A Soft Affiliation Graph Model for Scalable Overlapping Community Detection

Nishma Laitonjam(✉), Wěipéng Huáng, and Neil J. Hurley

Insight Centre for Data Analytics, University College Dublin, Ireland
{nishma.laitonjam,weipeng.huang,neil.hurley}@insight-centre.org

Abstract. We propose an overlapping community model based on the Affiliation Graph Model (AGM), that exhibits the pluralistic homophily property that the probability of a link between nodes increases with increasing number of shared communities. We take inspiration from the Mixed Membership Stochastic Blockmodel (MMSB), in proposing an edgewise community affiliation. This allows decoupling of community affiliations between nodes, opening the way to scalable inference. We show that our model corresponds to an AGM with soft community affiliations and develop a scalable algorithm based on a Stochastic Gradient Riemannian Langevin Dynamics (SGRLD) sampler. Empirical results show that the model can scale to network sizes that are beyond the capabilities of MCMC samplers of the standard AGM. We achieve comparable performance in terms of accuracy and run-time efficiency to scalable MMSB samplers.

1 Introduction

Designing a scalable Markov Chain Monte Carlo (MCMC) inference for a Bayesian model is challenging due to the sequential nature of the mechanism, especially when the model parameters are huge in number and the dataset is large. Probabilistic graphical models define how the observed data is generated and often involve a large number of random variables. A case in point is the modelling of network data, where the datasets of interest nowadays scale to millions of nodes and a typical problem of interest is the extraction of community structure from the network. Many heuristic methods and probabilistic models have been proposed for this problem. In this paper, we focus on the extraction of overlapping community structure. Considering that any subset of the nodes could constitute such an overlapping community, we have *a-priori*, 2^N candidate communities, where N is the number of nodes in the network.

The Affiliation graph model (AGM) [22] is a probabilistic graphical model of overlapping community structures in networks. It proposes a likelihood that exhibits the *pluralistic homophily* property, meaning that the probability of a link between nodes increases with increasing number of shared communities. This property has been observed in the ground truth communities of real world data [22]. The heuristic algorithm proposed in [23] maximises the likelihood through a Non-negative Matrix Factorization (NMF) step and is a good benchmark for community-finding at scale. However, we are interested in developing MCMC algorithms that can sample from the true posterior distribution of the communities. A number of works have examined MCMC inference on models based on the AGM likelihood. For instance, using a Gamma process prior, [25] develop a non-parametric model which is sampled through Gibbs sampling and apply it to networks with the number of nodes and the number of edges below 10^4. The Infinite Multiple Membership Relational model (IMRM) [15] finds general overlapping block structure and reduces to the AGM likelihood when constrained to only within-block interactions. IMRM scales to networks of around 10^5 edges, on which it takes around 70 hours for $2,500$ iterations.

Another network model of block structure in networks is the Mixed Membership Stochastic Blockmodel (MMSB) [2] and its variant, the assortative-Mixed Membership Stochastic Blockmodel (a-MMSB), that models overlapping communities in the sense that nodes have mixed affiliations to multiple communities. However, the a-MMSB does not exhibit pluralistic homophily because the probability of an edge between two nodes does not increase with the total number of communities that they share. In contrast to the AGM, scalable inference techniques for the MMSB have been proposed in the state-of-the-art, for example, through the use of Stochastic Variational Inference (SVI) [9] and Stochastic Gradient-MCMC (SG-MCMC) [13], that achieve scalability by considering only a mini-batch of the dataset in each update step. Our contribution in this paper, is to propose a new variant of the AGM, which we call the *Soft* AGM (S-AGM), that is inspired by the a-MMSB but maintains the pluralistic homophily property of the AGM. Our model is amenable to the same inference strategies that have proven capable of scaling the MMSB to big network problems. In particular, in this paper, we will discuss how we have developed a SG-MCMC for the *Soft* AGM. Along with the advantage of using a mini-batch in each iteration, the SG-MCMC algorithm is highly parallelizable. We have developed it on `Tensorflow` and achieved tractable inference, beyond the capabilities of other MCMC samplers of the AGM, with networks of 10^5 edges converging within 2 hours on a 2.2 GHz Intel Core i7 processor.

The paper is structured as follows. In Section 2, we present the generative model of the S-AGM and show how, by collapsing the edge-wise community affiliation parameters, it may be interpreted as an AGM model with soft community affiliations. In Section 4, we discuss how to apply a Stochastic Gradient Riemannian Langevin Dynamics sampler to the model parameters, and derive the required gradients. In Section 5, we present some experimental results. Finally, we discuss the comparison of the resultant communities with ground truth communities and the merits of our model in comparison to the AGM.

2 Model

Consider an unweighted graph of $N > 0$ nodes, with adjacency matrix $\mathrm{A} = \{a_{ij}\}$. Let the training set node pairs, E, be partitioned into the non-link pairs,

$E_{\mathrm{NL}} = \{(i,j)|a_{ij} = 0\}$ and the link pairs $E_{\mathrm{L}} = \{(i,j)|a_{ij} = 1\}$, such that $E = E_{\mathrm{NL}} \cup E_{\mathrm{L}}$. We seek overlapping community structure with $K > 0$ communities. The Affiliation Graph Model (AGM) provides a generative model for networks with latent overlapping community structure, where the likelihood of the network is given by

$$p(\mathrm{A}|\Theta) = \prod_{i=1}^{N} \prod_{j>i}^{N} p_{ij}^{a_{ij}} (1 - p_{ij})^{1-a_{ij}} \tag{1}$$

with $\Theta = \{\mathrm{Z} = \{z_{ik}\}, \pi\}$ and $p_{ij} = 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k)^{z_{ik} z_{jk}}$, such that $z_{ik} = 1$ whenever node i is a member of community k, $p(z_{ik}|w_k) \sim \mathrm{Bernoulli}(w_k)$. The community edge density parameters are $\pi_k \sim \mathrm{Beta}(\eta_{k0}, \eta_{k1})$ and π_ϵ is a fixed background edge density. That the model exhibits *pluralistic homophily*, can most easily be observed by noting that, if all the community densities π were equal, then the probability that an edge (i,j) does not exist is proportional to $(1-\pi)^{\sum_k z_{ik} z_{jk}}$ i.e. $(1-\pi)^{s(i,j)}$, where $s(i,j) = \sum_k z_{ik} z_{jk}$ is the number of shared communities. One challenge for Bayesian inference from this model is that the conditional probabilities of the communities assignments $\mathrm{Z} = \{z_{ik}\}$ given the network are all inter-dependent and thus require sequential Gibbs sampler.

