Proceedings of Machine Learning Research 129:673–688, 2020 ACML 2020

Scalable Inference on the Soft Affiliation Graph Model for Overlapping Community Detection

Nishma Laitonjam NISHMA.LAITONJAM@INSIGHT-CENTRE.ORG
Wěipéng Huáng WEIPENG.HUANG@INSIGHT-CENTRE.ORG
Neil J. Hurley NEIL.HURLEY@INSIGHT-CENTRE.ORG
Insight Centre for Data Analytics, University College Dublin, Ireland

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

The Soft Affiliation Graph model (S-AGM) is a Bayesian generative model of overlapping community structure in social networks. Inference on this model is challenging due to the complexity of both the underlying network structure and the presence of non-conjugacy in the model. Scalable MCMC on the model is possible through the use of Stochastic Gradient Riemannian Langevin Dynamics (SGRLD). In this paper, we develop a novel and scalable Stochastic Gradient Variational Inference (SG-VI) algorithm and compare it to SGRLD inference. Similarly to MCMC inference, handling non-conjugacy in the S-AGM is a significant challenge for developing an SG-VI and requires the application of stochastic Monte Carlo estimation. We carry out a thorough empirical comparison of the SG-VI and SGRLD approaches, and draw some general conclusions about scalable inference on the S-AGM.

1. Introduction

Communities in social networks consist of clusters of nodes in which the edge density is high, corresponding to groups of people who tend to have many mutual friendship links. It may be observed that in real social networks communities tend to be overlapping and that any individual node may belong to multiple communities, corresponding to the different social groupings that an individual may be part of. Hence, identifying overlapping community structure is a problem with a lot of interest to social network analysts. The Affiliation Graph Model (AGM) was introduced in (Yang and Leskovec, 2012), as an overlapping community model, exhibiting *pluralistic homophily*, the characteristic that the probability that two nodes are connected monotonically increases with the number of community memberships that they share. In (Laitonjam et al., 2019), a first scalable yet fully Bayesian approach to inference on the AGM was introduced. To obtain a tractable inference method, the AGM was modified to allow for soft community memberships and the new model was called *soft*-AGM (S-AGM). Given a node i and a community k, the latent binary membership variable z_{ik} of the AGM was replaced with a degree of membership variable w_{ik} such that $0 \leq w_{ik} \leq 1$. It was shown that the modified model maintained the pluralistic homophily characteristics of the original model and inference on networks of $> 10^6$ nodes was demonstrated, by using an algorithm implemented on a GPU, exploiting the inherent parallelism

of the S-AGM. To achieve this scaling the MCMC inference was developed using Stochastic Gradient Riemannian Langevin Dynamics (SGRLD).

In this paper, we follow this work to explore another scalable inference method using a variational inference algorithm for the S-AGM. Because of the presence of non-conjugacy in the model, the mean-field approach does not yield closed-form updates for the variational parameters. Instead, we develop a stochastic gradient optimisation, which relies on re-parameterisation to approximate some of the required expressions. The resulting algorithm is compared with SGRLD inference. As well as examining its performance in the context of the overlapping community problem, we make some general observations about these two contrasting approaches to achieving scalable inference on real world networks.

2. Generative Model

The generative process of the S-AGM is presented in Algorithm 1 and its corresponding graphical model is given in Figure 1. Assume there are N data points and K communities. We define the following notation: π_k is the probability of an edge between any pair of nodes in community k; π_ϵ is a background probability of an edge between any node pairs regardless of community membership, capturing noisy edges; w_{ik} is the probability of node i belonging to community k; α_k is a parameter that captures the node density within community k; and W, π, α represent the matrix $\{w_{ik}\}$ and the vectors π_k, and α_k, respectively; finally, p_{ij} is the edge probability between nodes i and j. The functional form of this edge probability, p_{ij}, distinguishes the S-AGM and AGM, and is defined as

$$p_{ij} = 1 - (1 - \pi_\epsilon) \prod_{k=1}^{K} (1 - \pi_k w_{ik} w_{jk}) \,. \tag{1}$$

where, in the original AGM, rather than soft community affiliations w_{ik}, hard community assignments $z_{ik} \in \{0, 1\}$, drawn from a *Bernoulli* distribution are used.

