Scalable Inference of Overlapping Communities

Prem Gopalan  David Mimno  Sean M. Gerrish  Michael J. Freedman  David M. Blei
{pgopalan,mimno,sgerrish,mfreed,blei}@cs.princeton.edu
Department of Computer Science
Princeton University
Princeton, NJ 08540

Abstract

We develop a scalable algorithm for posterior inference of overlapping communities in large networks. Our algorithm is based on stochastic variational inference in the mixed-membership stochastic blockmodel (MMSB). It naturally interleaves subsampling the network with estimating its community structure. We apply our algorithm on ten large, real-world networks with up to 60,000 nodes. It converges several orders of magnitude faster than the state-of-the-art algorithm for MMSB, finds hundreds of communities in large real-world networks, and detects the true communities in 280 benchmark networks with equal or better accuracy compared to other scalable algorithms.

1 Introduction

A central problem in network analysis is to identify communities, groups of related nodes with dense internal connections and few external connections [1, 2, 3]. Classical methods for community detection assume that each node participates in a single community [4, 5, 6]. This assumption is limiting, especially in large real-world networks. For example, a member of a large social network might belong to overlapping communities of co-workers, neighbors, and school friends.

To address this problem, researchers have developed several methods for detecting overlapping communities in observed networks. These methods include algorithmic approaches [7, 8] and probabilistic models [2, 3, 9, 10]. In this paper, we focus on the mixed-membership stochastic blockmodel (MMSB) [2], a probabilistic model that allows each node of a network to exhibit a mixture of communities. The MMSB casts community detection as posterior inference: Given an observed network, we estimate the posterior community memberships of its nodes.

The MMSB can capture complex community structure and has been adapted in several ways [11, 12]; however, its applications have been limited because its corresponding inference algorithms have not scaled to large networks [2]. In this work, we develop algorithms for the MMSB that scale, allowing us to study networks that were previously out of reach for this model. For example, we analyzed social networks with as many as 60,000 nodes. With our method, we can use the MMSB to analyze large networks, finding approximate posteriors in minutes with networks for which the original algorithm takes hours. When compared to other scalable methods for overlapping community detection, we found that the MMSB gives better predictions of new connections and more closely recovers ground-truth communities. Further, we can now use the MMSB to compute descriptive statistics at scale, such as which nodes bridge communities.

The original MMSB algorithm optimizes the variational objective by coordinate ascent, processing every pair of nodes in each iteration [2]. This algorithm is inefficient, and it quickly becomes intractable for large networks. In this paper, we develop stochastic optimization algorithms [13, 14] to fit the variational distribution, where we obtain noisy estimates of the gradient by subsampling the network.
Figure 1: Figure 1(a) shows communities (see §2) discovered in a co-authorship network of 1,600 researchers [16] by an a-MMSB model with 50 communities. The color of author nodes indicates their most likely posterior community membership. The size of nodes indicates bridgeness [17], a measure of participation in multiple communities. Figure 1(b) shows a graphical model of the a-MMSB. The prior over multinomial \( \pi \) is a symmetric Dirichlet distribution. Priors over Bernoulli \( \beta \) are Beta distributions.

Our algorithm alternates between subsampling from the network and adjusting its estimate of the underlying communities. While this strategy has been used in topic modeling [15], the MMSB introduces new challenges because the Markov blanket of each node is much larger than that of a document. Our simple sampler usually selects unconnected nodes (due to sparse real-world networks). We develop better sampling methods that focus more on the informative data in the network, e.g., the observed links, and thus make inference even faster.

2 Modeling overlapping communities

In this section, we introduce the assortative mixed-membership stochastic blockmodel (a-MMSB), a statistical model of networks that models nodes participating in multiple communities. The a-MMSB is a subclass of the mixed-membership stochastic blockmodel (MMSB) [2].

Let \( y \) denote the observed links of an undirected network, where \( y_{ab} = 1 \) if nodes \( a \) and \( b \) are linked and 0 otherwise. Let \( K \) denote the number of communities. Each node \( a \) is associated with community memberships \( \pi_a \), a distribution over communities; each community is associated with a community strength \( \beta_k \in (0, 1) \), which captures how tightly its members are linked. The probability that two nodes are linked is governed by the similarity of their community memberships and the strength of their shared communities.

We capture these assumptions in the following generative process of a network.

