ECEN5322: Search Engines and Analysis of High Dimensional Datasets 10/18/2015

Lab Report 3

Community Detection in Networks *Author: Prasanth Prahladan(100817764)*

We model a network comprising of n agents by using a $n \times n$ symmetric Adjacency Matrix, **A**, defined by

$$A = [a_{ij}] = \begin{cases} 1 & \text{if i,j are linked/friends} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Given a random realization A of a *Planted Partition G(n,p,q)*, the goal is to identify the two communities. Our ability to detect the partitions will decrease as $(p-q) \to 0$. Any $A \in G(n,p,q)$ can be generated by the following model,

$$A = TBT^T \tag{2}$$

$$B = \begin{bmatrix} B_{1,1}(p) & \cdots & B_{1,n/2}(p) & B_{1,n/2+1}(q) & \cdots & B_{1,n}(q) \\ \vdots & & \vdots & \vdots & & \vdots \\ B_{n/2,1}(p) & \cdots & B_{n/2,n/2}(p) & B_{n/2,n/2+1}(q) & \cdots & B_{n/2,n}(q) \\ B_{n/2+1,1}(q) & \cdots & B_{n/2+1,n/2}(q) & B_{n/2+1,n/2+1}(p) & \cdots & B_{n/2+1,n}(p) \\ \vdots & & \vdots & \vdots & & \vdots \\ B_{n,1}(q) & \cdots & B_{n,n/2}(q) & B_{n,n/2+1}(p) & \cdots & B_{n,n}(p) \end{bmatrix} \tag{3}$$

where, $B_{i,j}(p)$ are Bernoulli random variables and T is a Permutation Matrix.

For the analysis below, we make the following assumptions: we assume that we have an oracle that gives us access to T. Our algorithm will not require this assumption, though this assumption helps in the analysis. (wlog. we assume $T = I(n)$, the Identity matrix).

Our approach relies on the computation of the second dominant eigenvector of A.

1 The eigenvectors of the expected value of A

Q1. Prove that the expected adjacency matrix, $M = \mathbb{E}[A]$, has the form

$$M = \begin{bmatrix} p & \cdots & p & q & \cdots & q \\ \vdots & & \vdots & \vdots & & \vdots \\ p & \cdots & p & q & \cdots & q \\ q & \cdots & q & p & \cdots & p \\ \vdots & & \vdots & \vdots & & \vdots \\ q & \cdots & q & p & \cdots & p \end{bmatrix} \tag{4}$$

We assume that $T = I(n)$, and since $M = \mathbb{E}[A] \implies M = [m_{i,j}] = [\mathbb{E}a_{i,j}]$, where each $a_{i,j}$ is a Bernoulli random-variable described in (3)

$$m_{i,j} = \mathbb{E}[a_{i,j}] = \begin{cases} 1 \times p + 0 \times (1-p) & 1 \le i,j \le n/2 \text{ and } n/2+1 \le i,j \le n \\ 1 \times q + 0 \times (1-q) & 1 \le i \le n/2 \text{ and } n/2+1 \le j \le n \text{ and otherwise.} \end{cases} \tag{5}$$

Thus, we prove M has the above structure.

Q2. The degree matrix, is defined as the diagonal matrix with entries $d_i = \sum_{j=1}^{n} A_{ij}$. Derive the expression for the Expected Degree Matrix $\mathbb{E}[D]$.

From (3) with $T = I(n)$, we have $A = B$. Further, by definition of the Degree Matrix we have each d_i to be the sum along a row of the adjacency matrix

$$d_i = \sum_{j=1}^{n} A_{ij} = \frac{n}{2}p + \frac{n}{2}q = \frac{n}{2}(p+q). \tag{6}$$