Representing the network of size N as an $N \times N$ adjacency matrix A with elements $a_{ij} = 1$ when there is an edge between nodes i and j and 0 otherwise, then the likelihood of an undirected network, given the model is

$$p(\mathrm{A} \mid \mathrm{W}, \pi) = \prod_{ij:i<j} p(a_{ij} \mid w_i, w_j, \pi)$$

where $p(a_{ij} \mid w_i, w_j, \pi) = p_{ij}^{a_{ij}} (1 - p_{ij})^{1 - a_{ij}}$.

Given the hyperparameters η and β, the joint probability of the model is,

$$p(\mathrm{A}, \mathrm{W}, \alpha, \pi \mid \eta, \beta) = \prod_{ij:i<j} p(a_{ij} \mid w_i, w_j, \pi) \prod_i \prod_k p(w_{ik} \mid \alpha_k) \prod_k p(\pi_k \mid \eta_{k0}, \eta_{k1}) \prod_k p(\alpha_k \mid \beta_0, \beta_1) \,.$$

This equation will be used to derive the SG-VI approach.

Algorithm 1 Generative process model

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

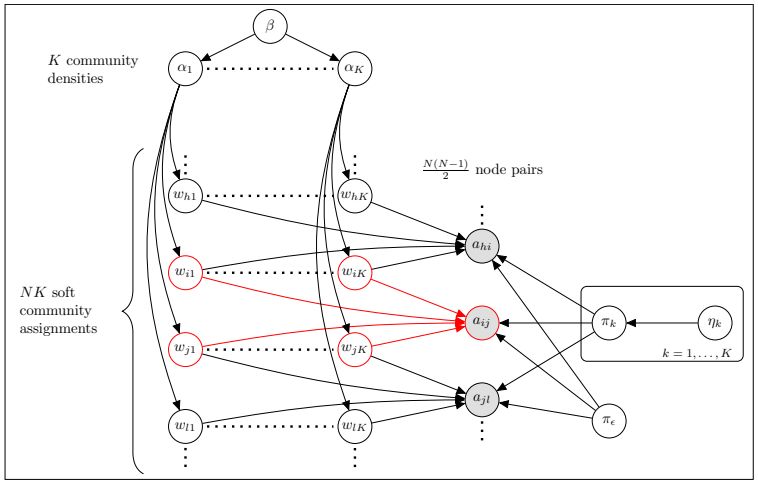

Figure 1: Graphical Model of S-AGM (with $1 \leq h < i < j < l \leq N$).

3. Inference

SGRLD (Patterson and Teh, 2013) is a scalable MCMC algorithm for Bayesian models. The SGRLD algorithm of the S-AGM developed in (Laitonjam et al., 2019) is summarised in Algorithm 2. It relies on mini-batch samples of nodes $\mathcal{V}_i^t$ and edges $\mathcal{E}^t$ on each round t, to update re-parameterisations w'_{ikm} and π'_{km} of the model parameters w_{ik} and π_k, through a stochastic gradient update. The α_k parameter is updated using Gibbs sampling. In particular, for each node i and community k, the re-parameterisations are defined as:

$$\pi_k = \frac{\pi'_{k0}}{\pi'_{k0} + \pi'_{k1}}, \quad w_{ik} = \frac{w'_{ik0}}{w'_{ik0} + w'_{ik1}},$$

where, for $m \in \{0, 1\}$, $\pi'_{km} \sim \text{Gamma}(\eta_{km}, 1)$, $\quad w'_{ikm} \sim \text{Gamma}(\gamma_{km}, 1)$ and $\gamma_{k0} = \alpha_k$ and $\gamma_{k1} = 1$. The update formulae depend on the preconditioning matrices $G^{-1}(\pi') = \text{diag}(\pi')$ and $G^{-1}(\text{W}') = \text{diag}(\text{W}')$, where $\pi' = \{\pi'_{k0}, \pi'_{k1}\}_{k=1,\ldots,K}$ and $\text{W}' = \{w'_{ik0}, w'_{ik1}\}_{i=1,\ldots,N;k=1,..,K}$.