1. For each community \( k \), draw community strength \( \beta_k \sim \text{Beta}(\eta) \).
2. For each node \( a \), draw community memberships \( \pi_a \sim \text{Dirichlet}(\alpha) \).
3. For each pair of nodes \( a \) and \( b \),
   (a) Draw interaction indicator \( z_{a\rightarrow b} \sim \pi_a \).
   (b) Draw interaction indicator \( z_{a\leftarrow b} \sim \pi_b \).
   (c) Draw link \( y_{ab} \sim \text{Bernoulli}(r) \), where
   \[
   r = \begin{cases} 
   \beta_k & \text{if } z_{a\rightarrow b,k} = z_{a\leftarrow b,k} = 1, \\
   \epsilon & \text{if } z_{a\rightarrow b} \neq z_{a\leftarrow b}.
   \end{cases}
   \] (1)

\(^1\)We use a subclass of the MMSB models that is appropriate for community detection in undirected networks. In particular, we assume assortativity, i.e., that links imply that nodes are similar. We call this special case the assortative MMSB or a-MMSB. In §2 we argue why the a-MMSB is more appropriate for community detection than the MMSB. We note that our algorithms are immediately applicable to the MMSB as well.
Figure 1(b) represents the corresponding joint distribution of hidden and observed variables. The a-MMSB defines a single parameter \( \epsilon \) to govern inter-community links. This captures assortativity—if two nodes are linked, it is likely that the latent community indicators were the same.

The full MMSB differs from the a-MMSB in that the former uses one parameter for each of the \( K^2 \) ordered pairs of communities. When the full MMSB is applied to undirected networks, two hypotheses compete to explain a link between each pair of nodes: either both nodes exhibit the same community or they are in different communities that link to each other.

We analyze data with a-MMSB via the posterior distribution over latent variables \( p(\pi_{1:N}, z, \beta_{1:K} | y, \alpha, \eta) \). The posterior lets us form a predictive distribution of unseen links and measure latent network properties of the observed data. The posterior over \( \pi_{1:N} \) represents the community memberships of the nodes, and the posterior over the interaction indicator variables \( z \) identifies link communities in the network [8]. For example, in a social network one member’s link to another might arise because they are from the same high school while another might arise because they are co-workers. With an estimate of this latent structure, we can characterize the network in interesting ways. In Figure 1(a), we sized author nodes according to their expected posterior bridgeness [17], a measure of participation in multiple communities (see §5).

3 Stochastic variational inference

Our goal is to compute the posterior distribution \( p(\pi_{1:N}, z, \beta_{1:K} | y, \alpha, \eta) \). Exact inference is intractable, so we use variational inference [18]. Traditional variational inference is a coordinate ascent algorithm. In the context of the MMSB (and the a-MMSB), coordinate ascent iterates between analyzing all \( O(N^2) \) node pairs and updating the community memberships of the \( N \) nodes [2]. In this section, we will derive a stochastic variational inference algorithm. Our algorithm iterates between sampling random pairs of nodes and updating node memberships. This avoids the per-iteration \( O(N^2) \) computation and allows us to scale to large networks.

3.1 Variational inference in a-MMSB

In variational inference, we define a family of distributions over the hidden variables \( q(\beta, \pi, z) \) and find the member of that family that is closest to the true posterior. (Closeness is measured with KL divergence.) We use the mean-field family, under which each variable is endowed with its own distribution and its own variational parameter. This allows us to tractably optimize the parameters to find a local minimum of the KL divergence. For the a-MMSB, the variational distributions are

\[
q(z_{a \rightarrow b} = k) = \phi_{a \rightarrow b, k}; \quad q(\pi_a) = \text{Dirichlet}(\pi_a; \gamma_p); \quad q(\beta_k) = \text{Beta}(\beta_k; \lambda_k).
\]

The posterior over link community assignments \( z \) is parameterized by the per-interaction memberships \( \phi \), the node community distributions \( \pi \) by the community memberships \( \gamma \), and the link probability \( \beta \) by the community strengths \( \lambda \). Notice that \( \lambda \) is of dimension \( K \times 2 \), and \( \gamma \) is of dimension \( N \times K \).

