EECS 598: Final report
Community detection in multilayer graphs
using spectral methods and core-finding

Yutong Wang Dejiao Zhang
yutongw, dejiao@umich.edu

Abstract—This report focuses on the problem of community detection in both the static and dynamic setting. In the static setting, we investigate a spectral method based on clustering the outer product of the top $K$ singular vectors of the (weighted) adjacency matrices of a graph. We derive a misclassification rate bound for the case of the hidden bipartition problem. In the dynamic setting where cluster membership do not change, we consider the setting in [Han et al., 2015] in which a dynamic stochastic blockmodel whose underlying edge probability is drawn from a stationary ergodic process is studied. We prove a finite-time bound on the misclassification rate. In the dynamic stochastic blockmodel where the cluster membership do change, we propose a heuristic algorithm inspired by Belief Propagation and identifying core communities. We discuss the performance of the algorithm on some synthetic data set where the cluster membership changes as a Markov chain [Matias and Miele, 2017].

I. INTRODUCTION

Given graph $G = (V, E)$ where $V = \{1, \ldots, n\}$ and $K$ positive integer, the problem of community detection asks for a $K$-partition of $V$, say $C = \{C_1, \ldots, C_K\}$, that conforms to the intuitive notion of a community. For instance, it is reasonable to expect that there should be more edges within a cluster $C_i$ and fewer edges between distinct clusters $C_i, C_j$. This is indeed the underlying idea of many community detection algorithm. Spectral clustering for community detection is one such algorithm and has recently been shown to have desirable theoretical properties [Chin et al., 2015].

Now, suppose we are given graphs $G^t = (V, E^t)$ over the same vertex set for each $t = 1, \ldots, T$. We can view $t$ as time and $G^t$ as a time-varying relationship. Alternatively, one can view $t$ as different type of interactions between the nodes of $V$. For instance, $V$ may represent people and $G^t$ represents the relationship graph for various types of relationships, e.g, friends, family, coworkers. The sequence of graphs $\{G^t\}$ is sometimes referred to as a multilayer graph in the literature. For this report, we think of $t$ as time.

As in the static setting, the community detection in the dynamic setting seeks to partition the nodes into cohesive communities. However, we must account for the fact that communities may evolve in time and try to infer a time series of partitions $C^t$ for $t = 1, \ldots, T$.

One simple yet interesting random graph model for studying community detection algorithm is the stochastic blockmodel. In the stochastic blockmodel, the cluster membership $C$ is a latent random variable that is used to generate the observed adjacency matrix $A$. As such, researchers have studied the necessary and sufficient conditions for when it is possible to recover $C$ from the observation $A$.

Since $A$ can be thought of a noise-perturbed version of low-rank matrices, spectral methods are often applied to to study community detection in stochastic blockmodel. This has generated a wealth of recent work in establishing the detectability limits of communities.

The first part of our work also examines the problem of spectral clustering in stochastic blockmodels. Many work focuses on analyzing the top $K$ singular vectors $V$. In our report, we propose that the matrix $VV^T$ is a more natural object to consider from a theoretical point of view. Under some assumptions, the matrix $VV^T$ can be shown to be close to an ideal matrix that is a “perfect” encoding of the cluster membership.

For the second part of our work, we focus on dynamic community detection. We consider the extension of the stochastic blockmodel to the dynamic setting first considered by [Yang et al., 2011], [Matias and Miele, 2017], where the community membership evolves in time as a Markov chain. Our approach focuses on first inferring the set of nodes whose cluster membership was stable during a given time window. We call this set of nodes the
A. Notations and Definitions

In this section, we introduce notation to be used in the rest of this paper. Let \( n \) be the number of nodes, \( K \) be the number of communities and \( V = [n] \) be the set of nodes. Let \( C = \{C_1, \ldots, C_K\} \) denote the hidden \( K \)-partition of \( V \). Let \( c : V \rightarrow [K] \) be the cluster assignment function, i.e., \( c(i) \in [K] \) is such that \( i \in C_{c(i)} \). Let \( P \in [0,1]^{K \times K} \). We first recall the definition of the (classical) stochastic blockmodel:

**Definition 1.** A stochastic blockmodel (SBM) random graph has the following generative model: \( G = (V,E) \) is such that the edge \((i,j) \in E\) appears with probability \( P_{c(i)c(j)} \). Furthermore, given the clustering \( C \), the edges appear independently. We denote this random graph by \( G \sim SBM(P,C) \).