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 $\mathcal{V}_i^t$
    $w'^{*}_{ikm} = \left| w'_{ikm} + \frac{\rho_w^{(t)}}{2} \times \tilde{g}(w'_{ikm}) + b(w'_{ikm}) \right|$
for $k = 1 : K$ **do** ▷ utilizing the sampled $\mathcal{E}^t$
  $\pi'^{*}_{km} = \left| \pi'_{km} + \frac{\rho_\pi^{(t)}}{2} \times \tilde{g}(\pi'_{km}) + b(\pi'_{km}) \right|$
for $k = 1 : K$ **do**
  $\alpha_k | w_{\cdot k} \sim \text{Gamma}\left(N + \beta_0, \beta_1 - \sum_i \log(w_{ik})\right)$

To summarise the notation of Algorithm 2:

- $\tilde{g}(w'_{ikm})$ is the gradient of the log posterior of W′ wrt w'_{ikm} in the Riemannian manifold with tensor metric $G(\text{W}')$,
- $b(w'_{ikm})$ is the corresponding term related to the Browning motion in the Riemannian manifold with tensor metric $G(\text{W}')$,
- $\tilde{g}(\pi'_{km})$ is the gradient of the log posterior of π' wrt π'_{km} in the Riemannian manifold with tensor metric $G(\pi')$,
- $b(\pi'_{km})$ is the corresponding term related to the Browning motion in the Riemannian manifold with tensor metric $G(\pi')$,
- ρ_w, ρ_π are the step-sizes.

3.1. Stochastic Gradient Variational Inference (SG-VI)

Variational inference infers a Bayesian model by proposing a variational distribution that approximates the target posterior well while being easier to compute. To achieve the computational simplicity, it is common to apply the mean field assumption to relax the variable dependence within the variational distribution. For Bayesian models with conditionally conjugate parts only, the update of the variational parameter is possible through a traditional coordinate ascent algorithm or using Stochastic Variational Inference (SVI) (Hoffman et al., 2013) that exploits the natural gradient of the variational parameter. For S-AGM, due to the presence of non-conjugacy in the model, SVI is not possible for all variational parameters.

3.1.1. SG-VI for S-AGM

With the mean field assumption, we consider the variational distribution for π_k, w_{ik} and α_k as

$$q(\text{W}, \pi, \alpha) = \prod_i \prod_k q(w_{ik} \mid \phi_{ik0}, \phi_{ik1}) \prod_k q(\pi_k \mid \lambda_{k0}, \lambda_{k1}) \prod_k q(\alpha_k \mid \tau_{k0}, \tau_{k1})$$

where the variational distributions of π_k and w_{ik} are set as Beta distributions and that of α_k as a Gamma distribution. The variational parameters, $\lambda = \{\lambda_{k0}, \lambda_{k1}\}_{k=1,\ldots,K}$, $\tau = \{\tau_{k0}, \tau_{k1}\}_{k=1,\ldots,K}$, $\phi = \{\phi_{ik0}, \phi_{ik1}\}_{i=1,\ldots,N;k=1,\ldots,K}$ of the these distributions are chosen so that the KL-divergence between q and the true posterior is minimised. This is equivalent to maximizing the following evidence lower bound (ELBO) (Blei et al., 2017) over the variational parameters:

$$\begin{aligned}
\mathcal{L}(\phi, \lambda, \tau) &= \mathrm{E}_q[\log p(\mathrm{A}, \mathrm{W}, \alpha, \pi \mid \eta, \beta)] - \mathrm{E}_q[\log q(\mathrm{W}, \pi, \alpha)] \\
&= \sum_{ij:i<j} \mathrm{E}_q[\log p(a_{ij} \mid w_i, w_j, \pi)] + \sum_i \sum_k \mathrm{E}_q[\log p(w_{ik} \mid \alpha_k)] \\
&+ \sum_k \mathrm{E}_q[\log p(\pi_k \mid \eta_{k0}, \eta_{k1})] + \sum_k \mathrm{E}_q[\log p(\alpha_k \mid \beta_0, \beta_1)] \\
&- \sum_i \sum_k \mathrm{E}_q[\log q(w_{ik} \mid \phi_{ik0}, \phi_{ik1})] \\
&- \sum_k \mathrm{E}_q[\log q(\pi_k \mid \lambda_{k0}, \lambda_{k1})] - \sum_k \mathrm{E}_q[\log q(\alpha_k \mid \tau_{k0}, \tau_{k1})] \qquad (2)
\end{aligned}$$

