Classification on SBM by Optimization Approach

October 12, 2014

Abstract

Both ASE (fast) and EBSBM (accurate relatively) can help us do the classification on SBM. Now we try to use the optimization approach instead, which might be much faster than EBSBM and more accurate than ASE hopefully.

1 Problem Description

We consider the stochastic blockmodel (K blocks) as a random dot product graph model here.

Let $\mathcal{X} \subset \mathbb{R}^d$ be a set such that $x, y \in \mathcal{X}$ implies $\langle x, y \rangle \in [0, 1]$. Let the probability vector $\rho \in (0, 1)^K$ satisfies $\sum_{k=1}^K \rho_k = 1$ and the distinct laten positions are represented by $\nu = [\nu_1 | \cdots | \nu_K]^T \in \mathbb{R}^{K \times d}$, where $\nu_k \in \mathcal{X}$ for $1 \leq k \leq K$. Then $X_i \overset{iid}{\sim} \sum_k \rho_k \delta_{\nu_k}$ is the latent positions in the SBM o RDPG model. In this setting, the block memberships $\tau_1, \cdots, \tau_n | K, \rho \overset{iid}{\sim} \text{Discrete}([K], \rho)$ such that $\tau_i = \tau_j$ if and only if $X_i = X_j$.

Given the latent positions, the adjacency matrix A of a random graph satisfies $A_{ij} \overset{iid}{\sim} \text{Bern}(\langle X_i, X_j \rangle)$. And our goal is to assign vertices to their correct blocks.

2 Optimization Problem

Assume the true parameters are X^*, ν^*, τ^*. Let $P = X^* X^{*T}$ be the probability matrix. Then we want to solve the following **equivalent** optimization problem:

$$\begin{array}{ll} \underset{X,\nu,\tau}{\text{minimize}} & \|P - XX^T\| \\ \text{subject to} & X_i = \nu_{\tau_i},\ i = 1, \ldots, n, \\ & \nu_k \in \mathcal{X},\ k = 1, \ldots, K, \\ & \tau_i \in \{1, \cdots, K\},\ i = 1, \ldots, n. \end{array}$$

Here we do not need to specify the norm type in the objective function. Since $X = X^*$ is part of the solution which leads the objective function to 0, any matrix norm would be good here.

But P is unkown. Instead we are given only one observation A which has the property $E[A] = P$. So we consider a **similar** optimization problem:

$$\begin{aligned} \underset{X,\nu,\tau}{\text{minimize}} \quad & \|A - XX^T\|_F \\ \text{subject to} \quad & X_i = \nu_{\tau_i},\ i = 1, \ldots, n, \\ & \nu_k \in \mathcal{X},\ k = 1, \ldots, K, \\ & \tau_i \in \{1, \cdots, K\},\ i = 1, \ldots, n. \end{aligned} \tag{1}$$

We use Frobenius norm since each element in the matrix should be treated equally. Maybe we need formal proof here?

Here we can see a possible advantage of this optimization method over the EBSBM: If we have multiple observations $A_1, \cdots, A_m$, we could easily take better use of them by optimize on the average graph $\bar{A} = \frac{1}{m}\sum_{i=1}^m A_i$. However, for EBSBM, it is not trivial to combine the results.

We previously agreed on the following relaxation:

$$\begin{aligned} \underset{X,\nu,\tau}{\text{minimize}} \quad & \|A - XX^T\|_F^2 + \lambda \sum_{i=1}^n \|X_i - \nu_{\tau_i}\|_2^2 \\ \text{subject to} \quad & X_i \in \mathcal{X},\ i = 1, \ldots, n, \\ & \nu_k \in \mathcal{X},\ k = 1, \ldots, K, \\ & \tau_i \in \{1, \cdots, K\},\ i = 1, \ldots, n. \end{aligned} \tag{2}$$

Remark: When we let λ go to infinity in Problem 2, we have exactly Problem 1.

What metric should be used in the pentalty term? Should we use the norm or the square of the norm? Here I use square of the 2-norm since we could apply k-means algorithm directly (explain later). But these do NOT explain why this is a good optimization problem related to the original one.