Sometimes, the above definition is also referred to as the hidden partition model. Next, let \( T \in \{1,2,\ldots\} \cup \{\infty\} \) denote the time horizon and \( \{p_t\}_{t=1}^{\infty} \) be such that \( p_t \in [0,1]^{K \times K} \).

**Definition 2.** A dynamic stochastic blockmodel (DSBM) with time horizon \( T \) is a sequence of random graphs \( \{G_t\}_{t=1}^{T} \) such that \( G_t \in SBM(p_t,C) \). Furthermore, for \( t_1, t_2 \in [T] \) distinct, \( G_{t_1} \) and \( G_{t_2} \) are independent.

Below, let \( \mathbf{1}_{m,n} \) be the \( m \times n \) matrix of all ones.

B. Proposed approach

Like many spectral methods for community detection, our main algorithm leverages a spectral decomposition \( U \Sigma V' \) of affinity matrix \( A \). However, in contrast to most other works, we post-process \( V \) differently using the following greedy algorithm, which accepts data points \( S \) from any metric space \( (\mathcal{X},d) \) and recursively compute a \( K \)-clustering \( C = \{C_1, \ldots, C_K\} \) of \( S \):
stochastic blockmodel random graphs using spectral method. In [McSherry, 2001], a projection algorithm is devised that projects onto $E[G]$.

[Vu, 2017] presents an answer to a question posed at [McSherry, 2001] by proposing a so-called perfect representation. I believe the definition is overly restrictive and unnecessary for clustering to work well. In a sense, our work is also based on studying a perfect representation (3) where distinct cluster centers have distance 2. I believe our method is simpler theoretically and performs well empirically. We prove Corollary 6 below which is analogous to Corollary 11 of [Vu, 2017].

In [Han et al., 2015], the matrices $P^t$ is assumed to be sampled from a stationary ergodic random process with identifiable mean $M := E[P^{(t)}]$. In Section V-A, we make one further (relatively harmless) assumption that $M$ is nonsingular and prove a finite-time bound on the clustering error using the machinery we introduce in Section IV.

[Krzakala et al., 2013] applies spectral method to the so-called non-backtracking matrix for community detection. They show link their method to belief propagation, which is a class of algorithm often used in inference on graphical models. Their variant of belief propagation requires good estimate of the edge probabilities. We introduce a variant of the update rule similar in spirit, but not identical to belief propagation which performs well in practice. Rather than requiring estimates of the edge probabilities, we require a good clustering of a subset of nodes.

[Matias and Miele, 2017], [Yang et al., 2011] studies dynamic stochastic blockmodel graphs where the cluster membership evolves as a Markov chain. [Yang et al., 2011] uses methods from Bayesian statistics and statistical mechanics while [Matias and Miele, 2017] develops a variational EM-algorithm for inferring the communities. We explore a heuristic algorithm based on finding a core community which can be clustered well, then using a dynamical system to extend to a full clustering.

III. OUTLINE OF WORK

In Section IV, we study the case when we only have a single graph from a stochastic blockmodel, i.e., the static setting. In Section V, we study the dynamic setting in its two variants, time-varying edge probability matrices, and time-varying cluster memberships.

IV. THE STATIC SETTING

In this section, we rigorously study Algorithm 2 for community detection when we have a singular sample $A \sim SBM(P, C)$. Since our algorithm works with top $K$ (right) singular vectors, our main ingredients are the Davis-Kahan theorem and a misclassification result. The version of Davis-Kahan theorem we use is from [Yu et al., 2014] and can be restated as

**Theorem 3 ([Yu et al., 2014]).** Suppose $\Sigma, \hat{\Sigma}$ are $n \times n$ symmetric matrices and $V, \hat{V}$ are the $n \times K$ matrices of the top $K$ right singular vectors, then there exists an $K \times K$ orthogonal transformation $O$ such that

\[
\|VO - \hat{V}\|_F \leq 2^{3/2} \min \{ \sqrt{K} \|\Sigma - \hat{\Sigma}\|_{op}, \|\Sigma - \hat{\Sigma}\|_F \} \sigma_K(\Sigma) \tag{1}
\]