3.2. Optimisation Strategy

To maximise Equation (2), we would like to solve for values of the variational parameters at which the gradient vanishes. Unfortunately, a closed form solution is not possible and we resort instead to a stochastic gradient ascent algorithm. This in turn poses significant challenges which must be addressed in order to develop a tractable solver. A gradient ascent algorithm requires that the variational parameters are updated at each step in the direction of the gradient of the objective, which requires the computation of gradients of the form:

$$\nabla_\theta \mathrm{E}_{q(z|\theta)}[f(z)]\,.$$

However, some of the expectations in Equation (2), in particular $\mathrm{E}_q[\log p(a_{ij} \mid w_i, w_j, \pi)]$, are not tractably solvable and so the gradients for the updates of π and w are intractable. The solution is to resort to estimating the gradient instead.

An unbiased estimate of the gradient can be computed with Monte Carlo estimation using the "REINFORCE" or *score function method* (Williams, 1992; Glynn and L'ecuyer, 1995; Ranganath et al., 2013; Titsias and Lázaro-Gredilla, 2014), such that, for any random number z drawn from the distribution $q(z \mid \theta)$, the score gradient of a function $f(z)$ is

$$\nabla_\theta \mathrm{E}_{q(z|\theta)}[f(z)] = \mathrm{E}_{q(z|\theta)}[f(z) \nabla_\theta \log q(z \mid \theta)]\,.$$

However, this method suffers from high variance, requiring a variance reduction technique, such as Rao-Blackwellization or designing an appropriate control variate (Ranganath et al., 2013), to ensure the convergence of the algorithm. The variance of the Monte Carlo estimate of the gradient could instead be reduced with the re-parameterisation trick (Price, 1958; Kingma and Welling, 2013; Salimans et al., 2013). Here, z is re-parameterised as $t(\epsilon; \theta)$ where ϵ is drawn from some distribution $r(\epsilon)$ which is independent of θ. The re-parameterisation gradient can then be written as

$$\nabla_\theta \mathrm{E}_{q(z|\theta)}[f(z)] = \mathrm{E}_{r(\epsilon)}[\nabla_\theta f(t(\epsilon; \theta))] = \mathrm{E}_{r(\epsilon)}[\nabla_z f(z) \nabla_\theta t(\epsilon; \theta)]\,. \qquad (3)$$

However, as $\pi_k \sim \text{Beta}(\lambda_{k0}, \lambda_{k1})$, $w_{ik} \sim \text{Beta}(\phi_{ik0}, \phi_{ik1})$ and since there is no such re-parameterisation of the Beta distribution, we cannot use this approach. Instead, as the intractable gradients can be expressed in terms of an expectation of Gamma distributions, by re-parameterisation of the corresponding Beta distributions, we can apply the *generalised* re-parameterisation gradient (G-REP) (Ruiz et al., 2016) method for the Gamma distribution. For a variational parameter θ, this amounts to writing the gradient of the expectation as

$$\nabla_\theta \mathrm{E}_{q(z|\theta)}[f(z)] = g_\theta^{\text{rep}} + g_\theta^{\text{corr}} \,,$$

where,

$$g_\theta^{\text{rep}} = \mathrm{E}_{q(z|\theta)}[\nabla_z f(z) h(\epsilon;\theta)] \tag{4}$$

$$g_\theta^{\text{corr}} = \mathrm{E}_{q(z|\theta)}[f(z)\{\nabla_z \log q(z \mid \theta) h(\epsilon;\theta) + \nabla_\theta \log q(z \mid \theta) + u(\epsilon;\theta)\}] \tag{5}$$

and $\epsilon = t^{-1}(z;\theta)$ for some invertible transformation that is weakly dependent on θ; $h(\epsilon;\theta) = \nabla_\theta t(\epsilon;\theta)$; and $u(\epsilon;\theta) = \nabla_\theta \log J(\epsilon;\theta)$, where J is the Jacobian of the transformation. Here, g_θ^{rep} corresponds to the re-parameterization term as in eq. (3) and g_θ^{corr} corresponds to the correction term for sampling ϵ from a distribution depended on θ. In this form, the above expectations can be estimated as $\hat{g}_\theta^{\text{rep}}$ and $\hat{g}_\theta^{\text{corr}}$, with the Monte Carlo method by using sampling from the distribution $q(z \mid \theta)$.