Motivated to develop a more scalable model that maintains pluralistic homophily, we take inspiration from the assortative Mixed Membership Stochastic Blockmodel (a-MMSB) and propose the *Soft* AGM (S-AGM) as follows: consider that, associated with each node i of the network and each community k, there is a soft community affiliation value, $w_{ik} \in [0,1]$. Now, for all possible interactions between nodes, i and j, each node draws a set of community membership assignments, $z_{ijk} \sim \mathrm{Bernoulli}(w_{ik})$ and $z_{jik} \sim \mathrm{Bernoulli}(w_{jk})$, and the interaction occurs with probability depending on the number of shared communities that are drawn:

$$p_{ij} = 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k)^{z_{ijk} z_{jik}} \,. \tag{2}$$

Note that in the S-AGM, each community affiliation is drawn independently from a Bernoulli distribution, so that multiple simultaneous affiliations are allowed and the existence of an edge depends on the overlap of the multiple affiliations between node pairs. In contrast, in the a-MMSB, for each interaction, a *single* community affiliation z_{ijk} is drawn from $\mathrm{Cat}(\mathbf{w}_i)$, where $\sum_k w_{ik} = 1$ and hence $\sum_k z_{ijk} = 1$. The existence of an edge is dependent on whether the single community drawn by node i coincides with that drawn by node j i.e. whether or not $z_{ijk} z_{jik} = 1$ is true. There is therefore no notion of multiple affiliations contributing to an interaction and hence the a-MMSB fails to model pluralistic affiliation.

From a scalability point-of-view, drawing the set of community affiliations independently for each interaction, has the effect of de-coupling the $\mathrm{Z} = \{z_{ijk}\}$, so that their conditional probabilities given the network (given in Section 3), can be updated in parallel.

The generative process model of S-AGM is given in algorithm 1. Note that a separate parameter α_k is drawn for each community, modelling that each community may have a different node density.

Algorithm 1 Generative process model

```
for $k = 1 : K$ do
    $\pi_k \sim \mathrm{Beta}(\eta_{k0}, \eta_{k1})$
for $k = 1 : K$ do
    $\alpha_k \sim \mathrm{Gamma}(\beta_0, \beta_1)$
    for $i = 1 : N$ do
        $w_{ik} \sim \mathrm{Beta}(\alpha_k, 1)$
for $i = 1 : (N - 1)$ do
    for $j = (i + 1) : N$ do
        for $k = 1 : K$ do
            $z_{ijk} \sim \mathrm{Bernoulli}(w_{ik})$
            $z_{jik} \sim \mathrm{Bernoulli}(w_{jk})$
        $p_{ij} = 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k)^{z_{ijk} z_{jik}}$
        $a_{ij} \sim \mathrm{Bernoulli}(p_{ij})$
```

In fact it is possible to marginalise $p(\mathrm{A}, \mathrm{Z}, \mathrm{W}, \alpha, \pi | \eta, \beta)$, with respect to Z. In Supplementary Material, we show the following lemma,

Lemma 1. *Collapsing* Z*:* $P(\mathrm{A}|\mathrm{W}, \pi) = \sum_{\mathrm{Z}} P(\mathrm{A}, \mathrm{Z}|\mathrm{W}, \pi)$ *is given by Equation* (1) *with* $\Theta = \{\mathrm{W} = \{w_{ik}\}, \pi\}$ *and* $p_{ij} = \rho_{ij}(\mathrm{W}, \pi) \triangleq 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k w_{ik} w_{jk})$.

In this form, we explicitly observe that the S-AGM corresponds to the AGM when w_{ik} are restricted to $\{0, 1\}$. The model may also be compared with the Gamma Process Edge Partition Model (GP-EPM), proposed in [25], in which w_{ik} are drawn from a Gamma distribution and $p_{ij} = 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k)^{w_{ik} w_{jk}}$. Note that aside from the difference in the form of the edge-connection probability, in the S-AGM, the w_{ik} are restricted to the probability simplex $[0, 1]$, while any positive affiliation weight is allowed in the GP-EPM.

3 MCMC on the non-collapsed Model

We firstly consider a simple inference strategy on the non-collapsed model and compare the results obtained from S-AGM with those obtained from AGM and a-MMSB. It may be verified that the posterior distribution of α is a Gamma distribution. A Gibbs sampling of the components of α can be carried out independently in parallel. In particular,

$$\alpha_k | w_{\cdot k} \sim \mathrm{Gamma}\left(N + \beta_0, \beta_1 - \sum_i \log(w_{ik}) \right) . \tag{3}$$

Similarly, we use Gibbs sampling of W with

$$w_{ik}|\alpha_k, Z \sim \mathrm{Beta}(\alpha_k + \sum_{j \neq i} z_{ijk}, 1 + \sum_{j \neq i}(1 - z_{ijk}))\,.$$

The community assignment for each training pair (i, j), i.e. $z_{ij.}$ and $z_{ji.}$ can be sampled in parallel. In particular for each community k, Gibbs sampling is used with

$$\begin{aligned} z_{ijk}, z_{jik}|\mathrm{Z} \setminus \{z_{ijk}, z_{jik}\}, \mathrm{A}, \mathrm{W}, \pi \\ \propto w_{ik}^{z_{ijk}} w_{jk}^{z_{jik}} (1 - w_{ik})^{1 - z_{ijk}} (1 - w_{jk})^{1 - z_{jik}} p_{ij}^{a_{ij}} (1 - p_{ij})^{1 - a_{ij}}\,, \end{aligned}$$

where p_{ij}, is given by Equation (2). As the posterior distribution of π is not in the form of a standard distribution, we use Hamiltonian Monte Carlo (HMC) MCMC to sample from π.