We apply this theorem with $\Sigma = E[A]$ and $\hat{\Sigma} = A$. However, rather than using the the singular vectors themselves, we use their outer product. For a matrix $M$, let $\|M\|_{1,1}$ denote the sum of the $\ell_1$-norms of its columns, i.e.,

\[
\|M\|_{1,1} := \sum_i \|M[:, i]\|_1
\]

In the situation of Theorem 3, it is easy to see that

\[
\frac{1}{n} \|VV^T - \hat{V}\hat{V}^T\|_{1,1} \leq 2\sqrt{K}\|VO - \hat{V}\|_F
\]

Let

\[
\epsilon := \sqrt{K}2^{5/2} \min \{ \sqrt{K} \|A - E[A]\|_{op}, \|A - E[A]\|_F \} \sigma_K(E[A]) \tag{2}
\]

then we have

\[
\frac{1}{n} \|VV^T - \hat{V}\hat{V}^T\|_{1,1} \leq \epsilon.
\]

Let $n_k = |C_k|$ be the size of the $k$-th ground-truth community. Assuming that $P$ is nonsingular (which is satisfied if $P_{ii} > P_{ij}$ for all $i \neq j$), then we have

\[
VV^T = \begin{bmatrix}
\frac{1}{n_1} & 1_{n_1,n_1} & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
0 & \cdots & \frac{1}{n_k} & 1_{n_k,n_k}
\end{bmatrix}
\]

(3)

In this case, note that the $\ell_1$ distance between the columns of $VV^T$ is either 0 or 2, indicating the true membership of the column. Thus, we believe that $\hat{V}\hat{V}^T$ is a better representation of the cluster membership of the nodes.
A. Performance guarantee upper bound

We show how the representation $\hat{V}V^T$ can be used to bound the misclassification rate in the hidden bipartition problem in Corollary 6. Our main tool is the following

**Proposition 4.** Let $(X, d)$ be a metric space, and let $S = \{s_1, \ldots, s_n\}$ and $T = \{c_1, \ldots, c_k\}$ be subsets of $X$. Let $C = \{C_1, \ldots, C_k\}$ be a partition of $[n]$. Define $n_k = |C_k|$. Suppose

i. $d(c_i, c_j) = 2$ for $i \neq j \in [K]$,

ii. $\frac{1}{n} \sum_{k \in [K]} \sum_{i \in [k]} d(s_i, c_k) \leq \epsilon$,

iii. $n_k > 6\epsilon$ for all $k \in [K]$

Then Algorithm 1 computes a clustering whose misclassification rate relative to $C$ is less than $9\epsilon$.

Thus, our pipeline works well if $\epsilon$ can be made small. To calculate $\epsilon$, we consider the more restricted setting of recovering hidden bipartition as stated in [Vu, 2017]:

**Proposition 5.** Suppose $K = 2$ and $n_1 = n_2 = n/2$ have equal sizes. Furthermore, suppose that

$$P = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

where $p > q$. For some $C > 0$, suppose the following holds:

$$\frac{p - q}{\sqrt{p}} \geq C \sqrt{\log n / n}$$

Then $\epsilon$ as defined at (2) satisfies

$$\Pr \left( \epsilon \leq \frac{C_1}{\sqrt{C \log n}} \right) \leq 1 - n^{-3}$$

for some universal constant $C_1$.

**Corollary 6.** In the situation of Proposition 5, suppose that

$$\frac{C_1}{\sqrt{C \log n}} < 1/12$$

Then with probability $1 - n^{-3}$ the hidden bipartition can be computed by Algorithm 2 with misclassification rate no greater than

$$\frac{9C_1}{\sqrt{C \log n}}$$

Proof of corollary. In Proposition 4, we just need to make sure that $n_k > 6\epsilon$ for all $k \in [K]$. Since $n_k = n/2$, we just need to make sure that $1/12 > \epsilon$ which is guaranteed if $\epsilon \leq \frac{C_1}{\sqrt{C \log n}} < 1/12$.

Proof of Proposition 4. We shall use [Vu, 2017]:

**Lemma 7** ([Vu, 2017]). There is a constant $C_0 > 0$ such that the following holds. Let $E$ be a symmetric matrix whose upper diagonal entries $e_{ij}$ are independent random variables where $e_{ij} = 1 - \rho_{ij}$ or $\rho_{ij}$ with probabilities $p_{ij}$ and $1 - p_{ij}$, respectively, where $0 \leq p_{ij} \leq 1$. Let $\sigma^2 := \max_{i,j} \rho_{ij}(1 - \rho_{ij})$. If $\sigma^2 \geq C_0 \log n / n$ then $P(\|E\| \geq C_0 \sigma n^{1/2}) \leq n^{-3}$.

