\|w_2 - u_2\|_2 \leq r\theta \approx \theta$$

$$\text{whp} \quad \|w_2 - u_2\|_2 \leq O(\frac{\sqrt{p}}{\sqrt{n}(p-q)})$$

$$w_2 = \begin{bmatrix} \frac{1}{\sqrt{n}} \\ . \\ \frac{1}{\sqrt{n}} \\ -\frac{1}{\sqrt{n}} \\ . \\ -\frac{1}{\sqrt{n}} \end{bmatrix}$$

If our algorithm makes mistakes,

$$\|w_2 - u_2\|^2 \geq \tfrac{k}{n}$$
$$\sqrt{\tfrac{k}{n}} \leq \|w_2 - u_2\|_2 \underset{whp}{\leq} O(\tfrac{\sqrt{p}}{\sqrt{n}(p-q)})$$
$$\therefore \text{ number of mistakes} \leq O(\tfrac{p}{(p-q)^2})$$

Example:
Suppose $p = \frac{1}{2}$, $q = \frac{1}{2} - \frac{100}{\sqrt{n}}$

$$\text{Number of mistakes} \leq O(\tfrac{n}{10000})$$

Suppose $p = \frac{10}{n}$, $q = \frac{5}{n}$

$$\text{Average degree} = \tfrac{n}{2}.\tfrac{10}{n} + \tfrac{n}{2}.\tfrac{5}{n} = 7.5 \qquad \text{M is a sparse graph.}$$
$$\text{Number of mistakes} = \tfrac{p}{(p-q)^2} = \tfrac{10n^2}{25n} = \tfrac{2n}{5} \qquad \text{which is a small value.}$$

Number of mistakes are very small. Counting common vertices won't work.

2 References

1. ReVan Vu. Spectral norm of random matrices. Combinatorica, 27(6):721736, 2007.

2. Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. SIAM Journal on Numerical Analysis, 7(1):146, 1970.