SCALABLE COMMUNITY DETECTION IN THE HETEROGENEOUS STOCHASTIC BLOCK MODEL

Andre Beckus and George K. Atia

Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816 USA

ABSTRACT

This paper studies the unsupervised clustering of large graphs generated from the heterogeneous Stochastic Block Model. We present a sketch-based community detection algorithm, which substantially reduces computational complexity by clustering only a small set of nodes sampled from the full graph followed by a retrieval algorithm. We first show cases where existing algorithms exhibit reduced error rates when all nodes possess the same average number of intra-cluster connections. This behavior is demonstrated for both convex-optimization-based and spectral algorithms. Based on this insight, we develop SPIN, a degree-based sampling method to produce sketches with cluster proportions more favorable for successful clustering. By sampling nodes inversely proportional to their degrees, SPIN can exploit this reduction in error to significantly improve the phase transition as compared to full graph clustering.

Index Terms— clustering, community detection, stochastic block model, random graphs, sampling

1. INTRODUCTION

The detection of clusters in graph-based data proves vital in a diverse array of applications ranging from image segmentation [1] to biology [2]. Community detection algorithms cluster graphs into communities of nodes within which edges are especially dense (see the survey paper [3] for more information). Classic approaches to community detection include hierarchical methods [4] and spectral clustering [1], while more recent approaches include matrix decomposition [5], expansion of communities from seeds [6], and a wide variety of other approaches.

However, these algorithms often store the full graph in memory, and depend on time-consuming computational steps (such as eigen decompositions of the entire adjacency matrix) which scale super-linearly with graph size. These weaknesses only become amplified as data sets grow in size. Additionally, as will be demonstrated in Section 2.1, many algorithms exhibit higher node misclassification rates when the graph contains small clusters containing sparse edge connections. Here, we aim to improve scalability over existing algorithms while at the same time reducing the error rates when the clusters possess differing edge densities.

We use a variant of the Stochastic Block Model (SBM), sometimes referred to as the heterogeneous SBM [7, 8]. This probabilistic model generates a graph containing a planted set of clusters with each node belonging to one cluster. The size of cluster i is denoted n_i. Edges within cluster i are created with probability p_i (thus the designation “heterogeneous”), and edges are created between clusters with probability q. The classic homogeneous SBM, is a special case in which all clusters have the same intra-cluster edge probability $p_i = p$.

This paper builds on our previous work [9], which proposed a sketch-based randomized framework for the homogeneous SBM. This framework significantly reduces the complexity of the clustering algorithm, and improves performance when the cluster sizes are unbalanced.

However, it is well-known that the homogeneous SBM is severely limited in its ability to fit real-world data [10]. A particular failing of this model is that all clusters must have the same edge densities. As a consequence, once the global value p is fixed, the degree of a node can only vary in proportion to the size of its parent cluster.

In contrast, the *heterogeneous* SBM considered extensively here allows cluster-by-cluster variation of the intra-cluster edge densities, and therefore allows both size *and* degree to vary independently. The extra degrees of freedom better capture the natural fluctuations found in real data sets.

Due to its improved qualities, several recent works have focused on this heterogeneous SBM. The sufficient conditions of some works, e.g., [11], fail when there is *either* a small cluster *or* a small p_i. However, as described in [8], these bounds appear too conservative in practice; if there is a large sparse cluster or a small dense cluster, then clustering tends to perform better than predicted by these loose bounds.

The authors of [8] conjecture that a better figure of merit is the minimum of the “relative densities”, which are defined as $(p_i - q)n_i$. Likewise, [12] expresses the sufficient conditions of their clustering algorithm using the quantity $n_i(2p_i - 1)$, named “effective density”. Both relative and effective densities share a common trait, namely that they depend heavily on the *product* of n_i and p_i. We show in our results that the min-

This work was supported by NSF CAREER Award CCF-1552497.

imum of this product indeed serves as a sound predictor of success among multiple algorithms, not just those of [8, 12]. We use this insight to design a new sampling method to be used in conjunction with our proposed algorithm.