4 Scalable MCMC for the model

The soft community assignments, W, are the output of most interest from the model. We consider ways to obtain scalable inference with Z collapsed i.e. we seek the posterior distribution, $p(\mathrm{W}, \pi, \alpha|\mathrm{A}, \eta, \beta)$. The MCMC algorithm iterates updating local parameters (W) and global parameters (π and α). In the case of W and π, we consider sampling strategies that can efficiently explore the sample space.

The Metropolis Adjusted Langevin Algorithm (MALA) [19] is a Metropolis Hastings algorithm with a proposal distribution $q(\theta^*|\theta)$ of the form

$$\theta^* = \theta + \frac{\epsilon}{2}\left(\nabla_\theta \log p(\theta) + \sum_{i=1}^{N} \nabla_\theta p(x_i|\theta)\right) + \xi$$

where ϵ is a fixed step size and $\xi \sim N(0, \epsilon I)$. In [8], it is suggested that MALA can be improved for ill-conditioned problems by introducing an appropriate Riemann manifold pre-conditioner $\mathrm{G}(\theta)$, so that the proposal distribution becomes

$$\theta^* = \theta + \frac{\epsilon}{2}\mu(\theta) + G^{-1/2}\xi\,,$$

where, for an M-dimensional θ, the j^{th} component of $\mu(\theta)$ is given by,

$$\mu(\theta)_j = \left(G^{-1}\nabla_\theta \log p(\theta|X)\right)_j - 2\sum_{k=1}^{M}\left(G^{-1}\frac{dG}{d\theta_k}G^{-1}\right)_{jk} + \sum_{k=1}^{M} G_{jk}^{-1}\,\mathrm{Tr}\left(G^{-1}\frac{dG}{d\theta_k}\right)\,.$$

In [21], the expensive Metropolis correction step is not adopted. Instead, a mini-batch of the dataset D_t is sampled from X for each iteration and an unbiased but noisy estimate of the gradient is used: $\sum_{i=1}^{N} \nabla_\theta p(x_i|\theta) \approx \frac{N}{|D_t|}\sum_{x_i \in D_t} \nabla_\theta p(x_i|\theta)$ with a variable step-size ϵ_t. Convergence to the true posterior is guaranteed as long as decaying step sizes satisfy $\sum_{t=1}^{\infty} \epsilon_t = \infty$ and $\sum_{t=1}^{\infty} \epsilon_t^2 < \infty$. When applied with a Riemann manifold pre-conditioner, this method is referred to as Stochastic Gradient Riemannian Langevin Dynamics (SGRLD).

We follow [18] to develop an SGRLD algorithm for sampling π and W for the S-AGM. In particular, as these parameters are restricted to $[0, 1]$, it is necessary to re-parameterize so that the update step yields valid proposals in the parameter range. We adopt the expanded mean re-parameterization with mirroring strategy for Dirichlet parameters which is recommended in [18]. In this case, the preconditioner is chosen as $\mathrm{G}^{-1} = \mathrm{diag}(\theta)$, and the last two terms of $\mu(\theta)_j$ evaluate to 2 and -1 respectively.

4.1 Sampling π and W

We re-parameterize $\pi_k = \frac{\pi'_{k0}}{\pi'_{k0}+\pi'_{k1}}$, where for $m \in \{0, 1\}$, $\pi'_{km} \sim \mathrm{Gamma}(\eta_{km}, 1)$. The SGRLD update equations for π', taking absolute value to maintain the proposal in the range $\pi'^{*}_{km} > 0$, becomes

$$\pi'^{*}_{km} = \left| \pi'_{km} + \frac{\epsilon_t}{2} \mu(\pi'_{km}) + (\pi'_{km})^{1/2} \xi_{km} \right| , \tag{4}$$

with $\xi_{km} \sim N(0, \epsilon_t)$. Then, for a mini-batch of node pairs $\mathcal{E}^t$, we obtain

$$\mu(\pi'_{km}) = \eta_{km} - \pi'_{km} + s(\mathcal{E}^t) \sum_{(i,j)\in\mathcal{E}^t} g^a_{ij}(\pi'_{km}) , \tag{5}$$

where $g^a_{ij}(\pi'_{km}) \triangleq \frac{\partial}{\partial \pi'_{km}} \log p(a_{ij}|\pi', w_{i.}, w_{j.})$ and $s(.)$, discussed below, appropriately scales the mini-batch gradient estimate.

For each node i, we re-parameterize $w_{ik} = \frac{w'_{ik0}}{w'_{ik0}+w'_{ik1}}$ where for $m \in \{0, 1\}$, $w'_{ikm} \sim \mathrm{Gamma}(\gamma_{km}, 1)$, $\gamma_{k0} = \alpha_k$ and $\gamma_{k1} = 1$. We perform an SGRLD update for w'_{ik} as follows:

$$w'^{*}_{ikm} = \left| w'_{ikm} + \frac{\epsilon}{2} \mu(w'_{ikm}) + (w'_{ikm})^{1/2} \xi_{ikm} \right| , \tag{6}$$

where $\xi_{ikm} \sim N(0, \epsilon_t)$ and for a mini-batch of nodes $\mathcal{V}^t_i$,

$$\mu(w'_{ikm}) = \gamma_{km} - w'_{ikm} + \frac{N}{|\mathcal{V}^t_i|} \sum_{j\in\mathcal{V}^t_i} g^a_{ij}(w'_{ikm}) , \tag{7}$$

where $g^a_{ij}(w'_{ikm}) \triangleq \frac{\partial}{\partial w'_{ikm}} \log p(a_{ij}|\pi, w'_{i.}, w'_{j.})$. Full expressions for $g^a_{ij}(\pi'_{km})$ and $g^a_{ij}(w'_{ikm})$ are given in the Supplementary Material.