Minimizing the KL divergence between \( q \) and the true posterior is equivalent to optimizing an evidence lower bound (ELBO) \( \mathcal{L} \), a bound on the log likelihood of the observations. We obtain this bound by applying Jensen’s inequality [18] to the data likelihood. The ELBO is

\[
\log p(y | \alpha, \beta) \geq \mathcal{L}(y, \phi, \gamma, \lambda) \triangleq \mathbb{E}_q[\log p(y, \pi, z | \alpha, \eta)] - \mathbb{E}_q[\log q(\beta, \pi, z)].
\]

The right side of Eq. 3 factorizes to

\[
\mathcal{L} = \sum_k \mathbb{E}_q[\log p(\beta_k | \eta_k)] - \sum_k \mathbb{E}_q[\log q(\beta_k | \lambda_k)] + \sum_n \mathbb{E}_q[\log p(\pi_n | \alpha)] - \sum_n \mathbb{E}_q[\log q(\pi_n | \gamma_n)] + \sum_{a,b} \mathbb{E}_q[\log p(z_{a \rightarrow b} | \pi_a)] + \mathbb{E}_q[\log p(z_{a \rightarrow b} | \pi_b)] - \sum_{a,b} \mathbb{E}_q[\log q(z_{a \rightarrow b} | \phi_{a \rightarrow b})] - \mathbb{E}_q[\log q(z_{a \rightarrow b} | \phi_{a \rightarrow b})] + \sum_{a,b} \mathbb{E}_q[\log p(y_{ab} | z_{a \rightarrow b}, z_{a \rightarrow b}, \beta)]
\]

Notice the first line in Eq. 4 contains summations over communities and nodes; we call these global terms. They relate to the global variables, which are the community strengths \( \lambda \) and per-node memberships \( \gamma \). The remaining lines contain summations over all node pairs, which we call local terms. They depend on both the global and local variables, the latter being the per-interaction memberships \( \phi \). This distinction is important in the stochastic optimization algorithm.
3.2 Stochastic optimization

Our goal is to develop a variational inference algorithm that scales to large networks. We will use stochastic variational inference [14], which optimizes the ELBO with respect to the global variational parameters using stochastic gradient ascent. Stochastic gradient algorithms follow noisy estimates of the gradient with a decreasing step-size. If the expectation of the noisy gradient is equal to the gradient and if the step-size decreases according to a certain schedule, then we are guaranteed convergence to a local optimum [13]. Subsampling the data to form noisy gradients scales inference as we avoid the expensive all-pairs sums in Eq. 4.

Stochastic variational inference is a coordinate ascent algorithm that iteratively updates local and global parameters. For each iteration, we first subsample the network and compute optimal local parameters for the sample, given the current settings of the global parameters. We then update the global parameters using a stochastic natural gradient\(^2\) computed from the subsampled data and local parameters. We call the first phase the local step and the second phase the global step [14].

The selection of subsamples in each iteration provides a way to plug in a variety of network subsampling algorithms. However, to maintain a correct stochastic optimization algorithm of the variational objective, the subsampling method must be valid. That is, the natural gradients estimated from the subsample must be unbiased estimates of the true gradients.

The global step. The global step updates the global community strengths \(\lambda\) and community memberships \(\gamma\) with a stochastic gradient of the ELBO in Eq. 4. Eq. 4 contains summations over all \(O(N^2)\) node pairs. Now consider drawing a node pair \((a,b)\) at random from a population distribution \(g(a,b)\) over the \(M = N(N - 1)/2\) node pairs. We can rewrite the ELBO as a random function of the variational parameters that includes the global terms and the local terms associated only with \((a,b)\). The expectation of this random function is equal in objective to Eq. 4. For example, the fourth term in Eq. 4 is rewritten as

\[
\sum_{a,b} \mathbb{E}_q[\log p(y_{ab}|z_{a\rightarrow b}, z_{a\leftarrow b}, \beta)] = \mathbb{E}_g[\frac{1}{g(a,b)} \mathbb{E}_q[\log p(y_{ab}|z_{a\rightarrow b}, z_{a\leftarrow b}, \beta)]]
\]

Evaluating the rewritten Eq. 4 for a node pair sampled from \(g\) gives a noisy but unbiased estimate of the ELBO. Following [15], the stochastic natural gradients computed from a sample pair \((a,b)\) are

\[
\frac{\partial \gamma^t_{a,k}}{\partial \gamma^t_{a,k}} = \alpha_k + \frac{1}{g(a,b)} \gamma^t_{a\rightarrow b,k} - \gamma^t_{a,k}
\]

\[
\frac{\partial \lambda^t_{k,i}}{\partial \gamma^t_{a,k}} = \eta_{k,i} + \frac{1}{g(a,b)} \phi_{a\rightarrow b,k} \phi_{a\leftarrow b,k} \cdot y_{ab,i} - \lambda^t_{k,i}
\]

where \(y_{ab,0} = y_{ab}\), and \(y_{ab,1} = 1 - y_{ab}\). In practice, we sample a “mini-batch” \(S\) of pairs per update, to reduce noise.