The algorithm proposed here can invoke a wide variety of clustering algorithms when clustering the sketch. Here, we will use two classes of community detection algorithms: convex-optimization-based and spectral algorithms. Among the convex algorithms, we have two convex relaxations of Maximum Likelihoood (ML) estimators [11] and [8], henceforth referred to as (Cai, 2015) and (Jalali, 2016), respectively. The third algorithm, CMM [10] maximizes a convex relaxation of the modularity quality function. Among the spectral methods, we use the classic Spectral Clustering algorithm proposed in [1], as well as the more recent SCORE algorithm [13] which exhibits improved resilience in the presence of varying node degree. The data model is defined as follows [8, 12].

Data Model 1 (Heterogeneous SBM)**.** *The graph consists of N nodes partitioned into r disjoint clusters. The size of cluster i is denoted n_i. Any two nodes within cluster i are connected with probability p_i, and two nodes belonging to different clusters are connected with probability q.*

1.1. Summary of contributions

We propose a sketch-based algorithm and demonstrate its effectiveness when used on graphs generated from the heterogeneous SBM. During the sampling process, the algorithm has the opportunity to optimize the sketch to improve the overall success rate of the algorithm. To exploit this opportunity, we identify a condition on cluster size under which the probability of clustering success is demonstrably high among multiple clustering algorithms. From this insight, we formulate a suitable sampling method from first principles, thus improving the success rate of the algorithm, while also maintaining the fast execution speeds inherent to the sketch-based approach.

2. SKETCH-BASED CLUSTERING

In this section, we describe the proposed sketch-based approach shown in Algorithm 1.

The algorithm first produces a small sketch by sampling a subset of the nodes from the full graph. The sketch graph is the sub-graph induced by these sampled nodes, i.e., containing only the sampled nodes and edges between these nodes.

In the second step, the sketch graph is clustered using a clustering algorithm. While we only consider five algorithms, there is no inherent barrier to using other clustering algorithms as well. Based on the insight developed in Section 2.1, we expect that a wide variety of algorithms will perform well provided that the first sampling step produces a good sketch.

Algorithm 1 Sketch-based Clustering Algorithm

```
Input: Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
1. Random Node Sampling:
Randomly sample $N'$ nodes without replacement using either the URS or SPIN sampling methods (see Section 2.2 for definitions and details). Construct a graph containing only the $N'$ sampled nodes and the edges between them. Let $\mathcal{I}$ be the set of samples.
2. Sketch Clustering:
Cluster sketch graph using one of the following algorithms:
(Cai, 2015) [11], (Jalali, 2016) [8], CMM [10], Spectral Clustering [1], SCORE [13].
Let $\hat{r}$ be the number of detected clusters (for algorithms that require the number of clusters be specified, let $\hat{r} = r$). The output of this step will be the set of characteristic vectors of the clusters in the sketch matrix $\{\mathbf{v}_i \in \mathbb{R}^{N'}\}_{i=1}^{\hat{r}}$.
Each characteristic vector represents one cluster, and has an element for each node, containing a 1 if the node belongs to the cluster, and a 0 otherwise.
3. Full Graph Clustering:
Let $\mathbf{a}_{k\ell}$ be 1 if there is an edge between nodes $k$ and $\ell$, and 0 otherwise.
For each node $j$ which is in the full graph but not in the sketch graph calculate
    $u_j = \arg\max_i \frac{\sum_{k \in \mathcal{I}} \mathbf{a}_{jk} \mathbf{v}_{ik}}{\sum_{k \in \mathcal{I}} \mathbf{v}_{ik}}$.
    Assign the $j^{\text{th}}$ node to the $u_j^{\text{th}}$ cluster.
End For
```

Finally, the results of the clustering are extrapolated to each node in the full graph based on the number of connections that the node has to each cluster in the sketch.

The sampling step is key to the algorithm, as the choice of sampled nodes will heavily influence the success of the remaining steps. For this reason, we next consider what qualities we want the sampling method to possess.