4.2 Mini-batch Selection

We follow the stratified random node sampling strategy which is shown to give the best gains in convergence speed for variational inference on an MMSB model in [9]. All the node pairs incident with each node i are partitioned into u sets, $\mathcal{N}_{il} \subset E_{\mathrm{NL}}$, $l = 1, \ldots, u$ of non-link pairs and one set, $\mathcal{L}_i \subseteq E_{\mathrm{L}}$, of the link pairs. Note that each node pair occurs within these sets exactly $c = 2$ times. To select the mini-batch $\mathcal{E}^t$, firstly a node i is selected at random, and then with probability 1/2, either the link set is chosen or, otherwise, one of the non-link sets is chosen with probability $1/u$. Let $s(\mathcal{E}^t) = Nu$ if $\mathcal{E}^t = \mathcal{N}_{il}$ for some l and $s(\mathcal{E}^t) = N$ if $\mathcal{E}^t = \mathcal{L}_i$. In the Supplementary Material, we show that this choice of scaling results in an unbiased estimate of the true gradient. To update w'_{ikm}, for each node i in mini-batch $\mathcal{E}^t$ we sample a fixed number of nodes at random to form the mini-batch $\mathcal{V}_i^t$.

The pseudo-code for the full MCMC algorithm is given in Algorithm 2. All the for loops in Algorithm 2 are parallelizable.

Algorithm 2 MCMC for the S-AGM using SGRLD

```
Sample a mini-batch $\mathcal{E}^t$ of node pairs.
for Each node $i$ in $\mathcal{E}^t$ do
    Sample a mini-batch of nodes $\mathcal{V}_i^t$.
    for $k = 1 : K$ do                                  ▷ utilizing the sampled $\mathcal{V}_i^t$
        Update $w_{ik}$ according to Equations (6) and (7).
for $k = 1 : K$ do                                      ▷ utilizing the sampled $\mathcal{E}^t$
    Update $\pi_k$ according to Equations (4) and (5).
for $k = 1 : K$ do
    Update $\alpha_k$ according to Equation (3).
```

5 Experimental Results

We initially developed a proof-of-concept `Matlab` code[1] both for the uncollapsed S-AGM model and for the SG-MCMC algorithm. To take advantage of the parallelizability of the collapsed model, we then implemented the SG-MCMC algorithm using `Tensorflow` [1] and ran it on a GPU.

Throughout the experiments we have chosen $\eta_{k0} = 5$, $\eta_{k1} = 1$ as the hyperparameters for the community edge probability incorporating the prior information that a community consists of strongly connected nodes. For the hyperparameters of α_k, we have chosen $\beta_0 = \beta_1 = 1$. We have initialized the probability of a node belonging to a community for S-AGM and a-MMSB to be $1/K$ which also satisfies the condition that $\sum_k 1/K = 1$ for the membership vector of the a-MMSB. The edge probabilities for each community are initialized by drawing from the prior, $\mathrm{Beta}(\eta_{k0}, \eta_{k1})$ for all models.

To compare different community assignments we use the overlapping Normalised Mutual Information (NMI) [11]. To compare the convergence of the

[1] https://github.com/nishma-laitonjam/S-AGM

MCMC chain, we use area under the Receiver Operating Characteristics curve (AUC-ROC) to predict missing links of hold-out test set, T. We also use perplexity defined as the exponential of the negative average predictive log likelihood on the hold-out test set [9], i.e. $\mathrm{perp}(\mathrm{T}|\pi, \mathrm{W}) = \exp\left(-\frac{\sum_{(i,j)\in \mathrm{T}} \log p(a_{ij}|\pi, w_{i.}, w_{j.})}{|\mathrm{T}|}\right)$. For small datasets, the change in log likelihood of the training dataset is also used to check for convergence.

5.1 Networks generated by AGM

To observe whether S-AGM can recover the network structure of the AGM, we compare the two models applied to networks generated from the AGM. For this experiment, we run the SGRLD batch algorithm for S-AGM in `Matlab` and compare it to a `Matlab` implementation for AGM that uses Gibbs sampling along with HMC. We use these implementations to examine the run-time advantages of the batch SGRLD algorithm over Gibbs and HMC.

Specifically, networks with two communities are generated using the generative process of AGM, i.e set $K = 2$ and edges between nodes i and j are generated with probability $p_{ij} = 1 - (1 - \pi_\epsilon) \prod_k (1 - \pi_k)^{z_{ik} z_{jk}}$. A community assignment Z is chosen such that in each network, 20% of the nodes belong to the overlapping region of the two communities and 40% of the nodes belong to each community only. The network size is $n = 100$. For Fig. 1, we fix $\pi_k = 0.8\ \forall k$ and vary the background edge probability π_ϵ. For Fig. 2, we fix $\pi_\epsilon = 0.00005$ and vary π_k. When fitting the models, we fix $\pi_\epsilon = 0.00005$ in all cases, so that the first experiment tests the ability of the algorithm to recover the network with different levels of background noise.

