Statistica Sinica Preprint No: SS-2022-0411	
Title	Scalable Community Detection in Massive Networks using Aggregated Relational Data
Manuscript ID	SS-2022-0411
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0411
Complete List of Authors	Timothy Jones, Owen G. Ward, Yiran Jiang, John Paisley and Tian Zheng
Corresponding Authors	Tian Zheng
E-mails	tz33@columbia.edu
Notice: Accepted version subject to English editing.	

Scalable Community Detection in Massive Networks using Aggregated Relational Data

Timothy Jones[1], Owen G. Ward[2], Yiran Jiang[3],

John Paisley[1] and Tian Zheng[1*]

[1] *Columbia University*

[2] *Simon Fraser University*

[3] *Purdue University*

[*] *Corresponding Author. tian.zheng@columbia.edu*

Abstract: The mixed membership stochastic blockmodel (MMSB) is a popular Bayesian network model for community detection. Fitting such large Bayesian network models quickly becomes computationally infeasible when the number of nodes grows into hundreds of thousands and millions. In this paper we propose a novel mini-batch strategy based on aggregated relational data that leverages nodal information to fit MMSB to massive networks. We describe a scalable inference method that can utilise nodal information that often accompanies real-world networks. Conditioning on this extra information leads to a model that admits a parallel stochastic variational inference algorithm,

utilising stochastic gradients of bipartite graphs formed from aggregated network ties between node subpopulations. We apply our method to a citation network with over two million nodes and 25 million edges, capturing explainable structure in this network. Our method recovers parameters and achieves better convergence on simulated networks generated according to the MMSB.

Key words and phrases: Community Detection, Network Data, Aggregated Relational Data, Mixed-Membership.

1. Introduction

Relational data between objects is commonly represented by a graph or network that encodes pairwise interactions, and has been studied across the natural and social sciences. Due to this prevalence in modern applications, analysis of such data is vitally important. Among the tasks commonly considered for such data, community detection stands out as being one of the most crucial for practitioners and has been widely studied. Community detection algorithms aim to identify groups of nodes that exhibit similar connective behaviors. More specifically, nodes in networks often cluster into small communities, where nodes within a community show a similar propensity to form ties with other nodes (Bickel and Chen, 2009). These

clusters are often assortative, where nodes within a community interact more than nodes in different communities (Fortunato and Hric, 2016). Identifying such clusters of nodes often provides important scientific insights into the processes underlying the realised network, and can also be used for further analysis such as link prediction and node classification (Soundarajan and Hopcroft, 2012; Ward et al., 2021).

The mixed membership stochastic blockmodel (MMSB) is a popular model-based method for community detection in networks(Airoldi et al., 2008). An extension of the widely used stochastic block model (Holland et al., 1983; Nowicki and Snijders, 2001), the MMSB relaxes the assumption that each node belongs to exactly one community, instead allowing the node to belong to different communities for different interactions. By allowing nodes to belong to multiple communities in varying amounts, this model can capture more realistic features of networks (Airoldi et al., 2008).

A fundamental difficulty in analyzing networks with models such as the MMSB is the computational burden of fitting these flexible models to large real-world networks. The number of parameters grows quadratically in the number of nodes in the network. To mitigate this issue, many algorithms take advantage of the sparseness found in real-world networks by avoiding computations using all node pairs. In particular, Gopalan and Blei (2013) developed a stochastic variational inference (SVI) algorithm with a sub-sampling scheme that only uses links.

The computational burden of fitting network models may be reduced by

including nodal information. Although the MMSB models connectivity only, real-world networks often have rich nodal data that can help with model convergence. Additionally, modeling nodal covariates reveal an interesting interplay between "content" and "connections" in networks. For instance, Tan et al. (2016) modeled both text and links in a citation network to measure the topic-neutral impact of scientific articles.

Previous procedures for the MMSB have performed inference using the fully observed adjacency matrix, including sampling links to scale this to large networks. These methods aim to infer the global network structure by sampling edges from the network in a possibly online fashion, learning the parameters using gradient steps based on these small samples. One obstacle to the use of these methods is the inherent difficulty in sampling from networks. Such sampling schemes may only provide limited information about a subset of the parameters in the model. For example for models describing community structure, it may be challenging to generate a sample of nodes containing all possible pairs of community interactions. This can lead to difficulties in learning stable estimates and such models may not converge efficiently.

In this paper we propose a novel mini-batch strategy based on aggregated relational data that leverages nodal information to fit MMSB to massive networks. Instead of forming a subgraph by sampling nodes or links, we create counts of links called aggregated relational data (ARD) for selected nodes in a mini-batch. These represent aggregate counts of interactions

between a node and a specified subpopulation, derived from available nodal information. These node subpopulation counts can be more readily analyzed than the original full adjacency matrix. These subpopulations could be formed utilising nodal information (such as the journals academic papers appear in). The weighted bipartite graph formed by ARD retains more information from the original graph and can be used to estimate the parameters of an MMSB directly using the ARD data enumerated from the sampled mini-batches. We apply Bayesian modeling and derive a variational approximation algorithm for ARD that is used for each mini-batch. We show our strategy enjoys good computational efficiency and recovers true community structure in simulation experiments. A real-data analysis reveals insightful and meaningful structure from a citation network of scientific publications.

This paper is organized as follows. Section 2 provides a review of the MMSB model for network data, previous inference schemes for MMSB, and ARD models for network data. Section 3 develops our proposed extension, considering aggregate relational data for the mixed membership stochastic block model (ARDMMSB). In particular, Section 3.2 introduces a new variational algorithm for approximate posterior inference for this model, describing important practical considerations of our implementation. Section 4 evaluates the performance of our proposed method using simulated data while in Section 5 we examine the performance of our proposed for a large citation network. In Section 6 we summarize our contributions and describe

Mixed membership stochastic blockmodel (MMSB)

1. For each entry of the blockmatrix B, draw probability $B_{mn} \sim \text{Beta}(a_{mn}, b_{mn})$.
2. For each node $n = 1, \ldots, N$, draw mixed membership vector $\pi_n \sim \text{Dirichlet}(\alpha)$.
3. For each node pair (i, j), draw
   - the sender membership indicator $s_{ij} \sim \text{Multi}(1, \pi_i)$
   - the receiver membership indicator $r_{ij} \sim \text{Multi}(1, \pi_j)$.
   - the resulting interaction $\delta_{ij} \sim \text{Bernoulli}(B_{s_{ij}r_{ij}})$.

Figure 1: Data Generating Processes for the MMSB

potential future applications of our model.

2. Background

In this section, we review the Mixed-Membership Stochastic Blockmodel (MMSB) and past approaches for posterior inference for this model, before providing a review of the idea of ARD.

2.1 Review of MMSB

The MMSB is a mixed membership latent variable model for a directed graph that detects overlapping communities within a network. It assumes there are K latent communities (groups). The global connectivity is stored

in a block matrix B, where B_{lm} represents the probability of a directed link from a node that assumes membership in group l to a node in group m. Each node belongs to the K groups with varying degrees of affiliation encoded in a K-dimensional probability vector π_i. The membership vectors are drawn from a Dirichlet(α) prior. Each node i may assume different memberships when interacting with different nodes. For each (i, j), membership indicator vectors for the sender (s_{ij}) and receiver (r_{ij}) are drawn from a multinomial based on their membership vectors. If the lth and mth elements of s_{ij} and r_{ij} are ones respectively, the value of the interaction δ_{ij} is sampled from Bernoulli(B_{lm}). Figure 1 outlines this data generating process.