Q3. Prove that the vector $w_1 = \frac{1}{\sqrt{n}}\mathbf{1}$ is an eigenvector of M. Determine the corresponding eigenvalue μ_1.
We observe the following for $v_1 = Mw_1 = M\frac{1}{\sqrt{n}}\mathbf{1}$

$$Mw_1 = \begin{bmatrix} p & \cdots & p & q & \cdots & q \\ \vdots & & \vdots & \vdots & & \vdots \\ p & \cdots & p & q & \cdots & q \\ q & \cdots & q & p & \cdots & p \\ \vdots & & \vdots & \vdots & & \vdots \\ q & \cdots & q & p & \cdots & p \end{bmatrix} \frac{1}{\sqrt{n}}\mathbf{1} \tag{7}$$

$$\begin{aligned} v_{1,i} &= \frac{1}{\sqrt{n}} \sum_{j=1}^{n} [p \cdots q]\mathbf{1} \\ &= \frac{1}{\sqrt{n}} \frac{n}{2}(p+q) \\ v_1 &= \frac{n}{2}(p+q)(\frac{1}{\sqrt{n}}\mathbf{1}) = \frac{n}{2}(p+q)w_1 = Mw_1. \end{aligned} \tag{8}$$

Therefore, w_1 is an eigenvector of M, with eigenvalue $\mu_1 = \frac{n(p+q)}{2}$.

Q4. Prove that the vector w_2 is an eigenvector of M, and determine the corresponding eigenvalue, when

$$w_2(i) = \frac{1}{\sqrt{n}} \begin{cases} 1 & \text{if } 1 \le i \le n/2 \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

We observe the following for $v_2 = Mw_2$

$$v_{2,i} = \sum_{j=1}^{n} [p \cdots q] w_{2,i} \tag{10}$$

$$= \frac{1}{\sqrt{n}} \frac{n}{2}(p-q)$$

$$v_2 = \frac{n}{2}(p-q)w_2 = Mw_2. \tag{11}$$

Therefore, w_2 is an eigenvector of M, with eigenvalue $\mu_2 = \frac{n(p-q)}{2}$.

Q5. Sketch the graph of the eigenvectors of M, w_3 and w_4, where

$$w_3(i) = \frac{1}{\sqrt{n}} \begin{cases} 1 & \text{if } 1 \le i \le n/4 \\ -1 & \text{if } n/4 < i \le 3n/4 \\ 1 & \text{if } 3n/4 < i < n \end{cases} \tag{12}$$

$$w_4(i) = \frac{1}{\sqrt{n}} \begin{cases} 1 & \text{if } 1 \le i \le n/4 \\ -1 & \text{if } n/4 < i \le 2n/4 \\ 1 & \text{if } 2n/4 < i \le 3n/4 \\ -1 & \text{if } 3n/4 < i < n. \end{cases} \tag{13}$$

Q6. Prove that w_3 and w_4 are in the Null Space of the matrix $\mathbb{E}[A]$.
By the structures of M, w_3, w_4 we observe that for $v_3 = Mw_3$ and $v_4 = Mw_4$

$$v_{3,1} = \frac{n}{4}p + \frac{n}{4}(-p) + \frac{n}{4}(-q) + \frac{n}{4}q = 0$$
$$v_{4,1} = \frac{n}{4}p + \frac{n}{4}(-p) + \frac{n}{4}q + \frac{n}{4}(-q) = 0$$

Similarly, we can prove $v_{3,i} = 0, v_{4,i} = 0 \forall i \leq n$. Therefore, $Mw_3 = 0\mathbf{1} = Mw_4$. Thus, by definition of Null Space, we have $\{w_3, w_4\} \in NullSpace(M)$.

Q7. Prove that

$$M = \mu_1 w_1 w_1^T + \mu_2 w_2 w_2^T \tag{14}$$

Considering the RHS and solving, we obtain

$$\begin{aligned} RHS = \mu_1 w_1 w_1^T + \mu_2 w_2 w_2^T &= (\frac{n(p+q)}{2})\frac{1}{n}\mathbf{1}\mathbf{1}^T + (\frac{n(p-q)}{2})w_2 w_2^T \\ &= \begin{bmatrix} [\frac{p+q}{2} + \frac{p-q}{2}] & [\frac{p+q}{2} - \frac{p-q}{2}] \\ [\frac{p+q}{2} - \frac{p-q}{2}] & [\frac{p+q}{2} + \frac{p-q}{2}] \end{bmatrix} \\ &= \begin{bmatrix} [p] & [q] \\ [q] & [p] \end{bmatrix} = M = LHS \end{aligned}$$

where $[\cdots]$ represents a block matrix of size $n/2 \times n/2$. Thus, we have proved the above result.