The similarity of the resultant communities with the ground truth communities is reported as NMI. The step size of SGRLD is decreased using, $\epsilon_t = a\big(1 + \frac{t}{b}\big)^{-c}$ where a is the initial value, t is the iteration number, and $c \in (0.5, 1]$ is the learning rate. Following [18], we have chosen $b = 1,000$ and $c = 0.55$. For these networks, we find $a = 0.01$ performs well for sampling both π and $w_{i \cdot}$. Since S-AGM reports the community assignment of a node as a soft assignment, we use a threshold to convert to a hard assignment before computing NMI. After burn-in of 500 iterations, 500 samples are collected, and the average result of 5 random runs is reported. From Fig. 1, we can see that S-AGM with 0.5 as threshold is more tolerant to background noise than AGM. When there is no noise i.e. when $\pi_\epsilon = 0.00005$, both S-AGM and AGM are able to recover the ground truth network. As noise increases, S-AGM performs better to recover the ground truth communities as the noise is reflected in the inferred model only as a small positive probability of belonging to the other community. Thus S-AGM has higher NMI compared to AGM. From Fig. 2, when we change the within-community edge probability with fixed $\pi_\epsilon = 0.00005$, both S-AGM with 0.3 as threshold and AGM, gives similar NMI recovering the ground truth community well when the within-community edge probability is greater than or equal to 0.4.

To compare the runtime between AGM and S-AGM, we generate networks with $k = 2$, $\pi_k = 0.8\ \forall k$ and $\pi_\epsilon = 0.00005$ but of different sizes n ranging from

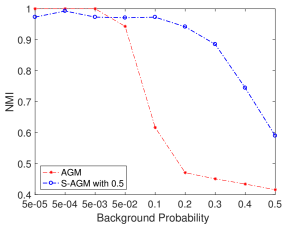

Fig. 1: NMI vs π_ϵ

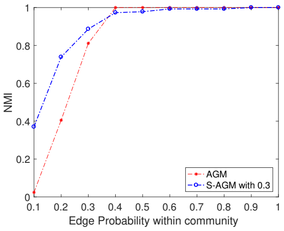

Fig. 2: NMI vs π_k

100 to 1,000 in a step-size of 100. After burn in of 500 iterations, 500 samples are collected, and the average AUC-ROC score of 5 random runs on Intel core i5, 4 cores is reported in Fig. 4.

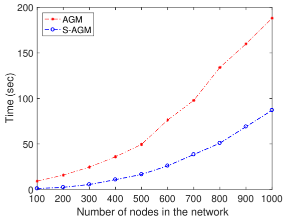

Fig. 3: Time vs n

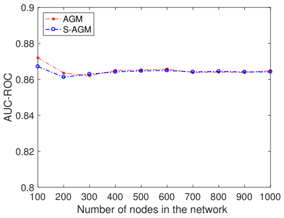

Fig. 4: AUC-ROC vs n

From Fig. 3, we can see that the batch SGRLD for S-AGM performs better than MCMC for AGM, while both give a similar AUC-ROC score. The scalability of the SGLRD for large n, using mini-batch and GPU, is explored in Section 5.3.

5.2 Comparing S-AGM with AGM and a-MMSB

In this section, first we show that both AGM and S-AGM exhibit pluralistic homophily. We then compare the performance of S-AGM with AGM and a-MMSB in terms of convergence of the log likelihood on the training dataset and predicting missing links on the hold-out dataset which is comprised of 20% of the node pairs in the dataset, chosen at random. For these experiments we use 3 small networks i.e. Football [17], NIPS234 [14] and Protein230 [3]. We use uncollapsed MCMC for all models with the `Matlab` code. We set the number of communities as $K = 5, 10, 15, 20$, and plot number of shared communities per node pair, i.e. $\sum_k z_{ijk} z_{jik}$ for S-AGM and $\sum_k z_{ik} z_{jk}$ for AGM, against edge probability. (As noted in Section 2, in the case of a-MMSB, as $\sum_k z_{ijk} z_{jik} \in \{0, 1\}$ always, there is no direct notion of pluralstic homophily in that model).

Fig. 5 shows a clear increase in edge probability with increasing number of shared communities in both the AGM and S-AGM models. These plots are obtained from a single run, for various K.

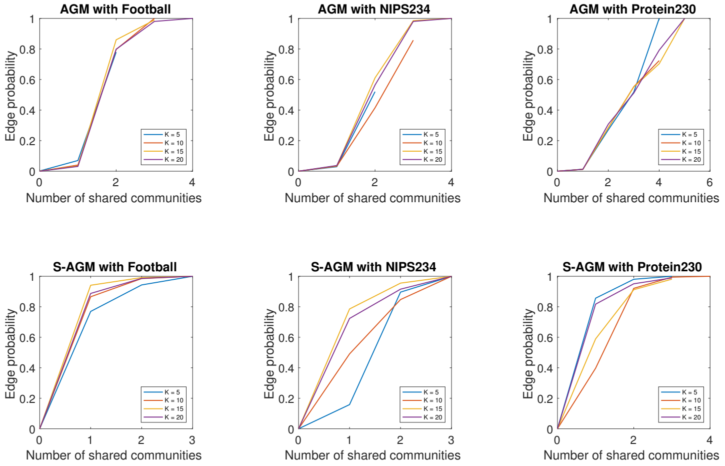

Fig. 5: Pluralistic Homophily by AGM and S-AGM

Table 1 shows the average AUC under ROC curve for predicting missing links, taken over 5 random runs where, in each run, 2,500 samples are collected after burn-in of 2,500 iterations. We can see that the AUC-ROC score is very similar for the 3 models with AGM performing best for NIPS234. It may be observed from Fig. 6, that the log likelihood for AGM is highest compared to the other two models. The perplexity is computed after every 100 iterations and the trace plots from a single run are shown in Fig. 7. Again, there is little difference between the three models, even though from Fig. 6 the convergence is slower for non-collapsed S-AGM due to the larger number of parameters to learn.