In our notations, we have

$$\sigma^2 = \max\{p(1 - p), q(1 - q)\} \leq p.$$ 

The matrix $A - E[A]$ satisfies the condition of $E$ in the Lemma. Hence, we have

$$\Pr(\|A - E[A]\| \geq C_0 \sqrt{np}) \leq n^{-3}.$$ 

Suppose that $\|A - E[A]\| \leq C_0 \sqrt{np}$, then by (2), we have

$$\epsilon \leq C_0^{2/3} \frac{\sqrt{np}}{\sigma_k(E[A])}$$

Now, note that, $E[A] = 1_{s,s} \otimes P$. Since $1_{s,s}$ has rank 1, we have

$$\sigma_k(E[A]) = \sigma_k(P)\sigma_1(1_{s,s}) = (p - q)n/2$$

Putting it all together, we have

$$\Pr \left( \epsilon \leq C_0 2^{4} \sqrt{\frac{p}{n(p - q)^2}} \right) \leq 1 - n^{-3}$$

If $\frac{p - q}{\sqrt{p}} \geq C \sqrt{\log n / n}$, then we have

$$\sqrt{\frac{p}{n(p - q)^2}} \leq \frac{1}{C \sqrt{\log n}}$$

Thus, we have

$$\Pr \left( \epsilon \leq \frac{C_0^{2/3}}{\sqrt{\log n}} \right) \leq 1 - n^{-3}$$

\]

V. DYNAMIC COMMUNITY DETECTION

Researchers have studied dynamic community detection mainly under two settings: (1) varying $p(t)$ and (2) varying $C(t)$. In [Matias and Miele, 2017], the authors gave an example illustrating the difficulty of community detection when both $p(t)$ and $C(t)$ varies. Thus, we also consider them separately here. First, we consider the case of varying $p(t)$.

A. Varying the edge probability matrix

We consider the setting studied in [Han et al., 2015], in which the authors established asymptotic consistency result for spectral clustering of the summation network of SBM graphs. The goal of this section is to extend their Theorem 1 to a finite time bound on the misclassification rate.
It is relatively easy to apply the methods we have described in the previous section to study the case when only the edge probability matrix varies. The setting is as follow: assume that the edge probability matrices $P^{(1)}, P^{(2)}, \ldots$ comes from an ergodic stationary process. Let $M = \mathbb{E}[P^{(i)}]$. Let $A^{(i)} \sim \text{SBM}(P^{(i)}, C)$ where $C$ does not vary with time. Let

$$
\bar{A}^{(T)} = \sum_{t=1}^{T} A^{(t)}
$$

Furthermore, let

$$
\sigma = \max_{k\ell} \text{Var}(p_{k\ell}^{(i)})
$$

To derive a finite time bound using Algorithm 2, we need to further assume that $M \in \mathbb{R}^{K \times K}$ is linearly independent.

**Theorem 8.** Suppose that $M$ is linearly independent. Then for all $\delta > 0$ sufficiently small, with probability at least

$$
1 - \frac{N^2(1 + 4\sigma^2)}{4T\delta^2}
$$

we have that Algorithm 2 can recover $C$ based on $\bar{A}^{(T)}$ with misclassification rate less than

$$
9 \times 2^{5/2}\sqrt{K}\delta.
$$

We note that all terms in (4), except for $\delta$, are determined by the model parameters (the number of communities, the relative sizes of the communities, and the ergodic random process.)