Claim: If τ are the same in the solutions to problem (1) and problem (2), then the two optimization problems are equivalent.
Proof: Assume $\{X, \nu, \tau\}$ is solution to problem (1) and $\{\hat{X}, \hat{\nu}, \tau\}$ is solution to problem (2). By changing $\hat{\nu}$ to ν, we can see $\{\hat{X}, \nu, \tau\}$ leads to a smaller or equal objective function value in problem (2), since the penalty term becomes zero and $\nu_k \in \mathcal{X}$ ensured by $\hat{X}_i \in \mathcal{X}$.

3 Algorithm

Algorithm 1 Basic Algorithm

```
1: Given the adjacency matrix $A$ of graph $G$;
2: Obtain adjacency spectral embedding $\widehat{X}$ and related $\widehat{\nu}$ and $\widehat{\tau}$;
3: Initialize $X^{(0)} = \widehat{X}$, $\nu^{(0)} = \widehat{\nu}$, $\tau^{(0)} = \widehat{\tau}$;
4: Choose a proper $\lambda$ (more exploration on this);
5: repeat
6:     At iteration $h$;
7:     Fix $\nu^{(h-1)}$ and $\tau^{(h-1)}$, use $X^{(h-1)}$ as initial value and get:
           $$X^{(h)} = \underset{X}{\operatorname{argmin}} \; \|A - XX^T\|_F^2 + \lambda \sum_{i=1}^{n} \|X_i - \nu^{(h-1)}_{\tau_i^{(h-1)}}\|_2^2 \quad \text{subject to} \; X_i \in \mathcal{X},\ i = 1, \dots, n. \qquad (3)$$
8:     Fix $X^{(h)}$, apply $K$-means algorithm on $X^{(h)}$ and get component means
       $\nu^{(h)}$ and labels $\tau^{(h)}$;
9: until Converge
```

In Step 8 of Algorithm 1, we should fix $X^{(h)}$ and solve:

$$\begin{aligned} \{\nu^{(h)}, \tau^{(h)}\} = \underset{\nu,\tau}{\operatorname{argmin}} \quad & \sum_{i=1}^{n} \|X_i^{(h)} - \nu_{\tau_i}\|_2^2 \\ \text{subject to} \quad & \nu_k \in \mathcal{X},\ k = 1, \dots, K, \\ & \tau_i \in \{1, \cdots, K\},\ i = 1, \dots, n. \end{aligned} \tag{4}$$

We can see it is the goal of k-means algorithm with extra constraints $\nu_k \in \mathcal{X}$. We know $X_i^{(h)} \in \mathcal{X}$, and in k-means algorithm ν_k is the average of some collection of $X_i^{(h)}$. Combined with the definition of $\mathcal{X}$, we know $\nu_k \in \mathcal{X}$ in each iteration. So the extra constraints are satisfied throughout the algorithm.

4 Improvement

In Step 7 of Algorithm 1, we need $O(n^2)$ constraints since we need to check if the inner product of two latent positions X_i and X_j is between 0 and 1, which is very slow.

However, we only need to know the relationship between the latent positions, which means they are variant according to rotations. So we can see that:

- X_i should be inside the unit hypersphere.

  If $X_i \in \mathcal{X}$, we know that $\langle X_i, X_i \rangle = \|X_i\|_2^2 \leq 1$.

- We can assume all X_i are inside the first quadrant (all the elements are non-negative).

If $X_i \in \mathcal{X}$, they should be sticking together inside a quadrant-like area. Since the latent positions are invariant according to rotations, we can always rotate them to the first quadrant.

Under this assumption, we need to rotate the adjacency spectral embedding $\widehat{X}$ to the first quadrant before using it as the initial.