2.2 Review of prior posterior inference approaches for MMSB

Since the MMSB models each directed edge, it quickly becomes computationally infeasible as the number of nodes grows. A popular approach to fit large Bayesian models is stochastic variational inference (Hoffman et al., 2013). Variational approximation algorithms turn a Bayesian inference problem into an optimization procedure. The researcher first posits a family of approximating densities and the algorithm seeks to find the density that minimizes the Kullback-Leiber divergence to the exact posterior (Blei et al., 2017). Stochastic variational inference implements gradient based optimization that combines natural gradients and stochastic optimization. The algorithm maintains a current estimate of the global variational parameters, while repeatedly subsampling data. It uses the current global

parameters to compute the optimal local parameters for the subsampled data, and adjusts the current global parameters appropriately.

Implementing a stochastic variational algorithm on the MMSB requires subsampling network data, a difficult problem, especially on sparse networks commonly found in practice. If one subsamples nodes and keeps the links between them, there will hardly be any links in the subgraph, making updates to parameters extremely noisy (Ma and Zheng, 2017). Gopalan and Blei (2013) instead proposed a sampling strategy that is guaranteed to sample a large number of links often. For instance, their stratified random node sampling selects a node and partitions its node pairs into links and many sets of non-links. The stochastic variational inference selects a node at random and picks a link set or one of its non-link sets half the time. By up-weighting the probability of selecting links, they ensure efficient updates to their global parameters by not allowing their subsampled graphs to be almost filled entirely of non-links at each iteration.

Fitting a MMSB can be improved by developing a sampling procedure that creates subgraphs which more closely resemble the original graph. Although Gopalan and Blei (2013) create subgraphs with many links, the resulting subgraph does not carry the same properties from the full network. They rely on the subgraph to get noisy, unbiased updates for the global parameters. Although their sampling strategy is superior to naively subsampling nodes, the global parameters recovered from the subgraph will not closely match those in the full graph. Instead, they rely on stochastic

optimization to get closer with each iteration towards the truth. Thus, for a fixed node, their procedure necessitates the need to periodically sample subgraphs that contain links involving that fixed node to efficiently update its variational parameter.

2.3 Review of ARD

We show in this paper that employing node-centric aggregated relational data (ARD) gives an efficient solution to the above issue. ARD can be used to create a multigraph using a subsample of nodes that can estimate the global parameters directly, eliminating the need to continuously update the global parameters with each sampled subgraph. Aggregated relational data is commonly used in the study of social networks (DiPrete et al., 2011). Sociologists are interested in the connections between people and gather information through sample surveys (McCarty et al., 2001). Ideally, a respondent would reveal some personal information and enumerate all the persons he or she knows (Shelley et al., 1995). However, this is not always feasible since people may be reluctant to report membership to a certain group due to social pressure or stigma. Additionally, enumerating one's network of friends and acquaintances is not practical especially since personal network sizes run to the hundreds of individuals. To get around this obstacle, survey enumerators ask questions of the form "How many X's do you know?," where X represents a subpopulation of interest. For example, X can be the subpopulation of people with first name Michael. Rather than

having data on the connections individually, we get the total number of links the respondent has with Michaels. This idea can be extended to other networks with well-defined subpopulations.

McCormick and Zheng (2015) modeled ARD as a partially observed full network. By first positing a model of the complete graph and deriving the model for the ARD, they were able to establish a framework that yielded an explicit relationship between complete graph features and the sampled data. This connection illuminates how inferences made on the smaller graph effects inferences made on the complete graph.

3. ARDMMSB Model and Algorithm

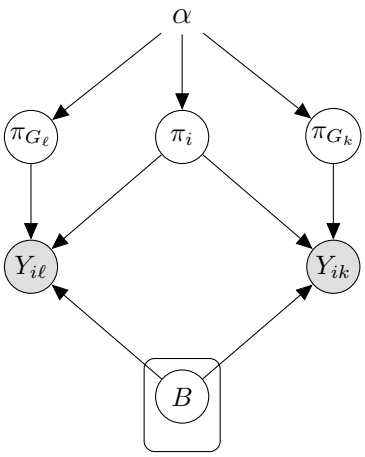
In this paper, we address the computational challenge posed by fitting massive real-world networks by proposing ARD to construct bipartie graphs that carry information about full network features. Ideally, we want inferences made for this bipartite graph to carry over to inferences made on the complete graph. To form ARD, we leverage background information on the dataset to create subpopulations, if it is available. Nodes within a subpopulation should have similar memberships, while the subpopulations themselves ideally should be spread across regions of the membership space. As an example, in a citation network, one can use the journals that the papers were published in as subpopulations. Intuitively, papers in the same journal should have similar community memberships. In this case, for each sampled node, one would summarize the number of citations to each journal

to form the ARD bipartite graph. We will show below that this method provides stable estimates of the blockmatrix B and the subsampled nodes' community memberships. However, other methods, such as constructing subpopulations based on papers which share keywords, is also possible.

In what follows, we model the entire full network as a MMSB, create random subgraphs from ARD mini-batches, and model the aggregated links to infer the blockmatrix B and membership vectors for each node in the complete graph. Central to this algorithm is a model for the ARD bipartite graph assuming the underlying full network is generated from an MMSB. We first introduce this model and then describe our variational inference procedure for ARD, before giving practical remarks related to implementation.

3.1 Proposed model: ARDMMSB

In a complete graph, we observe an $N \times N$ adjacency matrix $\boldsymbol{\delta}$ where $\delta_{ij} = 1$ if there is a directed edge from i to j and 0 otherwise. We assume that there are K latent communities present in this network, where the probability of an edge between these communities is determined by a $K \times K$ block matrix B, where B_{lm} is the probability of an edge from a node in community l to a node in community m. In the MMSB, the propensity to form ties is modeled conditionally on π_i and π_j, the memberships of sender i and receiver j. The probability is calculated by integrating over the sender and receiver indicators s_{ij} and r_{ij},



Generative process of ARDMMSB

1. For each entry of the blockmatrix B, draw probability $B_{ij} \sim \text{Beta}(a_{ij}, b_{ij})$.
2. For each node $n = 1, \ldots, n_b$ in minibatch b, draw $\pi_n \sim \text{Dirichlet}(\alpha)$.
3. For each subpopulation $k = 1, \ldots, \kappa$, draw $\eta_k \sim \text{Dirichlet}(\alpha)$.
4. For each node and subpopulation pair (i, k), draw $y_{ik} \sim \text{Poisson}(N_k \pi_i^T B \eta_k)$.