In the case of τ, we can exploit conditional conjugacy to compute the natural gradient exactly as in SVI.

Mini-batch Stochastic Gradient Ascent: Again for tractability, we adopt a *mini-batch* stochastic gradient (SG) ascent algorithm, to estimate the gradients. In fact, we apply pre-conditioned SG ascent, so that the general update rule for parameters θ is given by

$$\theta' = \theta + \rho\, \mathrm{G}^{-1}(\theta)\hat{g}(\theta)$$

where ρ is the learning rate, $\hat{g}$ is an unbiased estimate of the gradient at θ and $\mathrm{G}(\theta)$ is a symmetric, positive-definite pre-conditioner. In the case of τ, we choose $\mathrm{G}(\tau)$ to be the Fisher Information matrix, thus using the natural gradient as the ascent direction of this parameter. In the case of ϕ and λ, we take $\mathrm{G}^{-1}(\lambda) = \text{diag}(\lambda)$ and $\mathrm{G}^{-1}(\phi) = \text{diag}(\phi)$.

The mini-batch is selected in the same manner as for the SGRLD algorithm of (Laitonjam et al., 2019). On each iteration, a mini-batch of node pairs $\mathcal{E}_t$ is selected. The selection is carried out using a stratified sampling strategy (Gopalan et al., 2012) that prioritises linked pairs. The gradients wrt ϕ_{ik0}, ϕ_{ik1} are approximated for all $i \in \mathcal{E}_t$, using another mini-batch of nodes $\mathcal{V}_i$ sampled for each i, to select node pairs (i, j) with which the gradients are estimated. In the case of $\lambda_{k0}, \lambda_{k1}$, they are estimated using the node pairs $(i, j) \in \mathcal{E}_t$. We note that this strategy implies a doubly stochastic unbiased estimation (mini-batch with Monte Carlo estimation) of the ascent step for the parameters λ and ϕ.

3.3. Parameter Updates

As a first step, since ϕ and λ are restricted to positive values only, to allow for unconstrained optimisation, we make a change of variables to $\overline{\phi}$ and $\overline{\lambda}$ using

$$\overline{\phi} = \log(\exp(\phi) - 1) \qquad \overline{\lambda} = \log(\exp(\lambda) - 1)\,, \tag{6}$$

where $\overline{\phi}, \overline{\lambda} \in (-\infty, \infty)$. The optimisation is carried out wrt $\overline{\phi}$ and $\overline{\lambda}$.

We also use the fact that, due to independence, the full expectation over q can be decomposed as $\mathrm{E}_q[f] = \mathrm{E}_{q(v)}\big[\mathrm{E}_{q^{-v}}[f]\big]$, where q^{-v} represents the distribution over all variables except v.

3.3.1. Gradient wrt ϕ :

The gradient of ELBO wrt ϕ with the preconditioner is

$$\mathrm{G}^{-1}(\phi)\hat{g}(\phi^{(t-1)})$$

where $\hat{g}(\phi)$ is the estimate of the gradient of the ELBO wrt ϕ. Using the fact that $q(w_{ik})$ is a Beta distribution, for $m \in \{0, 1\}$, the partial derivative wrt ϕ_{ikm} of the component due to term $\sum_i \sum_k \mathrm{E}_q[\log q(w_{ik} \mid \phi_{ik0}, \phi_{ik1})]$, can be computed exactly, as

$$(\phi_{ikm} - 1)\Psi'(\phi_{ikm}) - (\phi_{ik0} + \phi_{ik1} - 2)\Psi'(\phi_{ik0} + \phi_{ik1})\,.$$