So the algorithm becomes:

Algorithm 2 Improved Algorithm

```
1: Given the adjacency matrix $A$ of graph $G$;
2: Obtain adjacency spectral embedding $\widehat{X}$ and related $\widehat{\nu}$ and $\widehat{\tau}$;
3: Find the “best” ratation matrix $R$ such that all $n$ vertices represented by $\widehat{X} \cdot R$ are in the first quadrant.
4: Initialize $X^{(0)} = \widehat{X} \cdot R$, $\nu^{(0)} = \widehat{\nu} \cdot R$, $\tau^{(0)} = \widehat{\tau}$;
5: Choose a proper $\lambda$ (more exploration on this);
6: repeat
7:     At iteration $h$;
8:     Fix $\nu^{(h-1)}$ and $\tau^{(h-1)}$, use $X^{(h-1)}$ as initial value and get:
```

$$\begin{aligned} X^{(h)} = \underset{X}{\operatorname{argmin}} \quad & \|A - XX^T\|_F^2 + \lambda \sum_{i=1}^{n} \|X_i - \nu^{(h-1)}_{\tau_i^{(h-1)}}\|_2^2 \\ \text{subject to} \quad & \|X_i\|_2 \leq 1,\ i = 1, \ldots, n, \\ & X_i \geq 0 \text{ in element},\ i = 1, \ldots, n. \end{aligned} \tag{5}$$

```
9:     Fix $X^{(h)}$, apply $K$-means algorithm on $X^{(h)}$ and get component means $\nu^{(h)}$ and labels $\tau^{(h)}$;
10: until Converge
```

In Step 8 of Algorithm 2, without pairwise comparing, we only need $O(n)$ constraints, which is better than $O(n^2)$ in Algorithm 1.

5 Simulation Results

Consider the 3-block SBM parameterized by

$$B = \begin{bmatrix} 0.6 & 0.4 & 0.4 \\ 0.4 & 0.6 & 0.4 \\ 0.4 & 0.4 & 0.6 \end{bmatrix} \quad \text{and} \quad \rho = [1/3, 1/3, 1/3].$$

In this experiment, we assume that $d = 3$ and $K = 3$ are known.

5.1 $n = 150$ Case

We run 1000 simulations on 150 vertices. For optimization algorithm, we use $\lambda = 1$ as the parameter for penalty term. Table 1 and Figure 1 displays error rate estimates for both algorithms. We can see that Optimization algorithm yields results superior to the ASGE algorithm. Based on the paired samples, the sign test p-value is less than 10^{-10}. In addition, Figure 2 presents the histogram

Algorithm	Mean	95% CI	Median
ASGE	0.2280	[0.2199, 0.2362]	0.1933
Optimization	0.0882	[0.0860, 0.0904]	0.0867

Table 1: Error rates of two algorithms.

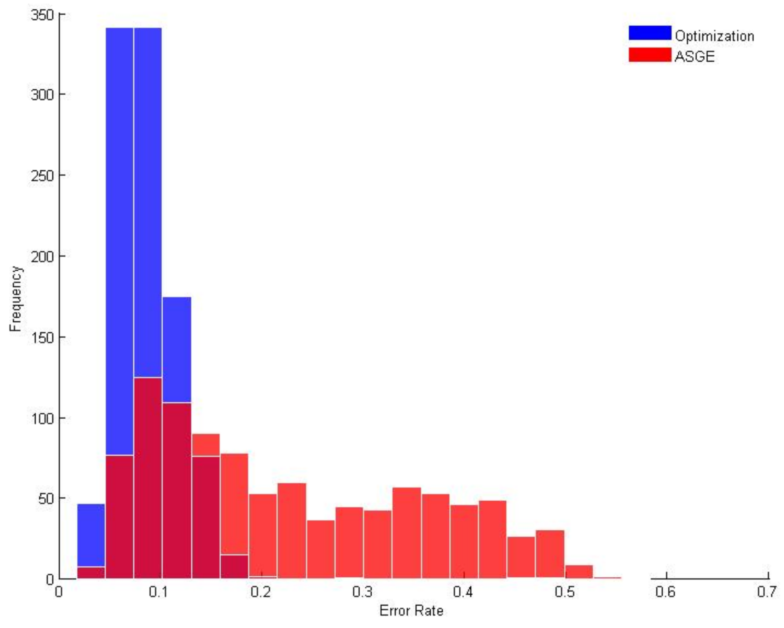

Figure 1: Histogram of error rates of two algorithms.

of the differential number of errors made by optimization and ASGE. It shows that for most graphs (972 out of 1000), optimization algorithm performs as well as or better than ASGE.

For now, Step 8 of Algorithm 2 **always fails** converging. And the results shown above are using the imperfect result without converging.