**Proof.** Similar to the proof of Proposition 5, we only need to bound

$$
\epsilon := \sqrt{K}2^{5/2} \frac{\|\bar{A}^{(T)} - \mathbb{E}[A^{(i)}]\|_F}{\sigma_K(\mathbb{E}[A^{(i)}])}
$$

Indeed, this is just (2) with the Frobenius norm rather than the operator norm. From [Han et al., 2015] Appendix A, we have that

$$
\Pr\left(\|\bar{A}^{(T)} - \mathbb{E}[A^{(i)}]\|_F > \delta\right) \leq \frac{N^2(1 + 4\sigma^2)}{4T\delta^2}
$$

Now, $\|\bar{A}^{(T)} - \mathbb{E}[A^{(i)}]\|_F \leq \delta$ implies that

$$
\epsilon \leq \sqrt{K}2^{5/2} \frac{\delta}{\sigma_K(\mathbb{E}[A^{(i)}])}
$$

Now, we simply apply Proposition 4, whose conditions are satisfied for all $\delta$ sufficiently small. \qed

B. Varying the cluster membership

The case of varying cluster membership is more challenging. In particular, we simply take the time average $\bar{A}^{(T)}$ as in the previous section, then it is no longer true that $\mathbb{E}[\bar{A}^{(T)}]$ has any obvious property that we can exploit. Nevertheless, one hopes that if the cluster membership doesn’t change very quickly, then techniques from the static setting should work well.

We first introduce the model that we study in this report. Following [Matias and Miele, 2017], we consider a time-series of adjacency matrices $A^{(t)}$ sampled from $\text{SBM}(P, C^t)$ where $C^t = (c^t_1, \ldots, c^t_N)$ evolves as a Markov chain. That is to say, there exists a matrix $Q$ such that

$$
\Pr(c^{t+1}_i = \ell | c^{t}_i = k) = Q_{k\ell}
$$

For simplicity, we focus only on the case when there are two communities. Since there are only two communities, we shall use $\pm 1$ to denote the community label rather than 1, 2.

**Definition 9** (Core nodes). Given time $t$ a finite time window $L$, the set of core nodes is defined as

$$
\text{Core}(t, L) = \{ i : c^{t}_i = c^{t+1}_i = \cdots = c^{t+L}_i \}
$$

In other words, the core nodes do not change community between time $t$ and $t + L$. Since $C^t$ evolves as a Markov chain, the set $\text{Core}(t, L)$ is random and almost surely empty for large $L$. On the other hand, if we take the summation network

$$
S(t, L) := A^{(t)} + \cdots + A^{(t+L)}
$$

then the $\text{Core}(t, L) \times \text{Core}(t, L)$ minor of $S(t, L)$ is essentially equivalent to the summation network of $L$ stochastic blockmodel random graphs whose community membership is static. From [Taylor et al., 2016], the detectability limit is

$$
\Delta^*\text{dynamic} = \sqrt{\frac{4\rho(1-\rho)}{L \times |\text{Core}(t, L)|}}
$$

for $|\text{Core}(t, L)|$ large. In comparison, if we analyze the static case at any time $t$, then static theory [Nadakuditi and Newman, 2012] tells us that the detectability limit is

$$
\Delta^*\text{static} = \sqrt{\frac{4\rho(1-\rho)}{N}}
$$

Since a lower detectability limit is desirable, we hope to have

$$
N < L \times |\text{Core}(t, L)|
$$

(5)
Below, we will examine how different choices of \( L \) influences the community detectability.

Once the core is selected, we cluster the core nodes. We use the clustering on the core nodes as the initialization to dynamical system that clusters the non-core nodes. But before we discuss the dynamical system, we first discuss how to find the core nodes.

1) Finding core nodes: We first consider the following simplification: suppose we simply have

\[
A_{ij}^{(t)} = \begin{cases} 
P_{C_t}^i \cdot P_{C_t}^j & : i \neq j \\
0 & : i = j
\end{cases}
\]

rather than

\[
A_{ij}^{(t)} \sim \text{Bernoulli}(P_{C_t}^i \cdot P_{C_t}^j).
\]

This is the noiseless case for which the community detection problem is completely trivial. Nevertheless, we study this to get intuition for finding the core community. Now, let \( B \) be the summation network over the time horizon \( T = 2 \), i.e.,

\[
B^{(2)} = \frac{(A^{(1)} + A^{(2)})}{2}
\]

During time \( t = 1 \), the matrix \( A^{(1)} \) looks like Figure 1.

Fig. 1. Left: noiseless \( A^{(1)} \), preclustered. The first community is column 0 through 4. The second community is column 5 through 9. Right: \( VV' \) corresponding to \( A^{(1)} \).

Now, suppose in \( t = 2 \), the second column switched cluster membership. Then the summation network \( B^{(2)} \) looks like Figure 2.