The intuition behind the above update is analogous to Online LDA [15]. When a single pair \((a,b)\) is sampled, we are computing the setting of \(\gamma\) that would be optimal (given \(\phi^t\)) if our entire network were a multigraph consisting of the interaction between \(a\) and \(b\) repeated \(1/g(a,b)\) times.

Our algorithm has assumed that the subset of node pairs \(S\) are sampled independently. We can relax this assumption by defining a distribution over predefined sets of links. These sets can be defined using prior information about the pairs, such as network topology, which lets us take advantage of more sophisticated sampling strategies. For example, we can define a set for each node, with each set consisting of the node’s adjacent links or non-links. Each iteration we set \(S\) to one of these sets sampled at random from the \(N\) sets.

In order to ensure that set-based sampling results in unbiased gradients, we specify two constraints on sets. First, we assume that the union of these sets \(s\) is the total set of all node pairs, \(U: U = \cup_i s_i\). Second, we assume that every pair \((a,b)\) occurs in some constant number of sets \(c\) and that \(c \geq 1\). With these conditions satisfied, we can again rewrite Eq. 4 as the sum over its global terms and an expectation over the local terms. Let \(h(t)\) be a distribution over predefined sets of node pairs.

For example, the fourth term in Eq. 4 can be rewritten using

\[
\sum_{a,b} \mathbb{E}_q[\log p(y_{ab}|z_{a\rightarrow b}, z_{a\leftarrow b}, \beta)] = \mathbb{E}_h[\frac{1}{h(t)} \sum_{(a,b) \in s_t} \mathbb{E}_q[\log p(y_{ab}|z_{a\rightarrow b}, z_{a\leftarrow b}, \beta)]]
\]

\(^2\)Stochastic variational inference uses natural gradients [19] of the ELBO. Computing natural gradients (along with subsampling) leads to scalable variational inference algorithms [14].
Our algorithm allows us flexibility around how the subset of pairs is sampled, as long as the expectation of the stochastic gradient is equal to the true gradient. There are several ways we can take advantage of this. We can sample based on informative pairs of nodes, i.e., ones that help us better assess the community structure. We can also subsample to make data processing easier, for example, to accommodate a stream of links. Finally, large, real-world networks are often sparse, with links...
accounting for less than 0.1% of all node pairs (see Figure 2). While we should not ignore non-links, it may help to give preferential attention to links. These intuitions are captured in the following four subsampling methods.

**Random pair sampling.** The simplest method is to sample node pairs uniformly at random. This method is an instance of independent pair sampling, with $g(a, b)$ (used in Eq. 5) equal to $\frac{1}{N( N - 1)/2}$.

**Random node sampling.** This method focuses on local neighborhoods of the network. A set consists of all the pairs that involve one of the $N$ nodes. At each iteration, we sample a set uniformly at random from the $N$ sets, so $b(t)$ (used in Eq. 8) is $\frac{1}{N}$. Since each pair involves two nodes, each link appears in two sets, so $c$ (also used in Eq. 8) is 2. By reweighting the terms corresponding to pairs in the sampled set, we maintain a correct stochastic optimization.

**Stratified random pair sampling.** This method samples links independently, but focuses on observed links. We divide the $M$ node pairs into two strata: links and non-links. Each iteration either samples a mini-batch of links or samples a mini-batch of non-links. If the non-link stratum is sampled, and $N_0$ is the estimated total number of non-links, then

$$g(a, b) = \begin{cases} \frac{1}{N_0} & \text{if } y_{ab} = 0, \\ 0 & \text{if } y_{ab} = 1 \end{cases}$$ (11)

The population distribution when the link strata is sampled is symmetric.