Q8. Describe a simple algorithm to recover two communities using the eigenvectors of M. Given the eigenvectors $\{w_1, w_2, w_3, w_4\}$ of M, we determine from the above results that

$$M = \mu_1 w_1 w_1^T + \mu_2 w_2 w_2^T$$
$$A = \gamma_1 w_1 w_1^T + \gamma_2 w_2 w_2^T + \gamma_3 w_3 w_3^T + \gamma_4 w_4 w_4^T$$

Using the components of w_2, all the indices corresponding to positive entries correspond to one component and the negative entries correspond to another component.

Q9. Prove that $\mathbb{E}[X] = 0$, where the expectation is computed over all possible realizations of the matrix B, with $A = M + X$ and X is the symmetric random matrix

$$x_{i,j} = \begin{cases} & \text{if } 1 \leq i \leq j \leq n/2 \text{ or } n/2 < i \leq j \leq n \\ (1-p) & \text{w.p. } p \\ -p & \text{w.p. } (1-p) \end{cases} \tag{15}$$

$$x_{i,j} = \begin{cases} & \text{if } 1 \leq i \leq n/2 \text{ and } n/2 < j \leq n \\ (1-q) & \text{w.p. } q \\ -q & \text{w.p. } (1-q) \end{cases} \tag{16}$$

From the definition of $X = [x_{i,j}]$ above, we obtain $\mathbb{E}[X] = [\mathbb{E}[x_i j]]$ as

$$\mathbb{E}[x_{i,j}] = \begin{cases} (1-p) \times p + (-p) \times (1-p) & \text{if } 1 \leq i \leq j \leq n/2 \text{ or } n/2 < i \leq j \leq n \\ (1-q) \times q + (-q) \times (1-q) & \text{if } 1 \leq i \leq n/2 \text{ and } n/2 < j \leq n \end{cases}$$
$$\mathbb{E}[x_{i,j}] = 0.$$

Therefore, we obtain $\mathbb{E}[X] = 0\mathbf{1}\mathbf{1}^T$.

2 Separating the dominant eigenvalues from the bulk

X is a symmetric random matrix with independent entries that have mean zero. One can show that the empirical spectral distribution converges towards a slightly modified form of the Wigner semi-circle law, given by

$$\frac{1}{\pi(p+q)}\sqrt{2n(p+q)-\lambda^2} \tag{17}$$

The dominant eigenvalues of A can be found from the decompositions of A to be

$$\lambda_1 = \frac{n}{2}(p+q)+1 \tag{18}$$

$$\lambda_2 = \frac{n}{2}(p-q)+\frac{p+q}{p-q} \tag{19}$$

The corresponding eigenvectors are w_1 and w_2. The remaining eigen-values are given by the semi-circle law.

Q10. Prove that λ_2 can be separated from the continuous "semi-circle" bulk, to detect the communities if

$$n(p-q) > \sqrt{2n(p+q)} \tag{20}$$

From (19) and considering the algebraic relationship (Arithmetic-Mean $\geq$ Geometric-Mean),

$$\begin{aligned}\lambda_2 &= \frac{1}{2}\left(n(p-q)+2\frac{p+q}{p-q}\right)\\ &= \text{ Arithmetic Mean } \{n(p-q), 2\frac{p+q}{p-q}\}\\ &>= \text{ Geometric Mean } \{n(p-q), 2\frac{p+q}{p-q}\} = \sqrt{2n(p+q)}\end{aligned}$$

$$\lambda_2 = \frac{1}{2}\left(n(p-q)+2\frac{p+q}{p-q}\right) \geq \sqrt{2n(p+q)}.$$

The above inequality always holds. However, for a strict inequality, we need to consider the following reasoning.