![Fig. 2](image)

Fig. 2. Left: noiseless \( B^{(2)} \). Column 2 switched cluster membership. Right: \( VV' \) corresponding to \( B^{(2)} \).

One can detect which node changed cluster membership simply by looking at the right heatmap of Figure 2. Namely, column 2 (numbering starts at 0) has no zero entry. We now describe this process in practice. Let \( USV' \) be the rank \( k \) approximation of \( B^{(2)} \) and consider \( S := VV' \). Let \( \bar{S} \) be the column-stochastization of \( S \), i.e., negative elements of \( S \) are first set to zeros then the columns are normalized to sum to 1. If column \( i \) of \( \bar{S} \) had no zeros, then we say that community membership of \( i \) changed from \( t = 1 \) to \( t = 2 \). Conversely, if the community membership of \( i \) did not change, then there must be zeros in the \( i \)-th column of \( \bar{S} \).

This is the core-finding algorithm that we use which we summarize in Algorithm 4 below.

**Algorithm 4 CoreFinding(\( A, K \))**

**Input:** \( A \) (weighted) adjacency matrix, \( K > 1 \) number of communities

1. \( USV' \leftarrow \) best rank \( K \) approximation of \( A \)
2. \( S \leftarrow VV' \)
3. \( \bar{S} \leftarrow \) column stochastization of \( S \)

**Output:** \( \mathcal{I}_I = \{ i : \bar{S}_{ij} = 0 \text{ for some } j \} \)

While we do not have any theoretical result to justify Algorithm 4, we will provide some observations from synthetic data to justify it. In Figure 3, we show an empirical example of when cluster membership changes. In Figure 4, histogram of values in the \( VV' \) matrix is shown.
Fig. 4. In this graph $A^{(1)}$ has two equal sized cluster both of size 50. The red block submatrix of $A$ is concentrated around the value $0.02 = 1/50$ while the black block submatrix concentrates around 0. This provides some heuristic as to why we can expect the black block submatrix to be filled with zeros if we column stochasticize $VV^T$.

2) Clustering non-core nodes using core nodes:
Given the core nodes $I$, we first cluster them and get $C_I = \{ c_i : i \in I \}$. We extend this to a clustering on the whole graph using the following method. Define a vector $v$ whose $i$-th component is

$$v_i = \begin{cases} c_i : i \in C_I \\ 0 : i \notin C_I \end{cases}$$

(6)

Then we run the iterate

$$v \leftarrow \text{sign}(\tilde{S}v)$$

(7)

where sign is applied entrywise. We repeat this iterate until a fix point $v^* = \text{sign}(\tilde{S}v^*)$ is reached. In practice, we find that running a single iteration already yields the fix point. The proposed two phases algorithm is summarized in Algorithm 5.

Algorithm 5 CoreFindingClustering($A$, $K$)

\textbf{Input:} $\{G^t\}_{t=1,...,T}$ a sequence of graphs, $K > 0$ a positive integer
1: Get the weighted adjacency matrix via Algorithm 3
2: Finding the core community via Algorithm 4
3: Initialize the labels via Eq (6)
4: For $t = 1, \ldots, T$, 
5: Update the labels via Eq (7) until it converged

Figure 5 illustrates Algorithm 5 on a small graph.

Fig. 5. Top: Ground truth. Bottom left: clustered core nodes found by Algorithm 5. Bottom right: final clustering produced via equation (4). For this example, we used 100 nodes and two equal sized communities. $p_{in} = 0.09$ and $p_{out} = 0.03$. Furthermore, the membership switching probability is $\epsilon = 0.1$. We used $T = 3$ time average. The graph shown is the last graph in the time series.

VI. Empirical

A. Simple aggregation of multilayer graphs

First, we test the performance of Algorithm 3 on graphs $A^{(1)}$ sampled from the identical stochastic blockmodel SBM($P$, $C$) by simply aggregating them. Although the post-process Algorithm 1 is good for analysis, we found that in practice, kmedoids with $\ell_1$-norm works better. This empirical test serves as a sanity check to ensure that clustering the outer product $VV^T$ performs on about the same level as clustering $V$. The advantage of studying $VV^T$ is that the entries of $VV^T$ contains a richer amount of information. For instance, Algorithm 4 was based on looking at the positivity of entries of the column-stochasticized $VV^T$ to find the core communities. It is not obvious how one can infer this information from $V$ directly.