Figure 2: Left: Graphical representation of a two-node segment of the ARD network. The complete model contains y_{ik} for every node, subpopulation pair. Circles denote variables and observed variables are shared. The plates contain variables to be replicated. Right: Data Generating Process for Aggregated Relational Data for MMSB.

$$P(\delta_{ij} = 1 | \pi_i, \pi_j) = \int P(\delta_{ij} = 1 | s_{ij}, r_{ij}) P(s_{ij} | \pi_i) P(r_{ij} | \pi_j) ds_{ij} dr_{ij}$$
$$= \pi_i^T B \pi_j.$$

Gopalan and Blei (2013) infer these global parameters $B, \boldsymbol{\pi}$ from subgraphs formed from the original graph. While these subgraphs can be constructed to contain many links, it may still be challenging to gener-

ate reasonably sized subgraphs which can infer these global parameters well. For example, when learning the off diagonal elements of B, the inter group connection probabilities, there may only be a limited number of edges with which to perform inference. This can lead to noisy estimates of these parameters, and may make convergence of the inference scheme challenging.

Instead of observing the connections between each pairs of nodes (i, j), we only observe aggregated counts of links between each node i and each of κ subpopulations, such that $y_{ik} = \sum_{j \in G_k} \delta_{ij}$ where G_k is the kth subpopulation, with $k = 1, \ldots, \kappa$. Conditional on the community memberships π_i and $\{\pi_j\}_{j \in G_k}$, $\{\delta_{ij}\}_{j \in G_k}$ are independent Bernoulli random variables, each with a small probability of success. It is then reasonable to assume,

$$y_{ik} \sim \text{Poisson}(\lambda_{ik})$$
$$\lambda_{ik} \approx \sum_{j \in G_k} P(\delta_{ij} = 1 | \pi_i, \pi_j) = \sum_{j \in G_k} \pi_i^T B \pi_j.$$

Since we do not observe δ_{ij} for $j \in G_k$, we will not be able to estimate the latent parameters $\{\pi_j\}_{j \in G_k}$ and thus not be able to infer the Poisson rate λ_{ik}. Instead, we approximate the rate by taking the expectation over the latent positions of nodes in subpopulation G_k. Defining N_k to be the number of nodes in subpopulation k, we have the approximation,

$$\frac{1}{N_k} \sum_{j \in G_k} \pi_i^T B \pi_j \approx \mathrm{E}_{\pi_j \sim P_k}(\pi_i^T B \pi_j)$$

where we introduced a distribution P_k over nodes j in subpopulation k. This P_k is a distribution over the K communities in the network. This gives

$$\lambda_{ik} \approx N_k \mathrm{E}_{\pi_j \sim P_k}(\pi_i^T B \pi_j). \tag{3.1}$$

The approximation in Equation (3.1) has two key features. First, the probability of a connection is no longer conditional on the membership of the two nodes but now conditions on the sender's membership and the expected membership of a node in the subpopulation. Second, it introduces a distribution over the set of latent membership vectors, P_k, with integration over the $K-1$ dimensional unit simplex.

Now we must make a choice for the subpopulation distribution. If we take $P_k = \text{Dirichlet}(\alpha_k)$, then

$$\text{E}_{\pi_j \sim P_k}(\pi_i^T B \pi_j) = \pi_i^T B \text{E}_{\pi_j \sim P_k}(\pi_j) = \pi_i^T B \eta_k$$

where $\eta_k = (\eta_k^1, \ldots, \eta_k^K)$ is a vector of length K and $\eta_k^j = \left[\frac{\alpha_k^j}{\sum_{j=1}^K \alpha_k^j}\right]$. Combining these results, we have

$$y_{ik} | \pi_i, \eta_k \sim \text{Poisson}(N_k \pi_i^T B \eta_k),$$

which is the likelihood of the MMSB for ARD. In this paper, we will infer η_k, the subpopulation mean, rather than α_k, which would additionally allow estimation of the subpopulations' concentration. To finalize our model specification we add prior distributions for the community memberships π_i and η_k as well as the blockmatrix B. The data generating processes for MMSB and the Aggregated Relational Data for MMSB is summarized in Figure 2.

3.2 Posterior Inference

To perform inference for large networks, we randomly sample minibatches of nodes and subpopulations and use the resulting ARD bipartite subgraphs to learn the parameters of the underlying MMSM model for the original network. The variational inference scheme we perform for each of these minibatches is described below.

As the true posterior of our model is not available in closed form, we develop an efficient variational algorithm for posterior approximation. We fit this procedure to each of the minibatches independently. This allows us to parallelize this algorithm across minibatches. These are minibatches of ARD data, namely samples of nodes and the corresponding subpopulation counts. This retains more expressive information than minibatches of the underlying adjacency matrix, $\boldsymbol{\delta}$.

Let Θ denote the set of unknown variables in the ARDMMSB, namely the community memberships, π, the subpopulation means, η, and the underlying blockmatrix, B. In variational inference, the true posterior is approximated by tractable distributions which are optimized to be close to the true posterior in terms of Kullback-Leibler divergence. Here we consider a fully-factorized family (commonly called mean field variational inference),

$$q(\Theta) = \prod_{i,k} q_K(\pi_i|\gamma_i) q_K(\eta_k|\phi_k) \prod_{m,n} \delta_{B_{mn}}$$

where we infer a separate variational family for each of the latent variables in our model. For both π_i and η_k a natural choice is to consider a

Dirichlet variational family, with parameters γ_i and ϕ_k respectively. We use q_K to denote these Dirichlet distributions of order K, while $\delta_{B_{mn}}$ is the point mass at B_{mn}. In variational inference we aim to infer the variational parameters such that the variational approximation best matches the true posterior distribution. We aim to optimise over the variational parameters, $\{\gamma, \phi, B\}$ to minimize the Kullback-Liebler divergence between the variational approximation, $q(\Theta)$, and the true posterior.

From Jensen's inequality, minimizing the Kullback-Leibler divergence between $q(\Theta)$ and the true posterior is equivalent to maximizing a lower bound $\mathcal{L}$ on the log marginal likelihood (Blei et al., 2017). The lower bound is given by

$$\begin{aligned}\mathcal{L} &= \mathrm{E}_q \log p(Y, \Theta) - \mathrm{E}_q \log q(\Theta) \\ &= \sum_{i,k} \mathrm{E}_q \log p(y_{ik}|\pi_i, \eta_k, B) + \sum_i \mathrm{E}_q \log p(\pi_i|\alpha) \\ &\quad + \sum_k \mathrm{E}_q \log p(\eta_k|\alpha) + \sum_{m,n} \mathrm{E}_q \log p(B_{mn}|a_{mn}, b_{mn}) + \mathcal{H}(q).\end{aligned} \tag{3.2}$$

Here note that $\mathrm{E}_q \log p(Y, \Theta)$ in (3.2) consists of both the priors for π, η and B, along with the likelihood of the MMSB for ARD. We consider a common Dirichlet prior with K dimensional parameter α for π and η. For entry B_{mn} of B we utilise a Beta(a_{mn}, b_{mn}) prior. In practice, we choose $a_{mn} = b_{mn} = 1$, giving a uniform prior on the entries of B, and a common concentration vector for the Dirichlet prior $\alpha = (\alpha, \ldots, \alpha)$.