**Stratified random node sampling.** This method combines set-based sampling and stratified sampling to focus on observed links in local neighborhoods. For each node we define a “link set” consisting of all its links, and $m$ “non-link sets” that partition its non-links. Since the number of non-links associated with each node is usually large, dividing them into many sets allows the computation in each iteration to be fast. At each iteration, we first select a random node and either select its link set or sample one of its $m$ non-link sets, uniformly at random. To compute Eq. 8 we define the number of sets that contain each pair, $c = 2$, and the population distribution over sets

$$h(t) = \begin{cases} \frac{1}{2N} & \text{if } t \text{ is a link set,} \\ \frac{1}{2Nm} & \text{if } t \text{ is a non-link set.} \end{cases}$$ (12)

Stratified random node sampling gives the best gains in convergence speed (see §5).

4 Related work

Newman et al. [3] described a model of overlapping communities in networks (“the Poisson model”) where the number of links between two nodes is a Poisson random variable. Recently, other researchers have proposed latent feature network models [20, 21] that compute the probabilities of links based on the interactions between binary features associated with each node. Efficient inference algorithms for these models exploit model-specific approximations that allow scaling in the number of links. These ideas do not extend to the MMSB. Further, these algorithms do not explicitly leverage network sampling. In contrast, the ideas in Algorithm 1 apply to a number of models [14]. It subsamples both links and non-links in an inner loop for scalability.

Other scalable algorithms include Clique Percolation (CP) [7] and Link Clustering (LC) [8], which are based on heuristic clique-finding and hierarchical clustering, respectively. These methods are fast in practice, although the underlying problem is NP-complete. Further, because they are not statistical models, there is no clear mechanism for predicting new observations or model checking.

In the next section we will compare our method to these alternative scalable methods. Compared to the Poisson model, we will show that the MMSB gives better predictions. Compared to CP and LC, which do not provide predictions, we will show that the MMSB more reliably recovers the true community structure.

5 Empirical study

In this section, we evaluate the efficiency and accuracy of Stochastic a-MMSB (AM). First, we evaluate its efficiency on 10 real-world networks. Second, we demonstrate that stratified sampling
perplexity

Figure 2: Network datasets. $N$ is the number of nodes, $K^{\text{max}}$ is the maximum number of communities analyzed and $d$ is the percent of node pairs that are links. The time until convergence for the different methods are $T^\text{tstoch}$ and $T^\text{batch}$, while the time required for 90% of the perplexity at a-MMSB’s convergence is $T^\text{tstoch}_{90\%}$.

<table>
<thead>
<tr>
<th>DATA SET</th>
<th>$N$</th>
<th>$K^{\text{max}}$</th>
<th>$d(%)$</th>
<th>$T^\text{tstoch}$</th>
<th>$T^\text{tstoch}_{90%}$</th>
<th>$T^\text{batch}$</th>
<th>TYPE</th>
<th>SOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>US-AIR</td>
<td>1.1K</td>
<td>19</td>
<td>1.2</td>
<td>1.7m</td>
<td>3.4m</td>
<td>40.5m</td>
<td>TRANS.</td>
<td>[22]</td>
</tr>
<tr>
<td>NETSCIENCE</td>
<td>1.6K</td>
<td>100</td>
<td>0.3</td>
<td>7.2m</td>
<td>11.7m</td>
<td>2.2h</td>
<td>COLLAB. [16]</td>
<td></td>
</tr>
<tr>
<td>RELATIVITY</td>
<td>5.2K</td>
<td>300</td>
<td>0.1</td>
<td>2.3h</td>
<td>4h</td>
<td>&gt; 20h</td>
<td>COLLAB. [23]</td>
<td></td>
</tr>
<tr>
<td>HEP-TH</td>
<td>9.9K</td>
<td>32</td>
<td>0.05</td>
<td>7.3h</td>
<td>8.7h</td>
<td>&gt; 67h</td>
<td>COLLAB. [23]</td>
<td></td>
</tr>
<tr>
<td>HEP-PH</td>
<td>12K</td>
<td>32</td>
<td>0.16</td>
<td>36m</td>
<td>2.8h</td>
<td>&gt; 67h</td>
<td>COLLAB. [23]</td>
<td></td>
</tr>
<tr>
<td>ASTRO-PH</td>
<td>18.7K</td>
<td>32</td>
<td>0.11</td>
<td>13.8h</td>
<td>22.1h</td>
<td>&gt; 67h</td>
<td>COLLAB. [23]</td>
<td></td>
</tr>
<tr>
<td>HEP-TH2</td>
<td>27.8K</td>
<td>512</td>
<td>0.09</td>
<td>8d</td>
<td>10.3d</td>
<td>-</td>
<td>CITE [23],[24]</td>
<td></td>
</tr>
<tr>
<td>ENRON</td>
<td>37K</td>
<td>158</td>
<td>0.03</td>
<td>1.5d</td>
<td>2.5d</td>
<td>-</td>
<td>EMAIL [25]</td>
<td></td>
</tr>
<tr>
<td>COND-MAT</td>
<td>40.4K</td>
<td>300</td>
<td>0.02</td>
<td>4.6d</td>
<td>5.2d</td>
<td>-</td>
<td>COLLAB. [26]</td>
<td></td>
</tr>
<tr>
<td>BRIGHTKITE</td>
<td>58.2K</td>
<td>64</td>
<td>0.01</td>
<td>8d</td>
<td>9.5d</td>
<td>-</td>
<td>SOCIAL [27]</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Stochastic a-MMSB (with random pair sampling) scales better and finds communities as good as batch a-MMSB on real networks (top). Stratified random node sampling is an order of magnitude faster than other sampling methods on the hep-ph, astro-ph and hep-th2 networks (bottom).