Table 1: AUC-ROC

Network		Football				NIPS234				Protein230			
K		5	10	15	20	5	10	15	20	5	10	15	20
AUC-ROC	AGM	0.7097 ±0.028	0.8055 ±0.043	0.8316 ±0.016	0.8240 ±0.015	0.8280 ±0.019	**0.9266** ±**0.009**	**0.9481** ±**0.008**	**0.9511** ±**0.008**	0.9088 ±0.017	0.9237 ±0.015	0.9236 ±0.015	**0.9290** ±**0.013**
	a-MMSB	**0.8242** ±**0.020**	**0.8637** ±**0.015**	**0.8587** ±**0.016**	**0.8615** ±**0.015**	**0.8855** ±**0.011**	0.9121 ±0.011	0.9274 ±0.021	0.9359 ±0.012	0.8867 ±0.022	0.8872 ±0.015	0.8875 ±0.019	0.8850 ±0.019
	S-AGM	0.7889 ±0.017	0.8426 ±0.021	0.8403 ±0.017	0.8430 ±0.017	0.8654 ±0.028	0.9039 ±0.016	0.9080 ±0.021	0.9039 ±0.0215	**0.9236** ±**0.014**	**0.9296** ±**0.009**	**0.9305** ±**0.010**	0.9287 ±0.010

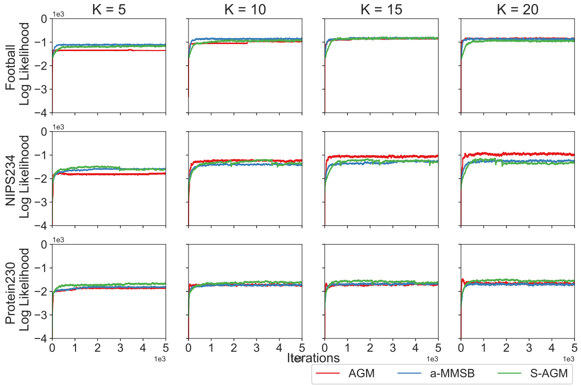

Fig. 6: Trace plot of log likelihood of training data

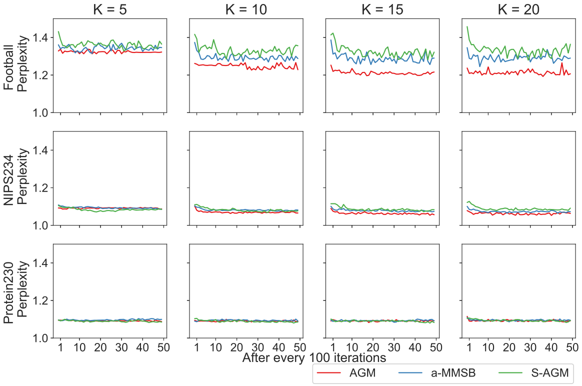

Fig. 7: Trace plot of perplexity of test data

Comparison with ground truth communities With the availability of ground truth communities for the Football network, we are able to compare the communities generated by S-AGM with these communities. The Football network contains the network of American football games between Division IA colleges during regular season, Fall 2000. There are 115 teams that are grouped into 11 conferences along with 8 independent teams that are not required to schedule each other for competition, like colleges within conferences must do [6]. We have used the Fruchterman-Reingold algorithm [20, 7] to plot the community structure found by S-AGM alongside the ground truth communities in Fig. 8. The 8 independent teams are the black nodes in the ground truth. For the S-AGM plot, the pie-chart at each node indicates its relative membership of each found community.

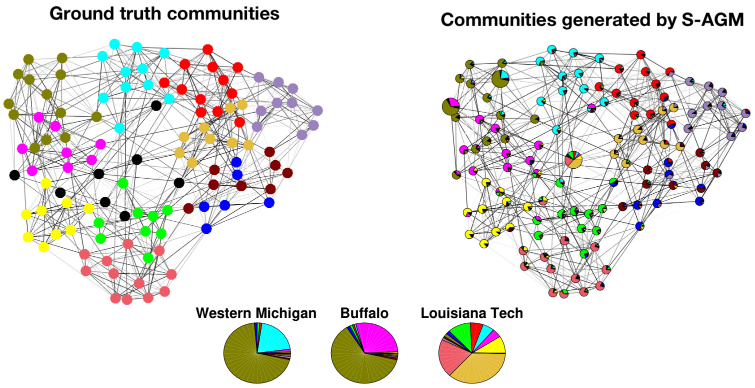

Fig. 8: Communities for Football network.

Conferences teams play more intra-conference games than inter-conference games, thus forming a clear community structure, while the 8 independent teams play with other teams as frequently as among themselves. The S-AGM recovers the 11 conferences well when $K = 15$. Three out of 15 found communities are empty. Games between teams from different conferences are not uniform. Rather, geographically close teams tend to play each other more often [10]. This pattern is captured in the overlapping structure identified by S-AGM, where each conference team belongs to a single dominant community, but has some small probability of belonging to another conference, proportional to its distance to teams within that conference.

In Fig. 8, we focus on Western Michigan and Buffalo, two Mid American conference teams, as well as Louisiana Tech, an independent team. Clearly, Louisiana Tech has no clear community assignment, rather, it can be considered as a part of multiple conferences. It plays more games with teams in the West Atlantic conference (dark yellow) and the Southeastern conference (maroon). While Western Michigan and Buffalo have very strong affiliation to their own conference, due to the geographical proximity, Western Michigan plays more with teams in the Big Ten conference (Iowa and Wisconsin) while Buffalo plays more games with teams in the Big East conference (Syracuse and Rutgers).