However, $\mathrm{E}_q \log p(y_{ik}|\pi_i, \eta_k, B)$ cannot be evaluated in closed form. To circumvent this issue, we lower bound this term further by introducing

auxiliary parameters.

$$\begin{aligned}
&\mathrm{E}_q \log p(y_{ik}|\pi_i, \eta_k, B) \\
&\quad = y_{ik}\mathrm{E}_q \log(\pi_i^T B\eta_k) + y_{ik}\log N_k - N_k \mathrm{E}_q \pi_i^T B\eta_k \\
&\quad \geq y_{ik}\sum_{m,n} p_{ik}^{(mn)} \mathrm{E}_q \log(\pi_i^m B\eta_k^n) - \sum_{m,n} p_{ik}^{(mn)} \log p_{ik}^{(mn)} \\
&\qquad + y_{ik}\log N_k - N_k \mathrm{E}_q \pi_i^T B\eta_k. \qquad (3.3)
\end{aligned}$$

where $\{p_{ik}^{(mn)}|m, n = 1, \ldots, K\}$ is an auxiliary probability vector for every (i, k) pair. The lower bound on the log marginal likelihood obtained by using Equation (3.3) is denoted by $\mathcal{L}^*$. The full expression for $\mathcal{L}^*$ is included in the supplementary materials.

We optimize $\mathcal{L}^*$ via coordinate ascent, optimizing both the variational parameters $\{\gamma, \phi, B\}$ and the corresponding auxiliary parameters $p_{ik}^{(mn)}$. For the auxiliary parameters $p_{ik}^{(mn)}$, we update $\mathcal{L}^*$ by tightening inequality (3.3). For $\{\gamma, \phi\}$, the likelihood is nonconjugate with respect to the prior. We appeal to nonconjugate variational message passing for updates of these parameters (Knowles and Minka, 2011). This is a fixed point iteration method for optimizing the natural parameters of variational posteriors in exponential families. The advantages of this approach is that it yields closed form updates and extends to stochastic variational inference naturally. However, $\mathcal{L}^*$ is not guaranteed to increase at each step and updates for $\{\gamma, \phi\}$ may be negative at times. To resolve these issues, we use the fact that nonconjugate variational message passing is a natural gradient ascent method with step size 1 and smaller step sizes may also be taken. An outline

of the overall procedure is shown in Algorithm 1. When updating $\{\gamma, \phi\}$ using nonconjugate variational message passing, we start with step size 1 and reduce the step size where necessary to ensure updates are positive. If $\mathcal{L}^*$ increases, these updates are accepted. Otherwise, we revert to the former values. As a point mass $\delta_{B_{mn}}$ was chosen for the variational family for B, the entries are updated by a standard gradient step. We show the expression for the updates of $\{\gamma, \phi\}$ in Algorithm 1. The derivation of these updates and their exact form is included in the Supplementary Material.

Multiple Passes. When fitting to a large network, each minibatch will contain a small fraction of nodes. After initialization, the nodes in each minibatch will be run through the algorithm with weakly informative subpopulation blockmatrix parameters. The fit of each node ignores link information from all other minibatches. However, after being fit with Algorithm 1, the subpopulation and blockmatrix parameters contain richer information since they are averaged over all minibatches. Running Algorithm 1 a second time with the fits as initial values will allow the node parameters to be fit using information across the network. This process is summarized in Algorithm S1 of the supplementary material. We found that two passes is usually sufficient for the stability of parameters. This entire process is illustrated in Figure 3.

Per-iteration Complexity. At first glance, it seems that we must store a matrix of ancillary parameters $[p_{ik}^{(mn)}]_{m,n}$ for each node-subpopulation

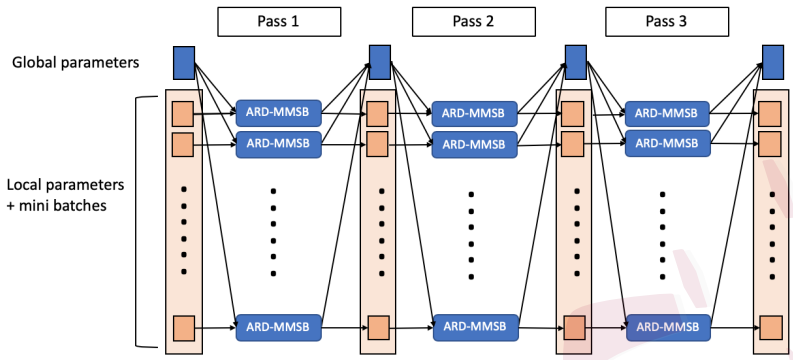


Figure 3: Illustration of inference process for multiple passes. Each tall orange rectangle represents all of the variational parameters for the nodes. The blue blocks represent the subpopulation and blockmatrix parameters while the orange blocks represent the parameters for each node. In each pass, the orange blocks are broken up into minibatches. Each of the minibatches are passed along with the current blue parameters and fit through the algorithm. After the pass, the orange blocks are stored and the blue parameters are averaged over before being stored.

pair. This would mean having a memory requirement of $O(D^2NK)$. This can be problematic when trying to fit the model with a large number of communities. However, all parameter updates depend on $p_{ik}^{(mn)}$ only through the term $y_{ik}p_{ik}^{(mn)}$. Thus, we can take advantage of the sparseness found in real-world networks by only storing ancillary parameters corresponding to nonzero counts y_{ik}.

Picking subpopulations. Ideally, subpopulations should be chosen so that the members of a subpopulation have similar community membership. In practice, one would need to use background information to pick subpopulations (McCormick and Zheng, 2013; McCormick et al., 2012, 2013), which may not be readily available. For instance, in a citation network, journals may be a good choice of subpopulations, as papers within a journal are often on the same topic. This will result in non-overlapping subpopulations. Alternatively, we could use specific keywords as a choice of subpopulations, which may result in overlapping subpopulations. For the simulation studies demonstrating the performance of our proposed method in Section 4 there is no background information and so the subpopulations were randomly chosen and still resulted in strong community recovery. Similarly, we note that while there is a computational cost to initially forming these subpopulations counts of $\mathcal{O}(n^2\kappa)$, this calculation is only required initially and does not need to be repeated.

Initialization From our synthetic data set, we found initialization of $\{\gamma, \phi\}$ is important for good recovery of the model parameters. We initialized by first forming a new matrix $\tilde{Y} = [y_{ik}/N_k]_{ik}$. That is $\tilde{Y}$ is a normalized version of the ARD. Communities for the sampled nodes were initialized using soft clustering on the top K left singular values of $\tilde{Y}$, while the subpopulation communities were clustered using the top K right singular values. We use this initialization scheme for all comparison methods considered in the simulation studies and real data example below.

4. Simulation Studies

In this section we perform simulation studies showing that bipartite graphs formed from ARD preserve enough information from the full network so that inference made using ARD minibatches carries over to inference based on the complete graph. Moreover, we compare our method to Gopalan and Blei (2013) and observe that we have excellent parameter recovery and model fit using ARD subgraphs, without observing all nodes in the network. The procedure of Gopalan and Blei (2013) is unable to correctly recover the true communities or model parameters using only a subgraph of the original network, only achieving comparable performance when the complete network is used.