The remaining part of the gradient can be written

$$\nabla_{\phi_{ikm}} \mathrm{E}_{q(w_{ik}|\phi)}\big[f_1(w_{ik}) + f_2(w_{ik})\big] \tag{7}$$

with

$$f_1(w_{ik}) = \sum_{i \neq j} \mathrm{E}_{q^{-w_{ik}}}[\log p(a_{ij} \mid w_i, w_j, \pi)]$$

and, computing the expected value of $\log p(w_{ik} \mid \alpha_k)$ over $q(\alpha_k \mid \tau)$,

$$f_2(w_{ik}) = \Psi(\tau_{k0}) - \log(\tau_{k1}) + \big(\tau_{k0}/\tau_{k1} - 1\big)\log w_{ik}\,.$$

We resort to G-REP to approximate Equation (7). Here, Ψ and Ψ' are the digamma function and polygamma function of order 1 respectively.

3.3.2. Gradient wrt λ :

The gradient of ELBO wrt λ with the preconditioner is

$$\mathrm{G}^{-1}(\lambda)\hat{g}(\lambda^{(t-1)})$$

where $\hat{g}(\lambda)$ is the estimate of the gradient of the ELBO wrt λ. Using the fact that $q(\pi_k)$ is a Beta distribution, for $m \in \{0, 1\}$, the partial derivative wrt λ_{km} of the component due to term $\sum_k \mathrm{E}_q[\log q(\pi_k \mid \lambda_{k0}, \lambda_{k1})]$, can be computed exactly, as

$$(\lambda_{km} - 1)\Psi'(\lambda_{km}) - (\lambda_{k0} + \lambda_{k1} - 2)\Psi'(\lambda_{k0} + \lambda_{k1})$$

The remaining part of the gradient can be written as

$$\nabla_{\lambda_{km}} \mathrm{E}_{q(\pi_k|\lambda)}\big[f(\pi_k)\big] \tag{8}$$

where

$$f(\pi_k) = \sum_{i \neq j} \mathrm{E}_{q^{-\pi_k}}[\log p(a_{ij} \mid w_{i\cdot}, w_{j\cdot}, \pi)] + \log p(\pi_k \mid \eta_{k0}, \eta_{k1})$$

Again G-REP is used to get an unbiased estimate of Equation (8).

3.3.3. Update equations for τ

$$\tau_k^{(t)} = \tau_k^{(t-1)} + \rho_\tau^{(t)} \mathrm{G}^{-1}(\tau_k)\hat{g}(\tau_k^{(t-1)})$$

where $\hat{g}(\tau_k)$ is the estimated gradient wrt $\tau_k = \{\tau_{k0}, \tau_{k1}\}$, which can be evaluated using conditional conjugacy. Following (Hoffman et al., 2013), with G chosen as the Fisher Information matrix, we can obtain an exact expression for the natural gradient as:

$$G^{-1}(\tau_{k0}, \tau_{k1}) \begin{bmatrix} \nabla_{\tau_{k0}} \mathcal{L} \\ \nabla_{\tau_{k1}} \mathcal{L} \end{bmatrix} = \begin{bmatrix} N + \beta_0 - \tau_{k0} \\ \beta_1 - \sum_i (\Psi(\phi_{ik0}) - \Psi(\phi_{ik0} + \phi_{ik1})) - \tau_{k1} \end{bmatrix}.$$

Finally, the estimators $\hat{g}(\tau_k)$ are obtained from the mini-batch $\mathcal{E}^t$ by summing in the above expression only over nodes $i \in \mathcal{E}^t$, and scaling the sum by $N/|i \in \mathcal{E}^t|$.