We note from (17) that for a finite number of eigenvalues to lie within the semi-circle, we require

$$2n(p+q)-\lambda^2 \geq 0 \implies \lambda \leq \sqrt{2n(p+q)}. \tag{21}$$

Hence, for λ_2 to lie outside the semi-circle we require

$$\lambda > \sqrt{2n(p+q)}$$

$$\lambda_2 = \left(\frac{n(p-q)}{2}+\frac{p+q}{p-q}\right) > \sqrt{2n(p+q)}.$$

Consider the relationship for the Arithmetic Mean of two numbers a, b being greater than c:

$$\text{A.M} = \frac{1}{2}(a+b) > c \implies a > c \text{ or } b > c$$

Using this result above, we get

$$\begin{aligned}\lambda_2 = \frac{1}{2}\left(n(p-q)+2\frac{p+q}{p-q}\right) &> \sqrt{2n(p+q)}\\ \implies n(p-q) &> \sqrt{2n(p+q)}.\end{aligned}$$

Q11. Derive the condition for separability of fully connected communities with

$$p = \frac{\alpha}{n}\log(n) \tag{22}$$
$$q = \frac{\beta}{n}\log(n) \tag{23}$$

With the condition being $n(p-q) > \sqrt{2n(p+q)}$ we substitute the values of p and q to obtain

$$n(p-q) = \log(n)(\alpha-\beta) > \sqrt{2\log(n)(\alpha+\beta)}$$
$$(\alpha-\beta) > \frac{2}{\sqrt{\log(n)}}\sqrt{\frac{(\alpha+\beta)}{2}}$$

Q12. Derive the condition for separability of non-connected communities with

$$p = \frac{a}{n} \tag{24}$$
$$q = \frac{b}{n} \tag{25}$$

With the condition being $n(p-q) > \sqrt{2n(p+q)}$ we substitute the values of p and q to obtain

$$n(p-q) = (a-b) > \sqrt{2(a+b)}$$
$$\frac{a-b}{2} > \sqrt{\frac{a+b}{2}}.$$

---

3 Experiments

3.1 Planted Partition Model

Q 13. Matlab function that takes p,q,n as input and generates the adjacency matrix of the planted partition model.

```
function [A, partitionIndicatorVec] = getPartitionGraphModel(n,p,q)
%{
Q 13.
n : total #nodes in G
p : probability of link between two vertices inside Cluster
q : probability of link edges between two vertices in opposite clusters
%}

if (mod(n,2) == 0 ) % n is EVEN
%     generate Permutation Matrix, T
    I = eye(n);
    ix = randperm (n);
    T = I(ix,:);

%     generate adjaceny matrix A of a planted partition over n nodes
    n2 = n/2;
    P = random('bino', 1, p, n2, n2); % upper left block
    dP2 = random('bino', 1, p, n2, 1); % diagonal of the lower right block
    Q = random('bino', 1, q, n2, n2); % upper right block
%     carve the two triangular and diagonal matrices that we need
    U = triu(P, 1);
```

```
    L = tril(P,-1);
    dP = diag(P);
    B0 = U + U' + diag(dP);
    B1 = Q;
    B2 = Q';
    B3 = L + L' + diag(dP2);
    B =[B0 B1;B2 B3];

    comm1 = ones(1,n2);
    originalCluster = [comm1 -1*comm1];


%  PERMUTE THE NODES
%     Re-index Nodes of the graph.
%     B*T' -> exchg columns; T*M -> exchg rows
    A = T*B*T';

%   Obtain the True_Cluster_NodeID for Graph A: Permute them
    partitionIndicatorVec   = originalCluster(ix);
%% -----------------------------------------------
else
   warning('n == ODD!')  ;
end
end
```

Q 14. Implementation of Partition Algorithm

```
function partitionIndicatorVec = runPartitionAlgo(A)
%{
Algorithm Partition
* compute the second dominant eigenvector, v2, of A, associated with the second largest
eigenvalue ?2.
* for i = 1 to n
if the coordinate i of v2 is positive,(v2)_i > 0, then  %what if its' ZERO?
assign node wi to community 1
else
assign node i to community 2.
end
end

partitionIndicatorVec : = {1(partition A), -1(partition B)}

%}


[V, D] = eigs(A,2);

vec = V(:,2)';
pos = (vec >0);
neg = (vec <0);
partitionIndicatorVec = pos - neg;

end
```

Q 15. Computing Overlap between the True-Partition and Predicted/Estimated-Partitions

$$\omega_i = \begin{cases} 1 & \text{if i belongs to partition 1} \\ -1 & \text{if i belongs to partition 2} \end{cases} \tag{26}$$

$$\tilde{\omega_i} = \begin{cases} 1 & \text{if } (v_2)_i > 0, \\ -1 & \text{otherwise} \end{cases} \tag{27}$$

$$rawoverlap = max\left(\sum_{i=1}^{n} \delta_{\omega_i,\tilde{\omega_i}}, \sum_{i=1}^{n} \delta_{-\omega_i,\tilde{\omega_i}}\right) \tag{28}$$

$$overlap = \frac{2}{n} rawoverlap - 1 \tag{29}$$

(a) Compute overlap score when $\tilde{\omega} = \omega$.