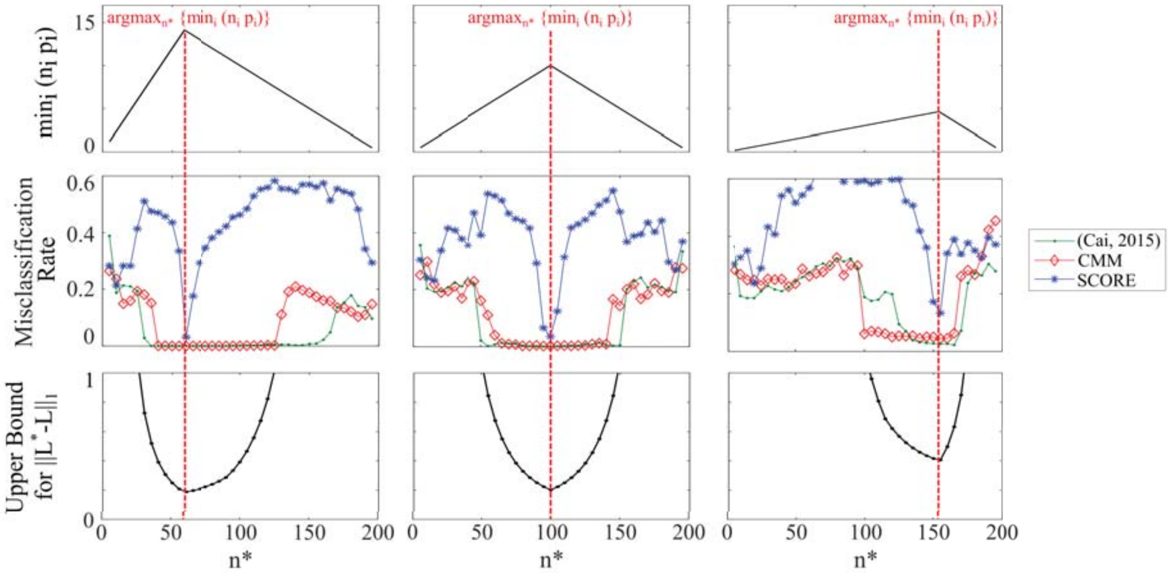
2.1. Desirable Sketch Properties

When generating the sketch, we wish to choose the nodes so as to maximize the probability of successfully clustering the sketch. The problem can be framed as follows: given that cluster i has intra-cluster edge probability p_i, and that we will sample N' nodes, how many of these samples should come from cluster i?

First, consider the homogeneous SBM. It was shown in [9] that the matrix-decomposition-based algorithm [5] has the strongest guarantees on success when an equal number of nodes are sampled from each cluster.

However, it is unclear if this property also holds for other algorithms as well. More importantly, it is not at all obvious what the corresponding goal should be in the *heterogeneous* case when the probabilities p_i are allowed to vary for each cluster. The following remark suggests an answer.

Remark 1. *In [8, Theorem 3], lower bounds are established for the success probability of a modified Maximum Likelihood (ML) estimator. Let $\eta = \frac{\min_i\{n_i(p_i - q)\}}{4D} - 17$, where D is dependent only on p_i, q, and N, and not on the cluster sizes. Applying the theorem gives that the success probability is not less than $1 - 5\frac{p_{max} - q}{p_{min} - q} N^{(2-\eta)}$ provided that $\eta > 0$, $n_{min} \geq 2$, and $N \geq 8$ (n_{min} is the smallest cluster size, $p_{min} = \min_i p_i$, and $p_{max} = \max_i p_i$). If we vary $n_1, \cdots, n_r$ while*



(a) p*=0.24

(b) p*=0.1

(c) p*=0.03

Fig. 1. Motivating example showing that the misclassification rate is often relatively small when condition (2) is met. Results are averaged over 10 trials.

holding all other parameters constant, the maximum value of η is attained when

$$n_1(p_1 - q) = n_2(p_2 - q) = \cdots = n_r(p_r - q). \qquad (1)$$

This value of η maximizes both the lower bound on success probability and the chance of meeting the sufficient condition $\eta > 0$.