Algorithm 3 SG-VI for the S-AGM at iteration t

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 $\mathcal{V}_i^t$
    $\overline{\phi}_{ik0}^{(t)} = \overline{\phi}_{ik0}^{(t-1)} + \rho_\phi^{(t)} \times \hat{g}(\overline{\phi}_{ik0})$
    $\overline{\phi}_{ik1}^{(t)} = \overline{\phi}_{ik1}^{(t-1)} + \rho_\phi^{(t)} \times \hat{g}(\overline{\phi}_{ik1})$
for $k = 1 : K$ **do** ▷ utilizing the sampled $\mathcal{E}^t$
  $\overline{\lambda}_{k0}^{(t)} = \overline{\lambda}_{k0}^{(t-1)} + \rho_\lambda^{(t)} \times \hat{g}(\overline{\lambda}_{k0})$
  $\overline{\lambda}_{k1}^{(t)} = \overline{\lambda}_{k1}^{(t-1)} + \rho_\lambda^{(t)} \times \hat{g}(\overline{\lambda}_{k1})$
for $k = 1 : K$ **do** ▷ utilizing the sampled $\mathcal{E}^t$
  $\tau_{k0}^{(t)} = \tau_{k0}^{(t-1)} + \rho_\tau^{(t)} \times \hat{g}(\tau_{k0})$
  $\tau_{k1}^{(t)} = \tau_{k1}^{(t-1)} + \rho_\tau^{(t)} \times \hat{g}(\tau_{k1})$

The pseudo code for one iteration of the update algorithm is given in Algorithm 3. Here $\hat{g}(x)$ represents the mini-batch stochastic estimates and ρ represents the step size. The detailed derivation of the gradients is given in supplemental material.

4. Experiments

We are interested in understanding the convergence characteristics of the SG-VI algorithm and in comparing it to SGRLD inference. In particular, we identify the following research questions in relation to the SG-VI algorithm:

R1: Is preconditioning necessary for SG-VI convergence?

R2: How does SG-VI convergence depend on the number of Monte Carlo estimations carried out to compute $\hat{g}(\phi)$ and $\hat{g}(\lambda)$?

R3: How does SG-VI convergence depend on the mini-batch size?

Finally, we have the research question relating to the two algorithms:

R4: With which algorithm, SGRLD or SG-VI, can most scalable performance be achieved?

We report comparisons between SGRLD and SG-VI on a number of different synthetic and real-world networks. For both SGRLD and SG-VI, we set the step-size schedule as $\rho^{(t)} = a(1 + t/b)^{-c}$ where $b = 1,000$ and $c = 0.55$ and a is varied depending on the experiment. For all experiments, we fix the background edge probability π_ϵ as $1e^{-5}$. All the experiments are run on a 2.2 GHz Intel Core i7 processor. Both the algorithms are implemented in `Tensorflow` (Abadi et al., 2015). [1]

For networks with ground-truth communities, the overlapping Normalised Mutual Information (NMI) (Lancichinetti et al., 2009) is used to measure the closeness of the found communities to the ground-truth. Note that since the S-AGM outputs soft community assignments w_{ik}, we must threshold in order to obtain the hard community assignment required by the NMI. We also compare solutions using missing link prediction on a hold-out set of link and non-link pairs and computing the AUC of the ROC. Finally, we compute the perplexity score on the hold-out set of test node pairs.

4.1. Synthetic Networks

Firstly, we consider synthetic networks. For both SGRLD and SG-VI, we use a mini-batch size of $|\mathcal{E}_t| = |\mathcal{V}_i| = 20$. When $\mathcal{E}_t$ corresponds to a link set, its size is the degree of the randomly selected node. To compute performance metrics, we hold out a test set consisting of 10% of the number of links in the network and an equal number of non-link node pairs. For SG-VI, Monte Carlo estimates are computed using 5 samples and each run is over $10,000$ iterations. For SGRLD, after a burn-in of $5,000$ iterations, samples are collected from the next $5,000$ iterations with a lag of 100.

We consider two different synthetic networks:

- *Net-AGM*: Since S-AGM is a soft community assignment version of the Affiliation Graph Model (AGM), we generate a network with 75 nodes and $K = 4$ (overlapping communities) using the generative process of the AGM.
- *Net-aMMSB*: We consider the synthetic network used in (Gopalan et al., 2012; Li et al., 2016) which has 75 nodes and is generated from the assortative Mixed Membership Stochastic Blockmodel (aMMSB).

Preconditioning in the SG-VI For $K = \{10, 20, 30, 40\}$ and *Net-AGM*, we plot the histogram of AUC-ROC of predicting missing links of the validation node pairs after $2,000$ iterations for various initial step-sizes by varying a over the values $\{1, 0.1, 0.01, 0.001\}$, using SG-VI with and without the preconditioner of λ and ϕ. From Figure 2, we can see that by using the preconditioner the convergence is faster. The AUC-ROC score is above 0.8 for many initial step-sizes after $2,000$ iterations. However, without the preconditioner, most initial step-sizes have only obtained an AUC-ROC of around 0.5 at this point.