significantly improves convergence speed on real networks. Third, we compare our algorithm with leading algorithms in terms of accuracy on benchmark graphs and ability to predict links.

We measure convergence by computing the link prediction accuracy on a validation set. We set a validation and a test set, each having 10% of the network links and an equal number of non-links (see §3.2). We approximate the probability that a link exists between two nodes using posterior expectations of $\beta$ and $\pi$. We then calculate perplexity, which is the exponential of the average predictive log likelihood of held-out node pairs.

For random pair and stratified random pair sampling, we use a mini-batch size $S = N/2$ for graphs with $N$ nodes. For the stratified random node sampling, we set the number of non-link sets $m = 10$. Based on experiments, we set the parameters $\kappa = 0.5$ and $\tau_0 = 1024$. We set hyperparameters $\alpha = 1/K$ and $\{\eta_1, \eta_0\}$ proportional to the expected number of links and non-links in each community. We implemented all algorithms in C++.

Comparing scalability to batch algorithms. AM is an order of magnitude faster than standard batch inference for a-MMSB [2]. Figure 2 shows the time to convergence for four networks of varying types, node sizes $N$ and sparsity $d$. Figure 3 shows test perplexity for batch vs. stochastic inference. For many networks, AM learns rapidly during the early iterations, achieving 90% of the converged perplexity in less than 70% of the full convergence time. For all but the two smallest networks, batch inference did not converge within the allotted time. AM lets us efficiently fit a mixed-membership model to large networks.

Comparing sampling methods. Figure 3 shows that stratified random node sampling converges an order of magnitude faster than random node sampling. It is statistically more efficient because the observations in each iteration include all the links of a node and a random sample of its non-links.

---

[1] Following [1], we treat the directed citation network hep-th2 as an undirected network.
D.M. Blei is supported by ONR N00014-11-1-0651, NSF CAREER 0745520, AFOSR FA9550-09-1-0668, the Alfred P. Sloan foundation, and a grant from Google.

\[ \text{AM, PM, LC, CP} \]

We generated 280 networks for combinations of these parameters: \#nodes \in \{400\}; \#communities \in \{5, 10\}; \#nodes with at least 3 overlapping communities \in \{100\}; community sizes \in \{equal, unequal\}, when unequal, the community sizes are in the range \{N/10, 2N\}; average node degree \in \{0.1N, 0.15N, ..., 0.35N, 0.4N\}, the maximum node degree = 2 \times \text{average node degree}; \% links of a node that are noisy \in \{0, 0.1\}; random runs \in \{1, 2, 3, 4, 5\}.

\[ \text{AM finds a solution per clique size; LC finds a solution per threshold at which the dendrogram is cut} \] in steps of 0.1 from 0 to 1; PM and a-MMSB find a solution \forall K \in \{k', k' + 10\} where k' is the true number of communities—increasing by 10 allows for potentially a larger number of communities to be detected; a-MMSB also finds a solution for each of random pair or stratified random pair sampling methods with the hyperparameters \text{\eta} set to the default or set to fit dense clusters.
References