From (28) we obtain $rawoverlap = n$ when $\tilde{\omega} = \omega$. Substituting into (29), we obtain

$$overlap = \frac{2}{n}(n) - 1 = 1$$

(b) Prove that a random guess for the detection of the communities returns overlap 0.
A random guess for the detection of communities is a Binary vector of $\{-1, 1\}^n$ with each component chosen with equal probability (0.5). From the definition (28), we obtain $rawoverlap = \frac{n}{2}$. Thus, we obtain

$$overlap = \frac{2}{n}\frac{n}{2} - 1 = 0.$$

```
function  overlap = getPartitionOverlap(w1, w2)
%{
Q15.
Derive the Overlap Metric based on the
w1 : true partition vector
w2 : estimated partition vector
%}
n = length(w1);
del_w1_w2 = sum(w1 == w2);
del_minus_w1_w2 = sum(-w1 == w2);

rawoverlap = max(del_w1_w2, del_minus_w1_w2);

% random choice of w2 generates a non-zero overlap. Accounting for this:
overlap = (2/n)*rawoverlap - 1;          % Interpret: Prob. of successfully detecting
    communities.
end
```

Q 16/17. Dense and Sparse Communities

```
n = 300;
numTrials = 20;
alphaMax = 70;
betaMax = 50;
alphaMin = 5;
betaMin = 1;

Alpha = alphaMin:1:alphaMax;
Beta = betaMin:1:betaMax;

overlapMatrix = zeros(length(Alpha), length(Beta));

for i=1:length(Alpha)
    for j= 1:length(Beta)

        alpha = Alpha(i);
```