4.1 Subsampling

In practice, a network is often presented to the researcher that is too large to conduct an analysis with the complexity required to gain an insight into the underlying network. Fitting Bayesian hierarchical models to such data simply take too much time. Because of the intractability of such models, the researcher oftentimes runs her analysis on a subgraph of the network. Choices of subgraphs include the largest connected component or a subset of nodes with similar covariates. However, inferences on the subgraph may be misleading when extended to the entire network. Ideally, the subgraph will in some way be representative of the original graph. Among the many sampling strategies of networks, simple random node selection has been shown to

create subgraphs that maintain many features of the graph (Leskovec and Faloutsos, 2006).

If the complete graph is a MMSB, subsampling nodes and doing inference on the resulting subgraph may lead to misleading inferences. Due to its data generating process, sampling nodes at random and keeping the edges between them will result in a MMSB with the same blockmatrix. However, such a procedure may lead to a large loss in efficiency and thus unstable parameter estimates. This is particularly an issue in sparse networks since such a subsampling procedure will leave out most of the links in the network, making it hard to learn the many parameters of the model. Using ARD leads to vastly improved estimates of both the blockmatrix and membership profiles.

We first wish to investigate the choice of subsample size on inference for such network data. Can valid inferences about the overall network structure be obtained using ARD subgraphs, rather than the whole network? Is this possible with existing methods for MMSB data, taking subgraphs of the underlying adjacency matrix instead? Figure 4 shows the results from a simulation experiment illustrating the instability of estimates from a subgraph formed by sampling nodes. We evaluate performance by community assignment and blockmatrix recovery. In this experiment, we simulated 10 networks, each a MMSB with $N = 10000$ nodes and $K = 6$ communities. Our model assumes that we have information other than the link structure that can define subpopulations or groups of similar nodes. Thus, for our

simulations, we must introduce a method of generating these subpopulations. We generated $\kappa = 50$ subpopulation centers and then generated the members or each subpopulation from a Dirichlet at the subpopulation center. The blockmatrix B is diagonally dominated to ensure exact recovery is possible with enough nodes (Zhao et al., 2012), with non-diagonal entries being 0. Three communities have within-linking probabilities of 0.1 while the other three have within- linking probabilities of 0.04. Given these simulated networks, we form subgraphs of the original network first, sampling $n = 500$ and $n = 5000$ nodes uniformly at random and keeping the edges between them. Similarly, ARD subgraphs were formed with $n = 500$ and $n = 1000$ sampled nodes. Throughout, the use of ARD corresponds to fitting our ARDMMSB model to subgraphs of ARD data using the inference scheme described in Section 3.2. We wish to examine the performance of these different network subgraphs (both original network subgraphs and ARD subgraphs) in terms of community and parameter recovery. To investigate how these samples compare to using the complete network, we also include the results from fitting the entire network using the stochastic variational inference scheme of Gopalan and Blei (2013). We adapt the stochastic variational inference algorithm presented in Gopalan and Blei (2013) to fit the subgraphs formed from random node sampling and using the entire network, (with SVI indicating the use of this inference scheme throughout this paper). Initialization of the nodes' variational parameters was performed using ten random restarts of spectral clustering. We report the best fit for

each of the random restarts.

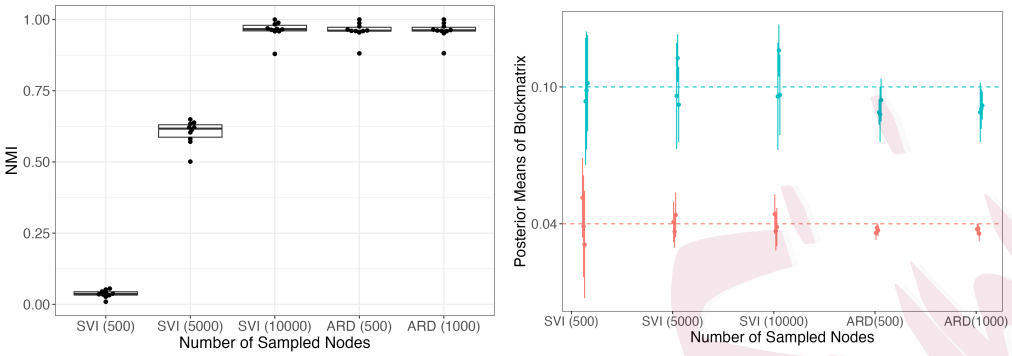


Figure 4: Left: Boxplots of normalized mutual information (NMI) among the subgraphs considered in simulations, using SVI for subgraphs of size n of the underlying adjacency matrix and ARD subgraphs of size n. Right: Posterior means and standard errors of estimation of diagonals of blockmatrix for SVI on subgraphs of the adjacency matrix and ARD data of size n. The true values of the block matrix are given as dashed horizontal lines.

The left panel of Figure 4 is a boxplot demonstrating performance of community recovery for the different subgraphs. We chose normalized mutual information between the fitted network's membership profiles and the true membership profiles as the measure of performance (Danon et al., 2005). For each node, we take its community assignment to be the maximum community in its membership profile. We show the result for $n = 500, 5000$ for subgraphs of the original network, using SVI and ARD subgraphs of size $n = 500, 1000$.

The bipartite graphs formed using ARD have excellent recovery, compa-

rable to using the entire network (SVI(10000)). Using SVI with subgraphs rather than the full graph struggles to recover the communities well. The right panel of Figure 4 plots the average posterior mean of the blockmatrix B's diagonal elements. We show these for ARD (ARD (500), ARD (1000)) and node subgraphs (SVI (500), SVI (5000)), along with using the entire network (SVI (10000)). As expected, we see that for the subgraphs formed from random sampling, the estimates have little bias and the standard error decrease as we increase the size of the subgraph, although it is still large with even large subgraphs. ARD incurs similar bias but has drastically smaller standard errors. Thus, ARD contains much more information about the original graph as it performs just as well as fitting on the entire network. The small bias of the ARD estimates here is related to the validity of the Poisson approximation, which may not be reasonable for the large subpopulations.

4.2 Comparison to Related Method

Expanding on the results showing the strong performance of our ARD procedure with small subsamples, we further compare our model and inference algorithm to the SVI algorithm in Gopalan and Blei (2013). Their method implements a stochastic variational inference algorithm for the MMSB. We utilise the same simulation setting discussed in section 4.1, generating networks of 10000 nodes. For a fixed subgraph size (500 nodes, 5% of the total network), we wish to compare the performance of these two procedures. We do this in terms of model convergence and community membership recovery.

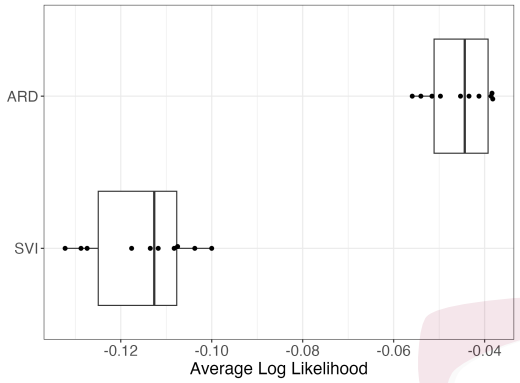


Figure 5: Comparison between ARDMMSB (ARD) and Gopolan *et. al.*'s Stochastic Variational Inference (SVI), showing the average predictive log likelihood with subgraphs of size $n = 500$.