The previous remark provides valuable insight for how best to construct a sketch graph. We now provide numerical results to demonstrate the validity of this insight, and to demonstrate that an optimality condition similar to (1) carries over from the computationally intractable estimator of [8] to practical polynomial-time algorithms.

In the ideal case, when $q = 0$, condition (1) becomes

$$n_1p_1 = n_2p_2 = \cdots = n_rp_r, \qquad (2)$$

i.e., the condition is met when all nodes have the same expected degree. For clarity of exposition, in the following we will assume that $q \ll p_i$, and use the simplified condition (2).

We consider three algorithms: (Jalali, 2016), CMM, and SCORE. Fig. 1 shows simulations for a graph of size $N = 400$ (chosen to match a typical sketch size used in this paper) having $r = 4$ clusters with probabilities $p_1 = p_2 = 0.1$, $p_3 = p_4 = p^*$, $q = 0.001$, and cluster sizes $n_1 = n_2 = (N - 2n^*)/2$, $n_3 = n_4 = n^*$. The parameters p^* and n^* are to be varied (note that all cluster sizes are uniquely determined by n^*). Two clusters each are included with the same size/probability in order to make the problem more challenging. For example, this avoids the situation where clusters can be identified using a simple threshold applied to the node degrees.

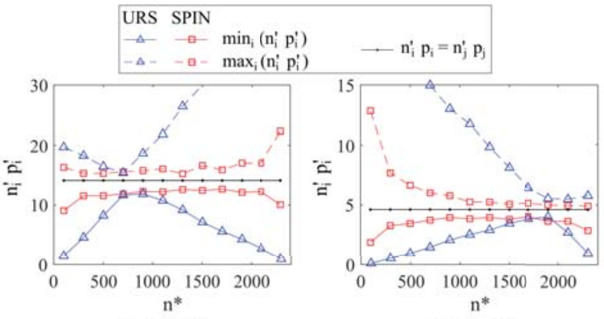
The top row of Fig. 1 shows the minimum product n_ip_i among all clusters, and the vertical red dashed line indicates where this quantity is maximized, thus satisfying (2). Each column contains results for a different value of p^*. In the left column, clusters 3 and 4 are more dense than clusters 1 and 2, which triggers the ideal condition when n^* is small. In other words, clusters 3 and 4 must be made smaller to compensate for the fact that they are denser. The opposite condition is shown in the right column, where clusters 3 and 4 are sparser than clusters 1 and 2, thus requiring a large n^* to satisfy the condition. The homogeneous condition is shown in the middle column, where all clusters should have the same size.

The middle row of Fig. 1 shows the misclassification rates for the three algorithms. The algorithms consistently work well in varying ranges around condition (2), with an especially sharp, narrow drop for SCORE.

The bottom row shows an analytic upper bound on the error $\|\mathbf{L}^* - \mathbf{L}\|_1$ for the CMM algorithm. Here, $\mathbf{L}$ is the cluster matrix output from the convex optimization step of the CMM algorithm, and $\mathbf{L}^*$ is the correct cluster matrix. The bound is numerically calculated using equation (3.7) found in [10, Theorem 3.1], with δ and λ set such that the bound is minimized while still satisfying the sufficient conditions. Note that the bound provided by [10] is scaled by an unknown constant, and so values in the plots are all normalized using the same scaling. As with the algorithm results of the middle row, this bound has a minimum in the vicinity of condition (2).

2.2. Sampling Methods

A natural first choice for the sampling method is Uniform Random Sampling (URS), in which each node has the same



(a) p*=0.24 **(b)** p*=0.03

Fig. 2. Example showing that SPIN approaches the ideal sketch cluster sizes over a wider range of n^* than URS.

probability of being sampled. This approach has the advantage of a simple implementation, and does not require calculations to be performed on the data prior to sampling. However, the expected number of samples from cluster i is $N'n_i/N$, which is proportional to the cluster sizes of the full graph. Thus, deviation from condition (2) in the full graph will tend to carry over to the sketch as well. For this reason, we now motivate the development of an improved sampling method.