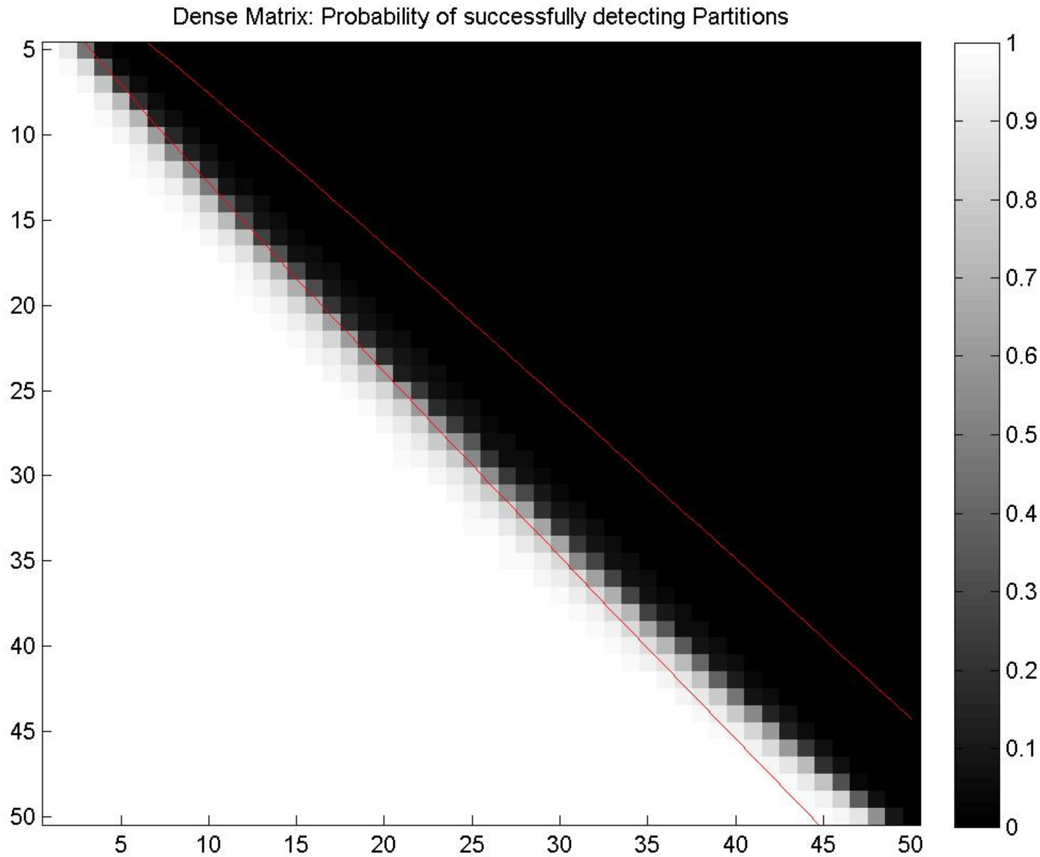

Figure 1: Dense Network: Probability of successfully detecting the partitions using the Partition-Algorithm. The decision boundaries are overlaid in red. The community-recovery algorithm is implemented only for case where $0 \leq q < p \leq 1 <$. The decision boundary lying inside the Dark region has (α, β) values that violate the $q < p$ requirement, and hence, can be ignored.

```
        beta = Beta(j);

%%          % Dense Matrix
%%          p = alpha/n*log(n);
%%          q = beta/n*log(n);

        % Sparse Matrix
        p = alpha/n;
        q = beta/n;

        overlapScore = zeros(1,numTrials);
        % ---------------------------------------------------------------
        % We need to run the algorithm only if q<p
        % We set the default values to zero!
        if (q<p)
         for iter = 1:numTrials
             [A, w] = getPartitionGraphModel(n,p,q);
             w_pred = runPartitionAlgo(A);
             overlapScore(iter) = getPartitionOverlap(w, w_pred);
         end
        end
```