The results are shown in Figure 5 and Figure 4.

While the previous results showed that SVI required the complete network to achieve similar community and parameter recovery to ARD subgraphs, we also wish to quantify how well our algorithm's solution fits the data compared to that of stochastic variational inference. Ideally, this would mean comparing the ELBOs, the target criterion of variational inference algorithm. Comparisons of the respective ELBOs is not possible since Gopalan and Blei (2013) implement a stochastic variational inference and therefore do not store any local parameters. Without the local parameters, the ELBO cannot be computed. Moreover, even if we could calculate the respective ELBOs, they would not be comparable since the models are different.

Since the ELBO cannot be computed when implementing stochastic variational inference, Gopalan and Blei (2013) evaluate model fitness through the predictive distribution (Geisser, 1975). Intuitively, a better model will have a higher predictive likelihood on a held-out set. The held-out predictive likelihood is thus used as a proxy for the ELBO. This is computed as follows:

$$
\begin{aligned}
&p(y_{ab}^{\text{new}} = 1 | y^{\text{obs}}) \\
&\quad = \int \theta_a^T B \theta_b p(\theta_a, \theta_b, B | y^{\text{obs}}) d\theta_a d\theta_b dB \\
&\quad = \int \theta_a^T B \theta_b p(\theta_a | \theta_b, B, y^{\text{obs}}) p(\theta_b, B | y^{\text{obs}}) p(B | y^{\text{obs}}) d\theta_a d\theta_b dB \\
&\quad \approx \int \theta_a^T B \theta_b q(\theta_a) q(\theta_b) q(B) d\theta_a d\theta_b dB = E_q \theta_a^T E_q B E_q \theta_b.
\end{aligned}
$$

We could also compute a predictive likelihood for our model, but they still would not be comparable. However, our algorithm yields a variational posterior for the node memberships and the blockmatrix. We can use these parameter estimates and plug them into the posterior likelihood defined above. Ideally, both of our models should be fit using the same held-out set of node pairs; but it is not obvious how the ARDMMSB model can handle such held-out node-pairs. Instead, we propose computing the predictive likelihood over the observed data. A good model fit should have a high predictive likelihood on the training set. We show the predictive likelihood over the observed data, using ARD subgraphs of size 500 and SVI with subgraphs of size 500 in Figure 5. Figure 5 shows that ARD achieves a higher model fit than SVI using a subgraph of the original network with the same number of nodes.

Due to space constraints, we defer further detailed comparisons to the supplementary material. There, we investigate our proposed method as we vary the number of communities, subpopulations, and sparsity of the network. We also examine the computational performance relative to SVI.

5. Applications

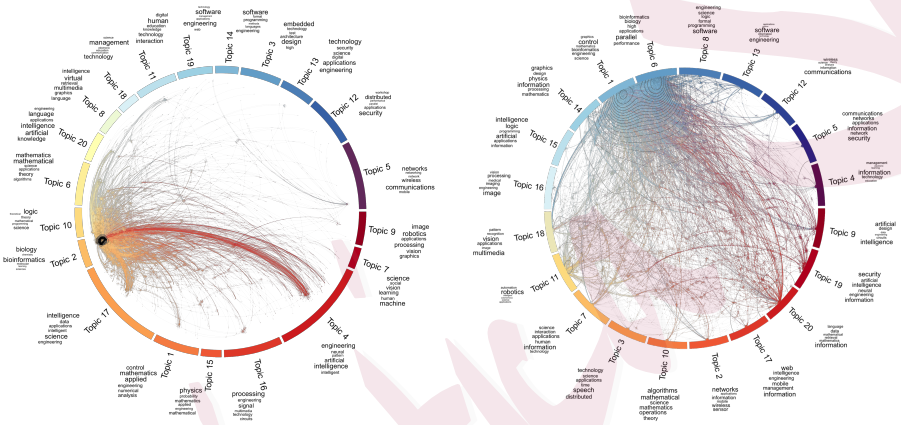


Figure 6: Left: Topic terrain plot of Bioinformatics using results from ARDMMSB. Right: Topic terrain plot of Bioinformatics using results from Gopalan and Blei (2013)'s stochastic variational inference. A word cloud of the top seven words is displayed with each topic. The font size is proportional to the term frequency.

We fit a MMSB using our algorithm to a real-world network to demonstrate how it can help study massive networks. We analyzed a citation network with over three million nodes and 25 million edges extracted from

DBLP, ACM, MAG, and other sources (Tang et al., 2008). We removed journals and papers that are isolated or do not have any outlinks. This reduced network has 2,139,891 papers and 4,349 journals.

For this network, we set the number of communities to twenty and used 1,000 mini-batches. The number of mini-batches used was chosen simply for convenience as the cluster used for fitting took a maximum of 1,000 jobs at once. In each minibatch, we used all the journals and a random subset of roughly 2,000 papers. We measure convergence according to our lower bound approximation of the ELBO. We stopped each mini-batch computation when the change in the ELBO was less than 10^{-2}.

To initialize the community membership profiles, we formed the journal to journal adjacency matrix and performed regularized spectral clustering. The result gave us hard community assignments for each journal. Each paper's membership was then initialized according to its journal's hard assignment. With this initialization our ARDMMSB model was used to obtain mixed memberships for each of the papers and journals.

Figure 6 shows our community detection results from the fitted model. Each panel shows what we call a topic terrain plot. This plot visualizes the breath and coverage of journals in the citation network. Each plot contains a circle plot of a particular journal. The circle is outlined with twenty colored blocks that represent the twenty communities that papers belong to. The size of a block corresponds to the size of the community or the number of papers in that community determined by the dominant community in its

membership profile vector. The placement of the bars around the circle is the result of hierarchical clustering of the twenty topics using inverse value of the probability of a link as the distance metric. Each topic also has a word cloud of the seven most frequent words found among the journal titles after removing stop words.

Inside the circle, we plot the papers' membership profile vectors within that journal as well as the coordinates of papers cited by papers within the journal. We randomly sampled 1000 papers from the journal to construct the plot. For each paper, we sample approximately fifteen of its cited papers on average. The bigger nodes within the wheel represent the papers from the journal and the smaller nodes represent the cited papers. Each link represents a citation activity. The coordinate of each node is the average of the coordinates of the centers of the twenty topic bars, weighted by the membership value of the node. The color of each node is also the average of the RGB value of twenty topic bars, weighted the same way. The color of each edge depends on the target node. The contour plot represents the density estimate of the papers from the journal.

From the word clouds, we can infer what communities correspond to which fields and topics. Those in yellow are related to mathematics. Those in orange correspond to biology and other biological applications, while those in light blue have topics relating to software and software design. Machine learning and artificial intelligence dominate the red region.