Condition (2) is equivalent to requiring that all nodes in the cluster have the same average number of intra-cluster edges. This suggests that when q is not too large, the ideal condition is roughly satisfied by setting the sampling probability inversely proportional to node degree.

Based on this observation, we introduce a sampling method named "SamPling Inversely proportional to Node degree" (SPIN). With this method, N' samples are obtained from the full graph without replacement. The probability of sampling node j is set as B/d_j, where d_j is the degree of node j and $B = \left(\sum_{k=1}^{N} \frac{1}{d_k}\right)^{-1}$ is a normalization constant.

The following theorem and remark explore the performance of SPIN in the context of Data Model 1. Although our algorithm samples without replacement, the analysis uses replacement for simplicity. Sampling without replacement avoids duplication in the sketch, so we can generally expect better results from the algorithm. Due to space constraints, a proof is deferred to an extended version of this work.

Theorem 1 (SPIN cluster sampling probabilities)**.** *Suppose a graph is such that the nodes in cluster i have the expected degree $\mu_i = n_i p_i + (N - n_i)q$ from Data Model 1. Then, the probability of sampling with replacement from cluster i is $\frac{Cn_i}{\mu_i}$, where $C = \left(\sum_{j=1}^{r} \frac{n_j}{\mu_j}\right)^{-1}$.*

Remark 2. *If the condition in the statement of Theorem 1 holds, and we sample with replacement, then the expected number of nodes sampled from each cluster is $\overline{n}'_i = \frac{C n_i N'}{\mu_i}$. Let p'_i be the probability that a given intra-cluster edge in cluster i of the sketch exists, and q' be the probability that a given inter-cluster edge exists. With SPIN, we will have $p'_i \approx p_i$ and $q' \approx q$, and therefore $\overline{n}'_i p'_i \approx CN' / \left(1 + \frac{q(N-n_i)}{p_i n_i}\right)$.*

When $q = 0$, $\overline{n}'_i p'_i \approx \overline{n}'_j p'_j$ for any two clusters i and j, thus approximately meeting condition (2)*. Based on the discussion of Section 2.1, this indicates that SPIN will produce sketches favorable to a low misclassification rate.*

In Fig. 2 a numerical example is given demonstrating the superior performance of SPIN over URS with respect to (2). A full graph of size $N = 5000$ is generated, and then a sketch is constructed using $N' = 400$ samples. The clusters in the full graph have the same parameters as in Fig. 1. Here we vary the cluster sizes by changing n^* in the *full* graph. We then perform the sketching operation using both URS and SPIN, empirically calculate the edge probability p'_i for each sketch cluster, and calculate its product with the corresponding sketch cluster size. The minimum and maximum products are shown in Fig. 2 as solid and dashed lines, respectively.

Define n'_i as the size of cluster i in the *sketch*. For reference, the dotted black line shows where the condition $n'_i p_i = n'_j p_j$ is met. For SPIN, the minimum and maximum products approach this ideal line over a wide range of values, thus compensating for the imbalance in the full graph. This indicates that the sketch clustering success rate will improve over that of the full graph. In Section 3, numerical examples will be given verifying that this is indeed the case. On the other hand, URS only approaches the ideal line when the full graph is itself balanced, suggesting that URS will not yield an improvement (in fact, because the sketch graph is smaller, we can expect a possible degradation in performance).