Such overlapping structure where a node belongs to multiple communities with a different degree of overlap cannot be captured by the AGM model. In AGM a node either belongs fully to the community or not. For the Football network, with $K = 15$, AGM generates one community that contains all nodes to capture the inter-community edges and other communities as the sub-communities to capture the intra-community edges corresponding to the ground truth communities. Thus the community structure generated by the AGM doesn't provide the information that even though a team belongs to a conference, the team also plays with other teams of different conferences with different frequencies.

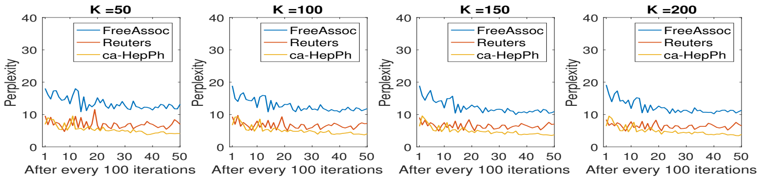

Fig. 9: Trace plot of perplexity of test data for various K

5.3 Larger Problems

For experiments on larger problems, we use the FreeAssoc network [16] (10,468 nodes and 61,677 edges), the Reuters network [4] (13,314 nodes and 148,038 edges) and the ca-HepPh network [12] (12,008 nodes and 118,489 edges) and run the mini-batch SGRLD algorithm for these networks. Taking advantage of the parallelizability of the algorithm, it is implemented on `Tensorflow` and run on a 2.2 GHz Intel Core i7 processor. To overcome the memory problem for larger networks, especially to run with GPUs, we store the network outside the limited GPU memory. Mini-batch samples are stored in the `tf.records Tensorflow` binary storage format. This speeds up the process of passing the mini-batch for each iteration to the GPUs. Thus, first the mini-batch of every 100 iterations is sampled and stored in a `tf.records` structure and one `tf.records` is read in every 100 iterations using an initializable iterator. For gradient computation, we implemented the analytical form directly, rather than using Tensorflow's gradient function. We take $K = 50$, $L = N/u = 1,000$ and $|\mathcal{V}^t| = 1,000$.

The step size of SGRLD is decreased similar to Section 5.1 and for these networks, we find $a = 0.001$ performs well for sampling both π and $w_{i\cdot}$ for these networks. To check the performance for these experiments, a test set consisting of 50% edges and 50% non-edges is chosen at random. The size of the test set is taken as 10% of the edges in the graph. The convergence of the perplexity for the test set is given in Fig. 9. Table 2 shows the runtime in hours for $5,000$ iterations and average AUC-ROC scores for $2,500$ samples collected after a burn-in of $2,500$ iterations. Along with Fig. 9, we can see that the performance of S-AGM does not decrease as K grows, which is also observed in SG-MCMC of a-MMSB [13].

Table 2: AUC-ROC scores of test data and runtime (hrs) for various K

	AUC-ROC				runtime (hrs)			
K	50	100	150	200	50	100	150	200
FreeAssoc	0.8989	0.9064	0.9041	0.9086	0.6434	1.0844	1.4563	1.8031
Reuters	0.9441	0.9455	0.9472	0.9472	0.6646	1.0725	1.5141	1.8709
ca-HepPh	0.9346	0.9480	0.9503	0.9470	0.6582	1.0815	1.4886	1.8637

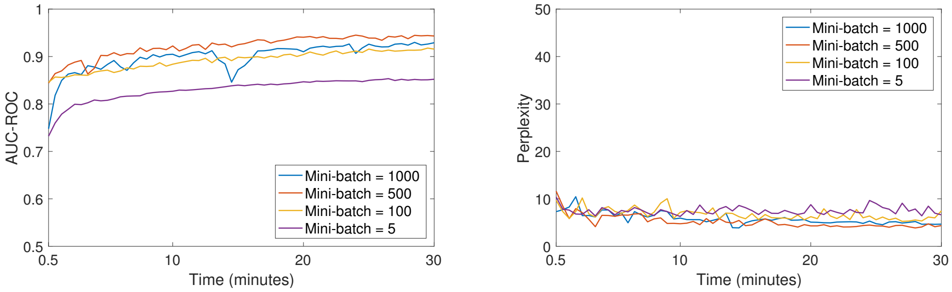

Fig. 10: Trace plot of AUC-ROC score and perplexity of test data for ca-HepPh

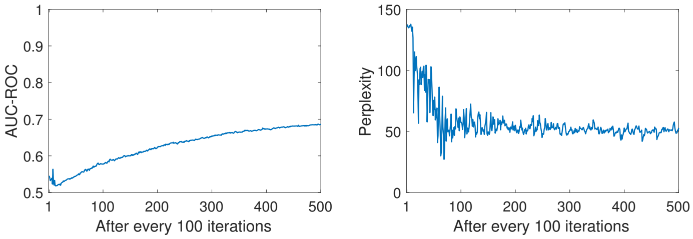

Fig. 11: Trace plot of AUC-ROC score and perplexity of test data for com-dblp

Effect of mini-batch size For this experiment, we vary the mini-batch size for the ca-HepPh network with $L = |\mathcal{V}^t| \in \{1000, 500, 100, 5\}$ respectively and study the effect of change in mini-batch size with $K = 50$. In SGRLD, the mini-batch size is a hyperparameter. The convergence speed greatly depends on the mini-batch size though the process with any mini-batch size will finally converge when the MCMC chain is run infinitely. With larger mini batch size, per iteration time is comparatively longer and hence the convergence runtime is also slow. Whereas with very small mini-batch size, only very few w will be updated per iteration and the process will achieve poor predictive performance for missing links due to the larger variance of the stochastic gradient. Shown in Fig. 10, the mini-batch size $L = |\mathcal{V}^t| = 500$ for ca-HepPh obtains the best predictive performance of missing links within 30 minutes. Although SGRLD with large mini-batch size is faster with no metropolis acceptance step, a better choice of mini-batch size with low variance in stochastic gradient also helps in speeding up the convergence.