B. Tracking the time varying networks by leveraging of the core community

In this section, we investigate the performance of proposed algorithm over different factors. In Figure

C. Effect of community switching on clustering accuracy

In Figure 12 below, we vary $\epsilon$ the probability that a node switches community labels. We use $p_{in} = 0.05$ and $p_{out} = 0.02$. In the static case, these parameters are well below the detectability limit. However, aggregation of many layers can boost the
In the above experimental results, the color shows the clustering accuracy. We let $\Delta = p - q$ and $\rho = (p + q)/2$. The two clusters are equal in size and there are 100 nodes. Each pixel is a one clustering instance on 16 graphs generated from the same SBM. The red curve shows the detectability limit. The top figure uses the kmedoids on the outer product $VV^\top$ with $\ell_1$-norm while the middle figure uses kmedoids on $V$ with $\ell_2$-norm. The bottom figure shows the difference in accuracy between the two post-processing steps. Evidently, the two methods achieve similar performance while the outer product post-processing has a more intuitively clear interpretation.

To illustrate this, consider Figures 9 through 11, which are summation networks draw from $SBM(P, C^t)$ at $t = 1, 3, 6$ respectively. The parameters $P$ were chosen so that at each time, the graph is below detectability limit. The community switching parameter $\epsilon = 0.2$ was chosen to ensure fast mixing. This means it unfavorable to have a summation network over too many layers because the nodes switches community very quickly, leading to loss of information.

In Figure 12, we swept across a range of community switching probabilities $\epsilon = 0.05$ to 0.3, corresponding to slow and fast mixing and investigated the clustering accuracy of clustering the summation network.

**VII. Conclusion and Future Directions**

In this work, we studied the spectral method for both the static and dynamic stochastic blockmodel. While we were able to derive some theoretical
Fig. 8. Clustering accuracy of the proposed algorithm over different $\rho$ and $\Delta$. For this plot, $P_{in} + P_{out} = \rho$ and $P_{in} = C \times P_{out}$. $N = 1000$, switching portion equals 0.2 at each time index, the core community portion equals to 0.2.

Fig. 9. Left: Summation network $B^{(1)} = A^{(1)}$ of a single layer draw from a SBM that is below the detectability limit. Right: The corresponding $VV^T$ matrix. Note that the community structure cannot be detected from $VV^T$.

Fig. 10. Left: Summation network $B^{(3)} = A^{(1)} + A^{(2)} + A^{(3)}$ of a SBM that is below the detectability limit (continuation of Figure 9). Right: The corresponding $VV^T$ matrix. Note that some community structure can be detected from $VV^T$. For this case, 3 layers is the sweet spot where not too much mixing occurred while the aggregation boosted the summation network above the detectability limit.

Fig. 11. Left: Summation network $B^{(6)} = A^{(1)} + \ldots + A^{(6)}$ of a SBM that is below the detectability limit (continuation of Figure 10). Right: The corresponding $VV^T$ matrix. Note that no community structure can be detected from $VV^T$. At this point, too much mixing has occurred.

Fig. 12. For each individual time, the SBM parameter is chosen to be well below the detectability limit ($P_{in} = 0.1$, $P_{out} = 0.03$, $N = 100$ node, equal community size). Clustering each time snapshot network leads to poor clustering result (accuracy $\approx 0.5$). Evidently, when $\epsilon = 0.05$, aggregation allows some recovery of community. However, when the community switching probability increase, communities can no longer be recovered.

result for the static and dynamic stochastic blockmodel when the cluster membership do not change, we were unable to rigorously analyze the cases when the cluster membership do change. Nevertheless, we proposed Algorithm 5 that finds a set of core nodes for which we suspect clustering should be easy.

We were only able to establish some preliminary empirical results of the algorithm. It remains a future work to thoroughly compare our algorithm to other dynamic community detection algorithms. We also would like to experiment with Markovian dynamics that takes into account of the graph topology. We suspect that when the cluster membership evolution respects the graph topology, finding a good set of core nodes should be easier than when the nodes evolves independently. This should be somewhat analogous to a smoothness assumption.
VIII. WORK DISTRIBUTION

Both Dejiao and Yutong performed empirical studies and researched idea for developing Algorithm 5. Yutong worked on the theoretical results in Section IV.

REFERENCES