The amount by which SPIN deviates from the ideal depends on a few factors. A larger q will result in a larger perturbation. However, if $\frac{q(N-n_i)}{p_i n_i} \to 0$, then this perturbation will diminish as the graph size increases. Remark 2 assumed that sampling is performed with replacement. However, in Algorithm 1, the sampling is performed without replacement. In this case, as columns are sampled from a cluster, the probability of attaining future samples from that cluster will diminish. For very small clusters, this effect can become pronounced, but for sufficiently large clusters the effect will be negligible. Finally, the degree of the nodes will not take their expected values, but will vary according to a binomial distribution. SPIN will tend to sample the nodes which have a smaller degree with higher probability.

3. NUMERICAL RESULTS

In all cases, to provide a fair comparison, we assume that the number of clusters r is known. Note that (Jalali, 2016) also requires knowledge of $\sum_{i=1}^{r} n_i^2$.

3.1. Running Time

In Fig. 3, we demonstrate the performance improvement afforded by Algorithm 1 versus clustering the full graph. We use the following parameters: $r = 2$, $p_1 = 0.4$, $p_2 = 0.8$,

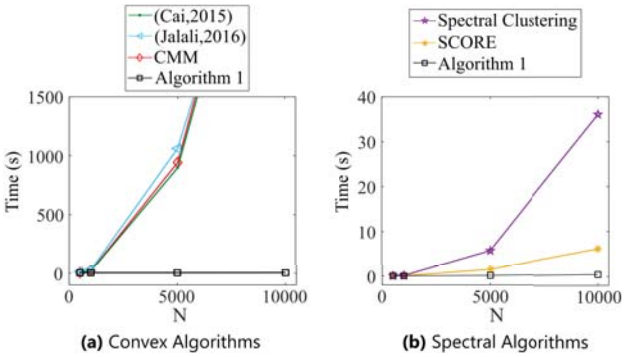


Fig. 3. Timings (averaged over three trials) showing the speed advantage of proposed algorithm using SPIN.

$q = 0.08$ $N' = 200$, $n_1 = N/2$, $n_2 = n_3 = N/4$ (in all cases, the algorithms achieve 100% success).

The plots show the time required to cluster the full graph using each algorithm. For comparison, we execute Algorithm 1 using SPIN. For the clustering step, in (a) we used (Jalali,2016), which had longer run times than (Cai,2015) and CMM. Likewise, in (b) we used Spectral Clustering, which was slower than SCORE. These results show a significant speed improvement of Algorithm 1 over full graph clustering.

Although the spectral algorithms run significantly faster than the convex algorithms, this does not necessarily mean that they are superior, e.g., in past experience we observed cases where the spectral methods yielded more errors than the convex algorithms. Furthermore, the spectral algorithms will still scale unfavorably as the graph size increases. For larger graphs, memory storage will become a limiting factor, regardless of the algorithm. The sketch-based approach can alleviate this problem by only requiring a sketch to be stored in memory, while keeping the full graph stored offline.

3.2. Clustering Unbalanced Graphs

In Fig. 4, we show the improved phase transition of Algorithm 1 when using SPIN over URS in the sampling step. The graph is generated from Data Model 1 with $N = 5000$, $r = 3$, and $q = 0.02$. It contains one dense cluster with probability $p_1 = 0.8$ and size $n_1 = 5000 - 2n_{sparse}$, and two sparse clusters with probabilities $p_2 = p_3 = 0.2$ and sizes $n_2 = n_3 = n_{sparse}$. Algorithm 1 is run using both URS and SPIN in the sampling step (in both cases, the CMM algorithm is used for the clustering step, although the other algorithms exhibit similar results as well). The probability of success is shown out of 10 trials. Success is defined as exact recovery, where every node in the full graph is correctly clustered.

To meet condition (2) when clustering the full graph, we need $n_{sparse} = 2222$, meaning the sparser clusters should be quite large. This same limitation is seen in URS, where even when 1000 samples are obtained, no successes occur until $n_{sparse} \geq 1400$. Meanwhile, SPIN achieves complete success with the same number of samples for $n_{sparse} = 400$. In fact, SPIN achieves success even when $n_{sparse} = 200$ with only 300 samples (only 6% of the nodes in the full graph).