Tensorflow on GPU To demonstrate the scalability of the inference algorithm, we run the `Tensorflow` code using the com-dblp network [24] which has more than 1 million edges. The experiment is carried out on a machine equipping with an AMD Ryzen 7 Eight-Core Processor at 2.2 GHz, Nvidia GTX TitanX with 12GB memory, and 64GB RAM. For this experiment, we consider $K = 2048$, $L = 4096$ and $|\mathcal{V}^t| = 32$. With the same initialization as the above experiments, except for a which is taken as $a = 0.0001$ here, the algorithm is run for $50,000$ iterations and takes 11.5 hours. The convergence of perplexity and AUC-ROC score on the test set is given in Fig. 11. From the experiment we can see that S-AGM achieves similar runtime scalability as a-MMSB when implemented with GPU [5].

6 Conclusion and Future Work

In this paper we have developed a new overlapping community model (Soft-AGM) that exhibits pluralistic homophily. Overlapping communities are modelled as soft node to community assignments, which, if constrained to be hard, would result in the Soft AGM likelihood reducing to the standard AGM likelihood. A highly parallelizable and scalable MCMC algorithm for inference based on a stochastic gradient sampler is developed for the model, allowing the inference to be carried out on networks that are well beyond the size of networks tackled by previous MCMC samplers applied to the AGM. In particular, a `Tensorflow` implementation has been used to run the model on a network with 10^6 edges. As future work, we would like to implement the algorithm on a HPC infrastructure to find community structure on very large networks, such as "Friendster", "LiveJournal" and so on. We will also consider to make the model non-parametric, allowing the number of non-empty communities to be learned.

Acknowledgments

This project has been funded by Science Foundation Ireland under Grant No. SFI/12/RC/2289.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org
2. Airoldi, E.M., Blei, D.M., Fienberg, S.E., Xing, E.P.: Mixed membership stochastic blockmodels. Journal of Machine Learning Research **9**(Sep), 1981–2014 (2008)
3. Butland, G., Peregrín-Alvarez, J.M., Li, J., Yang, W., Yang, X., Canadien, V., Starostine, A., Richards, D., Beattie, B., Krogan, N., et al.: Interaction network containing conserved and essential protein complexes in escherichia coli. Nature **433**(7025), 531 (2005)
4. Corman, S.R., Kuhn, T., McPhee, R.D., Dooley, K.J.: Studying complex discursive systems. centering resonance analysis of communication. Human communication research **28**(2), 157–206 (2002)
5. El-Helw, I., Hofman, R., Bal, H.E.: Towards fast overlapping community detection. In: 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid). pp. 175–178. IEEE (2016)
6. Evans, T.S.: Clique graphs and overlapping communities. Journal of Statistical Mechanics: Theory and Experiment **2010**(12), P12037 (2010)

7. Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. Software: Practice and experience **21**(11), 1129–1164 (1991)
8. Girolami, M., Calderhead, B.: Riemann manifold langevin and hamiltonian monte carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **73**(2), 123–214 (2011)
9. Gopalan, P.K., Gerrish, S., Freedman, M., Blei, D.M., Mimno, D.M.: Scalable inference of overlapping communities. In: Advances in Neural Information Processing Systems. pp. 2249–2257 (2012)
10. Gschwind, T., Irnich, S., Furini, F., et al.: Social network analysis and community detection by decomposing a graph into relaxed cliques. Tech. rep. (2015)
11. Lancichinetti, A., Fortunato, S., Kertesz, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics **11**(3), 033015 (2009)
12. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Transactions on Knowledge Discovery from Data (TKDD) **1**(1), 2 (2007)
13. Li, W., Ahn, S., Welling, M.: Scalable mcmc for mixed membership stochastic blockmodels. In: Artificial Intelligence and Statistics. pp. 723–731 (2016)
14. Miller, K., Jordan, M.I., Griffiths, T.L.: Nonparametric latent feature models for link prediction. In: Advances in neural information processing systems. pp. 1276–1284 (2009)
15. Mørup, M., Schmidt, M.N., Hansen, L.K.: Infinite multiple membership relational modeling for complex networks. In: Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on. pp. 1–6. IEEE (2011)
16. Nelson, D.L., McEvoy, C.L., Schreiber, T.A.: The university of south florida free association, rhyme, and word fragment norms. Behavior Research Methods, Instruments, & Computers **36**(3), 402–407 (2004)
17. Newman, M.E.: The structure and function of complex networks. SIAM review **45**(2), 167–256 (2003)
18. Patterson, S., Teh, Y.W.: Stochastic gradient riemannian langevin dynamics on the probability simplex. In: Advances in Neural Information Processing Systems. pp. 3102–3110 (2013)
19. Roberts, G.O., Rosenthal, J.S.: Optimal scaling of discrete approximations to langevin diffusions. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **60**(1), 255–268 (1998)
20. Traud, A.L., Frost, C., Mucha, P.J., Porter, M.A.: Visualization of communities in networks. Chaos: An Interdisciplinary Journal of Nonlinear Science **19**(4), 041104 (2009)
21. Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 681–688 (2011)
22. Yang, J., Leskovec, J.: Community-affiliation graph model for overlapping network community detection. In: Data Mining (ICDM), 2012 IEEE 12th International Conference on. pp. 1170–1175. IEEE (2012)
23. Yang, J., Leskovec, J.: Overlapping community detection at scale: a nonnegative matrix factorization approach. In: Proceedings of the sixth ACM international conference on Web search and data mining. pp. 587–596. ACM (2013)
24. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. Knowledge and Information Systems **42**(1), 181–213 (2015)

25. Zhou, M.: Infinite edge partition models for overlapping community detection and link prediction. In: Artificial Intelligence and Statistics(AISTATS). pp. 1135–1143 (2015)