The top of Figure 6 shows the topic terrain plot for Bioinformatics

using results from our algorithm. Intuitively, we would expect papers in this journal to develop technologies for biological applications. Thus, they should cite papers in fields such as machine learning and computer vision as well as biology papers which is the citation behavior we see in the figure. We immediately notice that this journal has a strong footprint and unique identity. Most of the papers within Bioinformatics have a high membership in topic 2, which corresponds to biology and biological related fields. Many of papers in published in Bioinformatics cite papers with high membership in topic 4 which correspond to artificial intelligence and machine learning. There are also a significant number of papers being cited in fields such as mathematics and logic.

The bottom of Figure 6 shows the topic terrain plot for Bioinformatics using results from Gopalan and Blei (2013)'s stochastic variational algorithm. First notice that the word clouds are much more homogeneous, indicating their algorithm has difficulty detecting meaningful communities. Also, most of the papers belong exactly to one community; that is, papers do not exhibit mixed membership. This could be due to the fact that the topics themselves are not very distinct from one another or the algorithm has difficulty capturing interdisciplinary papers.

Figure 7 shows the topic terrain plots of the Journal of Machine Learning and Research (JMLR) from the two algorithms. Although JMLR is a journal in machine learning, we expect papers to be broadly spread out among many communities since machine learning can be organized into

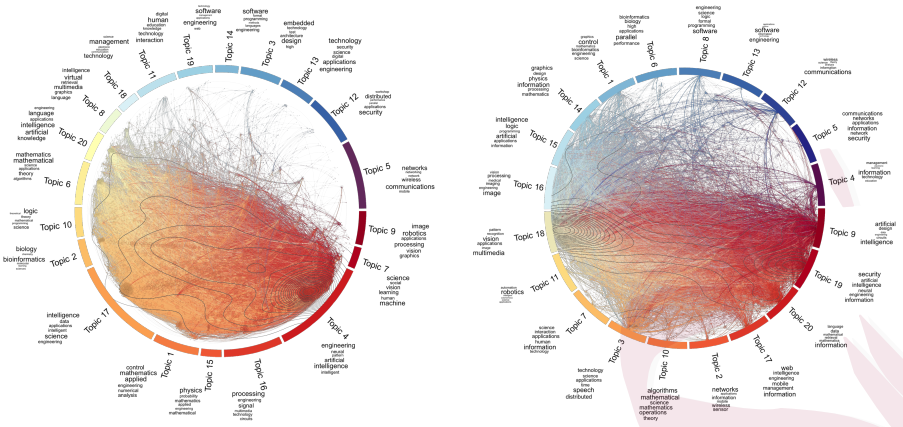


Figure 7: Left: Topic terrain plot of JMLR using results from ARDMMSB. Right: Topic terrain plot of JMLR using results from Gopalan and Blei (2013)'s stochastic variational inference. A word cloud of the top seven words is displayed with each topic. The font size is proportional to the term frequency.

smaller communities within its broad research landscape. It covers natural language processing, mathematics, logic, machine learning with biological applications, control theory, physics, vision and robotics. In the top plot of Figure 7, the half circle represents the foundations and applications of machine learning. Topic 4 has the highest concentration which contains journals based in artificial intelligence (AI) and neuroscience inspired AI, such as neural networks. Moreover, there is a large concentration in topic 17 which corresponds to AI without neuroscience. This agrees with our view of ML as a field. It confirms our method is recovering meaningful structure in

the citation network.

The bottom plot in Figure 7 continues the trend of having all the papers stuck on the edges, meaning that most papers belong to solely one topic. Thus, the algorithm cannot capture the interdisciplinary nature of many of the computer science papers that have applications in other areas. Also, the topics themselves do not have strong separation between fields. For instance, computer systems and ML Topics are mixed together and are dominated more by application areas rather than research areas.

By comparing the two plots, we illustrate how integrating nodal information can greatly improve results on real-world networks. For JMLR, our method divides the topic terrain plot into two half circles, one half being ML and the other about hardware, system design, security, etc. This is due to incorporating journal information which is essentially human-curated structure. Leveraging extra information allows ARDMMSB to uncover more structure in the citation network.

Figure 8 shows quantitative measures of comparison between our inference procedure and Gopalan and Blei (2013)'s stochastic variational inference algorithm. The left panel shows the ROC curves. The middle and right panels show boxplots of the area under the curve and relative ranks of both methods respectively. These plots were formed by subsampling 10,000 links and nonlinks and computing the probability of a link. Each of these subsamples forms an average rank measure and ROC curve which admits an area under the curve. The rank measure was formed by taking the average

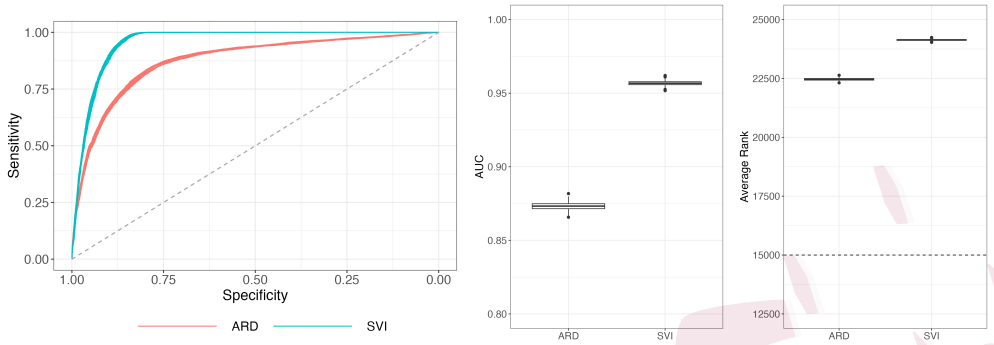


Figure 8: Comparison between ARDMMSB and Gopolan *et. al.*'s Stochastic Variational Inference. Left: ROC curves. Middle: Boxplot of the average area under the curve. Right: Boxplot of relative ranks.

rank of the links according to their predictive probabilities. We see that our method performs worse with regards to these metrics.

Although ARDMMSB does not do as well according to these measures, we only pick up communities formed by topics. ARD does not use paper to paper links. This can be beneficial as it removes some of the noise in link formation within the network. Since journals are human-curated by topics, aggregating links by journals ensure that ARD most likely uses link structure driven by topics. By only using paper to journal links, ARD removes some of the nuances due to other social structure such as community structure formed from coauthorships. Gopalan and Blei (2013)'s algorithm may capture this kind of structure, which is different from structure solely from topics. We have also created a Shiny app, available at `https://jyr123456.shinyapps.io/Topic_Terrain_Visualizer/`, to allow the exploration and interactive visualization of these results, allowing to user to

specify a selection of topics and journals.

6. Conclusion

The MMSB is popular model-based method of community detection of overlapping communities. However, this model is difficult to scale to networks on the order of millions of nodes since it parameterizes each pair of nodes. The algorithm from Gopalan and Blei (2013) implements a stochastic variational inference algorithm with a sampling scheme that takes advantage of the sparseness found in real-world networks. Although this algorithm works quite well, it ignores nodal information commonly found in networks that could reduce computational costs and improve model convergence.