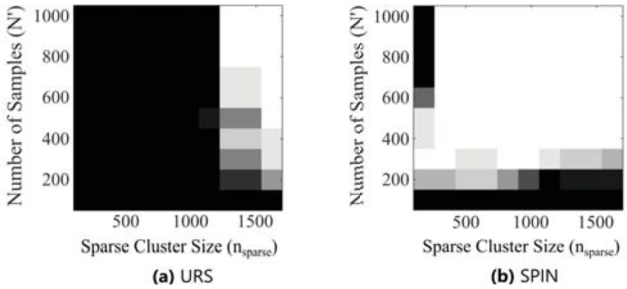


Fig. 4. Proposed algorithm with (a) URS and (b) SPIN sampling. White indicates success and black indicates failure.

In Fig. 5, we demonstrate the advantages described in Section 2: Algorithm 1 is not only faster than full-graph clustering, but it can also *reduce* the number of errors when SPIN sampling is used. Here, the parameters are the same as in Fig. 4, and the sketch is fixed at size $N' = 600$.

In Fig. 5(a), the green line with circle markers indicates the results when the generated graphs are clustered using (Cai, 2015). The other two lines show the results of Algorithm 1 using URS and SPIN, with (Cai, 2015) invoked in the sketch clustering step. In similar fashion, panels (b)-(e) compare results for the other four clustering algorithms applied to the full graph versus our proposed sketching-based approach.

Recall that in this scenario, the full graph has two sparse clusters and a single dense cluster. As expected, in Fig. 5(a)-(c) clustering of the full graph does not reach 100% success until n_{sparse} is between 1600 and 2200 (i.e., the sparse clusters are large). Meanwhile SPIN achieves full success anywhere between $n_{sparse} = 100$ and $n_{sparse} = 400$. In this range, the sparse clusters are very small in the *full* graph, which is opposite to the optimal condition (2). However, SPIN compensates for this by making these sparse clusters larger in the *sketch*, thus improving the phase transition. Sampling with URS, while also fast, yields worse phase transitions than SPIN, and in some cases no better than full-graph clustering. Fig. 5(d) and (e) show similar trends in the phase transitions of the spectral algorithms.

In Fig. 6, we consider smaller edge probabilities which yield a sparser graph: $p_1 = 0.1$, $p_2 = p_3 = 0.05$, $q = 0.005$. The remaining parameters are the same as for Fig. 5. In this regime, it is hard to correctly cluster each and every node (or impossible due to a disconnected graph), and so we use success rate to provide a more descriptive criteria for comparison. Success rate is defined as the proportion of nodes which are correctly clustered. For CMM, full graph clustering achieves $\approx 95\%$ success rate for $n_{sparse} = 1400$. Meanwhile, for Algorithm 1 with SPIN, the success rate remains above 99.8% for much smaller sizes of $n_{sparse} \geq 400$. For SCORE, SPIN exceeds a 99% success rate for $n_{sparse} \geq 500$, while full graph clustering does not attain this rate until $n_{sparse} \geq 800$.

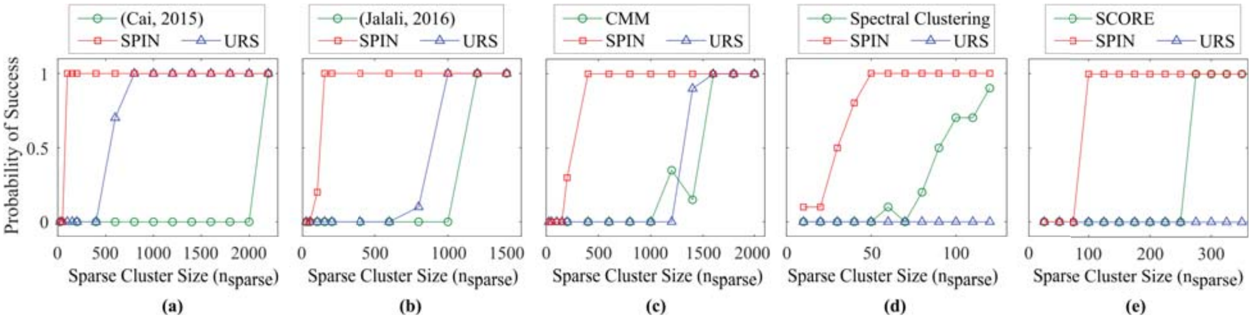


Fig. 5. Exact recovery probability for clustering the full graph versus using Algorithm 1 with URS and SPIN.