5.2 $n = 300$ Case

We run 1000 simulations on 300 vertices. For optimization algorithm, we use $\lambda = 1$ as the parameter for penalty term. Table 2 and Figure 3 displays error rate estimates for both algorithms. We can see that Optimization algorithm yields results superior to the ASGE algorithm. Based on the paired samples, the sign test p-value is less than 10^{-10}. In addition, Figure 4 presents the histogram of the differential number of errors made by optimization and ASGE. It shows that for most graphs (972 out of 1000), optimization algorithm performs as well as or better than ASGE.

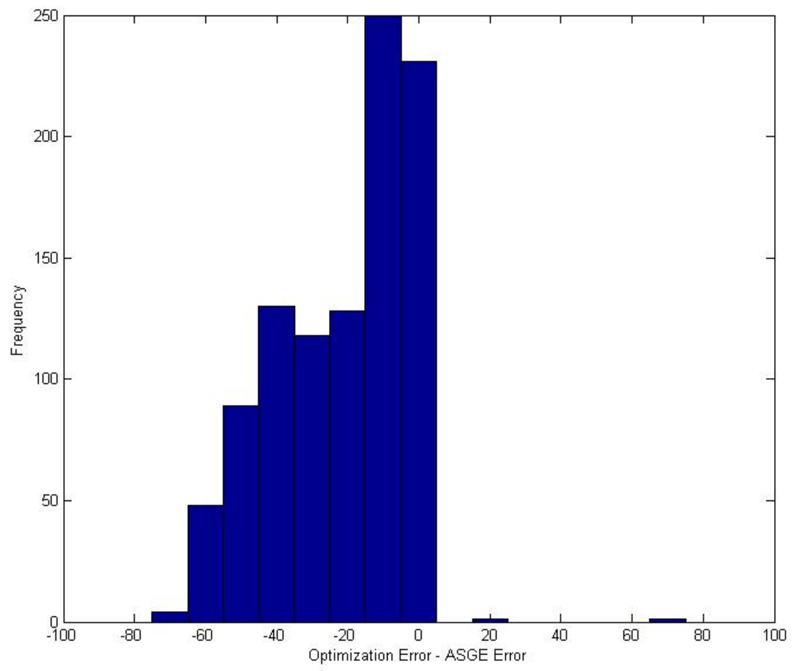

Figure 2: Histogram of the differential number of errors of two algorithms.

Algorithm	Mean	95% CI	Median
ASGE	0.0107	[0.0103, 0.0111]	0.0100
Optimization	0.0122	[0.0098, 0.0147]	0.0067

Table 2: Error rates of two algorithms.

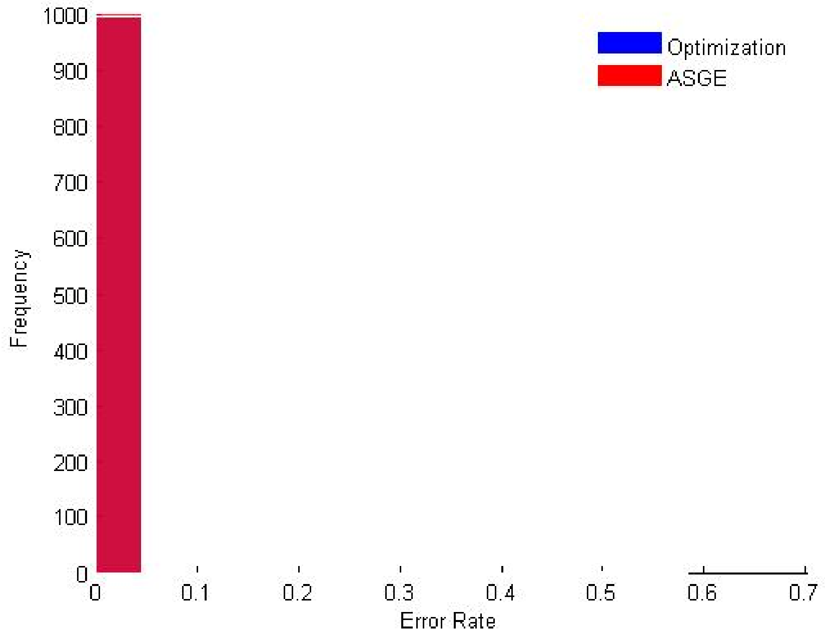

Figure 3: Histogram of error rates of two algorithms.