In this paper, we introduced aggregated relational data for the MMSB (ARDMMSB) that incorporates such information with the present link structure. ARDMMSB lends itself to a mini-batch strategy that can be carried out in parallel which can drive down computation times. It works by aggregating sets of links to form a bipartite graph which is of much smaller dimension than the network's adjacency matrix. It retains information from the original graph and can be used to estimate the parameters of the MMSB directly. Our simulation studies and real-world application to a citation network illustrate how our algorithm can achieve improved results by leveraging information outside of the links themselves. There are several important extensions of this procedure which should be considered in future work. Jin et al. (2023) have extended the MMSB model to incorporate a

degree correction parameter, which leads to better model fitting with real networks. Incorporating such a degree correction here is an important future direction. We have also made all code used to create the figures in this paper publicly available in a github repo.

Supplementary Materials

The attached supplementary material include multiple additional simulation results which were omitted due to space constraints in the in main text. These serve as further demonstration of the performance of our proposed algorithm. We include additional details on our inference scheme, including omitted algorithms and details of the approximate lower bound which is considered. We also describe the impact of performing multiple passes of our algorithm on the fitted model for the citation network.

References

Airoldi, E. M., D. M. Blei, S. E. Fienberg, and E. P. Xing (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research 9*(Sep), 1981–2014.

Bickel, P. J. and A. Chen (2009). A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences 106*(50), pnas–0907096106.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877.

REFERENCES

Danon, L., A. Diaz-Guilera, J. Duch, and A. Arenas (2005). Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment 2005*(09), P09008.

DiPrete, T. A., A. Gelman, T. McCormick, J. Teitler, and T. Zheng (2011). Segregation in social networks based on acquaintanceship and trust. *American Journal of Sociology 116*(4), 1234–83.

Fortunato, S. and D. Hric (2016). Community detection in networks: A user guide. *Physics reports 659*, 1–44.

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American statistical Association 70*(350), 320–328.

Gopalan, P. K. and D. M. Blei (2013). Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences 110*(36), 14534–14539.

Hoffman, M. D., D. M. Blei, C. Wang, and J. Paisley (2013). Stochastic variational inference. *The Journal of Machine Learning Research 14*(1), 1303–1347.

Holland, P. W., K. B. Laskey, and S. Leinhardt (1983). Stochastic blockmodels: First steps. *Social networks 5*(2), 109–137.

Jin, J., Z. T. Ke, and S. Luo (2023). Mixed membership estimation for social networks. *Journal of Econometrics*, 105369.

Knowles, D. A. and T. Minka (2011). Non-conjugate variational message passing for multinomial and binary regression. In *Advances in Neural Information Processing Systems*, pp. 1701–

1709.

Leskovec, J. and C. Faloutsos (2006). Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 631–636. ACM.

Ma, Y. and T. Zheng (2017). Stabilized sparse online learning for sparse data. *The Journal of Machine Learning Research 18*(1), 4773–4808.

McCarty, C., P. D. Killworth, H. R. Bernard, E. C. Johnsen, and G. A. Shelley (2001). Comparing two methods for estimating network size. *Human organization 60*(1), 28–39.

McCormick, T. and T. Zheng (2013). Latent demographic profile estimation in hard-to-reach groups. *The Annals of Applied Statistics 6*(4), 1795–1813.

McCormick, T. H., R. He, E. Kolaczyk, and T. Zheng (2012). Surveying hard-to-reach groups through sampled respondents in a social network. *Statistics in Biosciences 4*(1), 177–195.

McCormick, T. H., A. Moussa, J. Ruf, T. A. DiPrete, A. Gelman, J. Teitler, and T. Zheng (2013). A practical guide to measuring social structure using indirectly observed network data. *Journal of Statistical Theory and Practice 7*(1), 120–132.

McCormick, T. H. and T. Zheng (2015). Latent surface models for networks using aggregated relational data. *Journal of the American Statistical Association 110*(512), 1684–1695.

Nowicki, K. and T. A. B. Snijders (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American statistical association 96*(455), 1077–1087.

REFERENCES

Shelley, G. A., H. R. Bernard, P. Killworth, E. Johnsen, and C. McCarty (1995). Who knows your hiv status? what hiv+ patients and their network members know about each other. *Social networks 17*(3-4), 189–217.

Soundarajan, S. and J. Hopcroft (2012). Using community information to improve the precision of link prediction methods. In *Proceedings of the 21st international conference on world wide web*, pp. 607–608.

Tan, L. S., A. H. Chan, T. Zheng, et al. (2016). Topic-adjusted visibility metric for scientific articles. *The Annals of Applied Statistics 10*(1), 1–31.

Tang, J., J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su (2008). Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998. ACM.

Ward, O. G., Z. Huang, A. Davison, and T. Zheng (2021). Next waves in veridical network embedding. *Statistical Analysis and Data Mining: The ASA Data Science Journal 14*(1), 5–17.

Zhao, Y., E. Levina, J. Zhu, et al. (2012). Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics 40*(4), 2266–2292.

Columbia University E-mail: (timothy.jones@columbia.edu), (tian.zheng@columbia.edu), (jwp2128@columbia.edu)

Simon Fraser University E-mail: (owen_ward@sfu.ca)

Purdue University E-mail: (jiang693@purdue.edu)

Algorithm 1 Variational inference procedure for ARD mini-batches

Initialize the variational parameters γ, ϕ, B. Cycle through the following steps until all nodes are sampled.

1. Randomly sample n nodes without replacement, obtaining sample $\mathcal{S}$.
2. Randomly sample k of the κ subpopulations without replacement.
3. Initialize local auxiliary probability variables p and step size $s_t = 1$. Cycle through steps 4 to 8 until convergence.
4. Update B using a gradient step, $\hat{B}_{mn} = \frac{\sum_{i,k} y_{ik} p_{ik}^{(mn)}}{\sum_{i,k} N_k \frac{\gamma_i^m}{\sum_d \gamma_i^d} \frac{\phi_k^n}{\sum_d \phi_k^d}}$ for a uniform prior for B_{mn}.
5. Update $\gamma_i \leftarrow (1 - s_t)\gamma_i + s_t \hat{\gamma}_i$ for $i \in \mathcal{S}$ where $\hat{\gamma}_i = I_{\gamma_i}^{-1} \nabla_{\gamma_i} \mathrm{E}_q[\log p(y, \Theta)]$. If any element of $\gamma_i \leq 0$, reduce s_t (say by half each time) until $\gamma_i > 0$. Accept update only if $\mathcal{L}^*$ increases.
6. Update p by tightening the lower bound $\mathcal{L}^*$, $p_{ik}^{(mn)} \propto \exp\left[\mathrm{E}_q \log(\pi_i^m B \eta_k^n)\right]$
7. Update $\phi_l \leftarrow (1 - s_t)\phi_k + s_t \hat{\phi}_l$ for $l \in 1, \ldots, K$ where $\hat{\phi}_l = I_{\phi_l}^{-1} \nabla_{\phi_l} \mathrm{E}_q[\log p(y, \Theta)]$. If any element of $\phi_l \leq 0$, reduce s_t (say by half each time) until $\phi_l > 0$. Accept update only if $\mathcal{L}^*$ increases.
8. Update p given these variational estimates, $p_{ik}^{(mn)} \propto \exp\left[\mathrm{E}_q \log(\pi_i^m B \eta_k^n)\right]$
9. Average the subpopulation parameters ϕ_l and blockmatrix B across mini-batches.