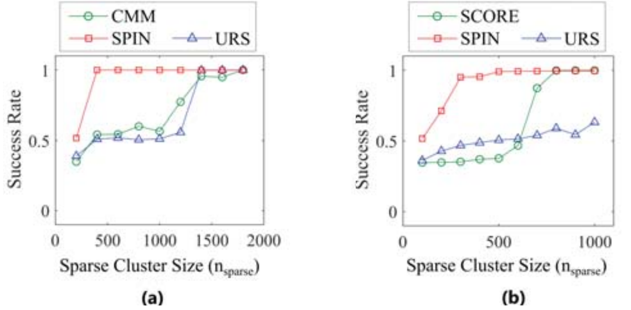


Fig. 6. Success rate for clustering the full graph versus using Algorithm 1 with URS and SPIN in sparse regime.

4. CONCLUSION

In this paper, we identified conditions under which clustering tends to yield consistently low error rates. This insight enabled the development of a scalable clustering algorithm which improves not only speed, but also success rate (under certain conditions). It is interesting to observe the same type of behavior across multiple disparate algorithms. More work is necessary to analytically pin down the precise requirements for improved clustering, and the exact influence of q.

Newer variants of the SBM, which introduce additional degrees of freedom beyond the heterogeneous SBM, have recently been the subject of intense research. In particular, the Degree Corrected SBM allows node degree to vary even at the level of individual nodes, and algorithms based on this model have been found to work exceedingly well on popular datasets [10]. Our work serves as an important step toward the use of sketch-based clustering in these models as well.

5. REFERENCES

[1] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[2] C. Nicolini, C. Bordier, and A. Bifone, "Community detection in weighted brain connectivity networks beyond the resolution limit," *arXiv e-prints*, Sept. 2016.

[3] Santo Fortunato and Darko Hric, "Community detection in networks: A user guide," *Phys. Rep.*, vol. 659, 2016.

[4] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, Feb. 2004.

[5] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu, "Clustering partially observed graphs via convex optimization," *J. Mach. Learn. Res.*, vol. 15, Jan. 2014.

[6] David F. Gleich and C. Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *KDD*, 2012, pp. 597–605.

[7] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt, "Stochastic blockmodels: First steps," *Soc. Netw.*, vol. 5, no. 2, pp. 109 – 137, 1983.

[8] Amin Jalali, Qiyang Han, Ioana Dumitriu, and Maryam Fazel, "Exploiting tradeoffs for exact recovery in heterogeneous stochastic block models," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 4871–4879. 2016.

[9] Mostafa Rahmani, Andre Beckus, Adel Karimian, and George Atia, "Scalable and Robust Community Detection with Randomized Sketching," *arXiv e-print*, 2019.

[10] Yudong Chen, Xiaodong Li, and Jiaming Xu, "Convexified modularity maximization for degree-corrected stochastic block models," *Ann. Statist.*, vol. 46, no. 4, pp. 1573–1602, Aug. 2018.

[11] T. Tony Cai and Xiaodong Li, "Robust and computationally feasible community detection in the presence of arbitrary outlier nodes," *Ann. Statist.*, vol. 43, 2015.

[12] R. K. Vinayak, S. Oymak, and B. Hassibi, "Sharp performance bounds for graph clustering via convex optimization," in *ICASSP*, May 2014, pp. 8297–8301.

[13] Jiashun Jin, "Fast community detection by SCORE," *Ann. Statist.*, vol. 43, no. 1, pp. 57–89, Feb. 2015.