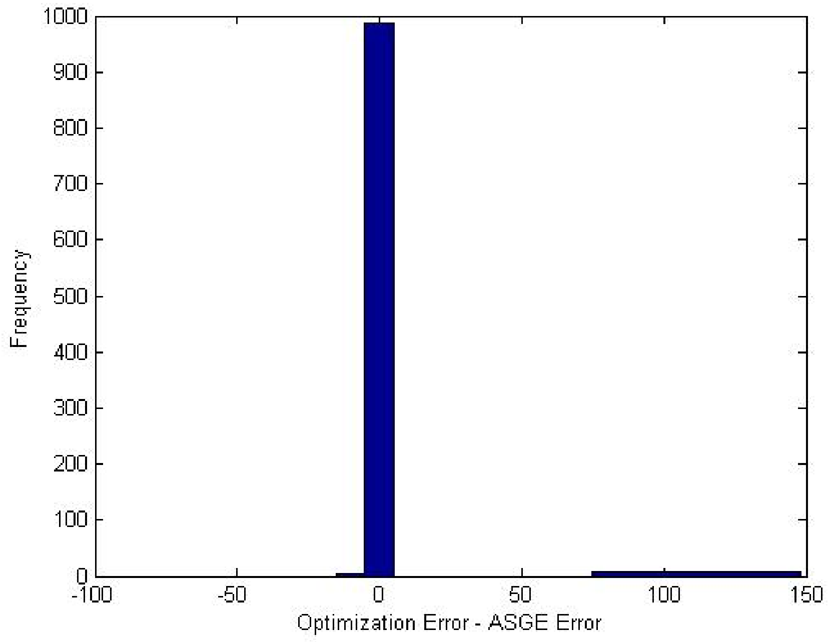

Figure 4: Histogram of the differential number of errors of two algorithms.

6 To be continue

Initialization

In Step 4 of Algorithm 2, we don't specify how to find the rotation matrix R. For now, we are solving the following optimization problem:

$$\begin{aligned} \underset{R}{\text{maximize}} \quad & \min_{i,j} \left(\widehat{X}R\right)_{ij} \\ \text{subject to} \quad & RR^T = I, \\ & \|\widehat{X}_i R\|_2 \leq 1,\ i = 1, \ldots, n, \\ & \widehat{X}_i R \geq 0 \text{ in element},\ i = 1, \ldots, n. \end{aligned} \tag{6}$$

In the objective function, $\min_{i,j} \left(\widehat{X}R\right)_{ij}$ is the minimal distance between rotated vector and the boundary of the first quadrant. We want to maximize the minimal distance, which is how "best" is defined in this situation. However, I believe there is a better way for this step.

Objective Function

Until now, our objective function is: $X^{(h)} = \text{argmin}_X \|A - XX^T\|_F^2 + \lambda \sum_{i=1}^n \|X_i - \nu^{(h-1)}_{\tau_i^{(h-1)}}\|_2^2$. By observation, we can change the penalty term, which is the sum of vector norms, to a Frobenius norm: $X^{(h)} = \text{argmin}_X \|A - XX^T\|_F^2 + \lambda \|X - \nu^{(h-1)}_{\tau^{(h-1)}}\|_F^2$, where $\nu^{(h-1)}_{\tau^{(h-1)}}$ is the notation for $[\nu^{(h-1)}_{\tau_1^{(h-1)}}, \cdots, \nu^{(h-1)}_{\tau_n^{(h-1)}}]^T$.

One possible benefit of the above formula for objective function is: We might be able to take the advantage of the relationship between the Frobenius norm and the trace. For example:
$\|A - XX^T\|_F^2 + \lambda \|X - \nu_\tau\|_F^2$
$= \text{Tr}((A - XX^T)^T(A - XX^T)) + \lambda \text{Tr}((X - \nu_\tau)^T(X - \nu_\tau))$
$= \text{Tr}(A^T A) - 2\text{Tr}(AXX^T) + \text{Tr}(X^T XX^T X) + \lambda \text{Tr}((X - \nu_\tau)^T(X - \nu_\tau))$
$= \cdots$