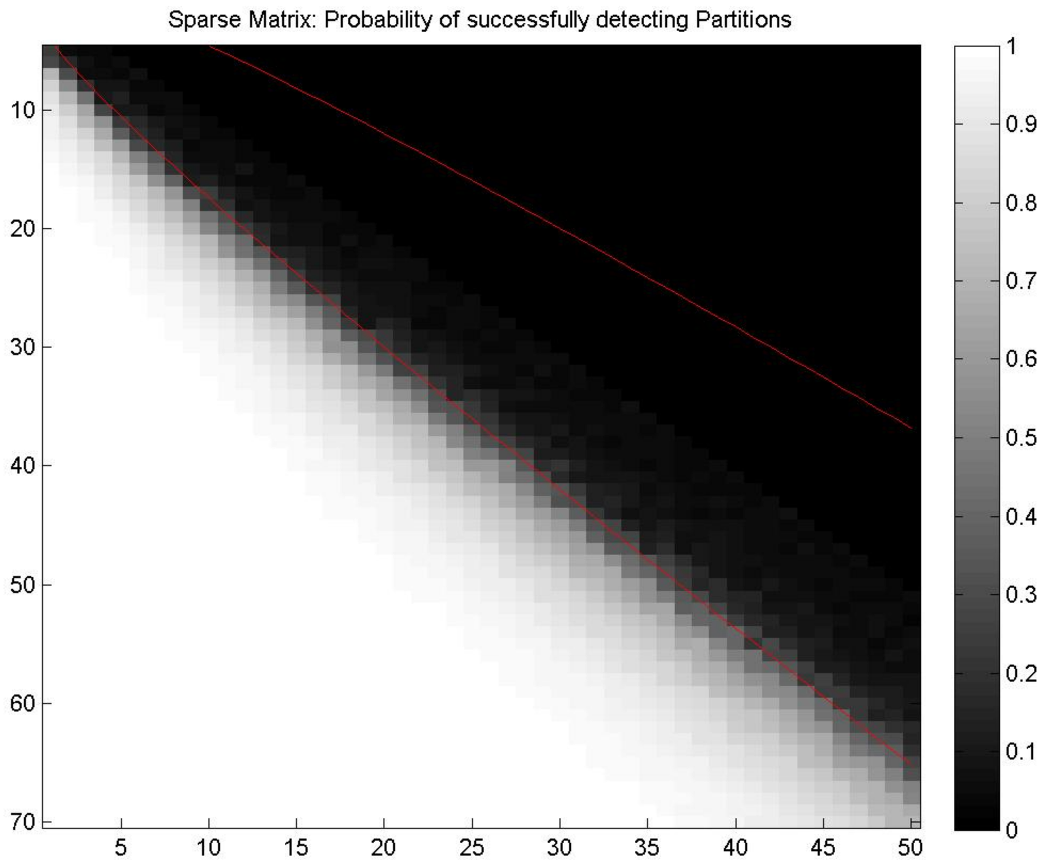

Figure 2: Sparse Network: Probability of successfully detecting the partitions using the Partition-Algorithm. The decision boundaries are overlaid in red. The community-recovery algorithm is implemented only for case where $0 \leq q < p \leq 1 <$. The decision boundary lying inside the Dark region has (a, b) values that violate the $q < p$ requirement, and hence, can be ignored.

```
        % ----------------------------------------------------------------
         % store the avg. Overlap score for given (alpha,beta)
         overlapMatrix(i,j) = sum(overlapScore)/ numTrials;
    end
end
%%%
%%%% ----------------------------------------------------------------
%%%% Plot the following curve on the
%%%% alpha - beta > sqrt(0.5*(alpha + beta))*2 / sqrt(log(n))
%%%% k = 2/sqrt(log n)
%%%% alpha > 0.5*( (2*beta + k^2/2)\pm k/2 sqrt(16*beta + k^2/2)    )
%%%
%%% k = 2/sqrt(log(n));
%%% Alpha1 = 0.5*((2*Beta + k^2/2) + (k/2)*sqrt(16*Beta + k^2));
%%% Alpha2 = 0.5*((2*Beta + k^2/2) - (k/2)*sqrt(16*Beta + k^2));
%%%% ----------------------------------------------------------------

% ----------------------------------------------------------------
% How does this change for the sparse matrix?
% SPARSE MATRIX BOUNDARIES
Alpha1 = (Beta + 1) + sqrt(1+ 4*Beta);
```

```
Alpha2 = (Beta + 1) - sqrt(1+ 4*Beta);

% ------------------------------------------------------------
close all
fig1 = figure(1)
% Plot the image as a GRAYSCALE
betaDim = [betaMin betaMax];
alphaDim = [alphaMin alphaMax];
img = imagesc(betaDim, alphaDim, overlapMatrix);
colormap(gray);
colorbar;
hold on
ax2 = plot(Beta,Alpha1,'r');
% colormap(ax2, parula )
hold on
ax3 = plot(Beta,Alpha2,'r');
% colormap(ax3, parula )
hold off
title('Sparse Matrix: Probability of successfully detecting Partitions')
% title('Dense Matrix: Probability of successfully detecting Partitions')
```

Q 18. Zachary's Karate Club

Overlap Score = 1. The code used for implementing the same is described below.

```
load zachary.mat

% From the question/visual graph: Identify true_Partition
nodeIDs = 1:34;
idx_teamA = [25 26 28 32 24 29 30 27 10 34 9 21 33 31 19 23 15 16]
idx_teamB = setdiff(1:34,idx_teamA)

truePartition = ones(size(nodeIDs));
truePartition(idx_teamB) = -1*ones(size(idx_teamB));


% Implement the algorithm to detect the partitions.

M = (A ~= 0 );
adjMatrix = zeros(size(A));
adjMatrix(M) = 1;

estPartition = runPartitionAlgo(adjMatrix);
cluster1 = (estPartition <0);
cluster1_idx_est = cluster1.*nodeIDs;

cluster2 = (estPartition >0);
cluster2_idx_est = cluster2.*nodeIDs;

overlapScore = getPartitionOverlap(truePartition, estPartition)
```

Definition 1. *Planted Partition G(n,p,q) The* Planted Partition G(n,p,q) *is the set of symmetric Adjacency matrices, A representing the network of n nodes (wlog. n = even). Randomly divide the nodes into equal sets of size n. Each set represents one community. For any two node pairs in a community, a edge exists between them with probability p. For any two node pairs, with nodes in opposite communities, an edge exists between them with a probability q, such that $0 \leq q < p \leq 1$. Self-loops can exist in these graphs. Note that the graphs are undirected, and hence require $a_{i,j} = a_{j,i}$.*