Similarly, for *Net-aMMSB*, preconditioning is necessary to boost performance. From Figure 3, the portion of initial step-size converting to an AUC-ROC of greater than 0.8 after $2,000$ iterations is higher with the preconditioner, than without.

1. The codes to reproduce the results are available in https://github.com/apple-apple-star/SG-VI-for-SAGM.

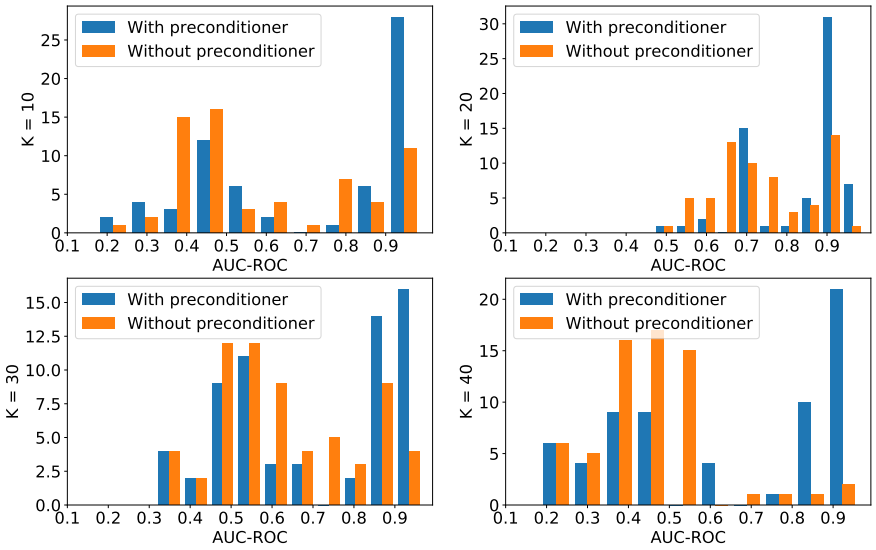

Figure 2: Convergence comparison on *Net-AGM*.

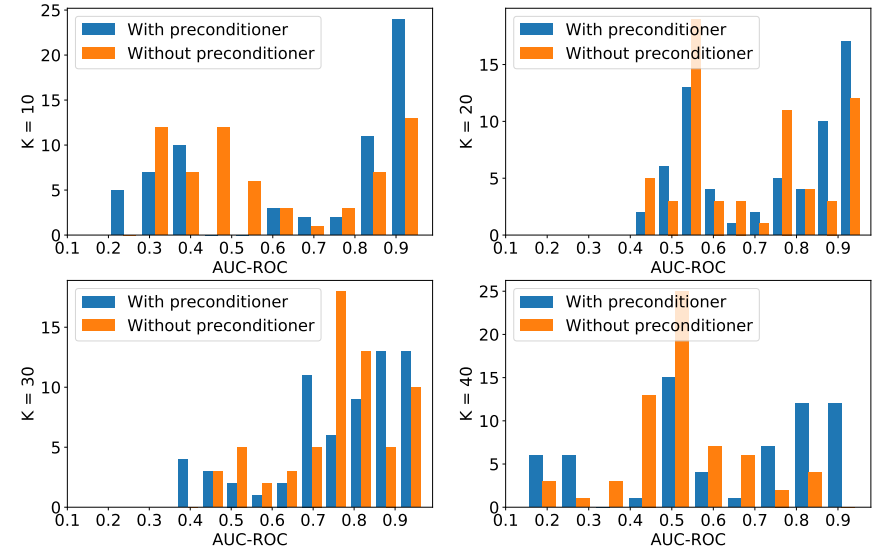

Figure 3: Convergence comparison on *Net-aMMSB*.

Performance comparison with SGRLD The initial parameter a of the step size for each parameter is selected by a line search over the values $\{1, 0.1, 0.01, 0.001\}$, choosing

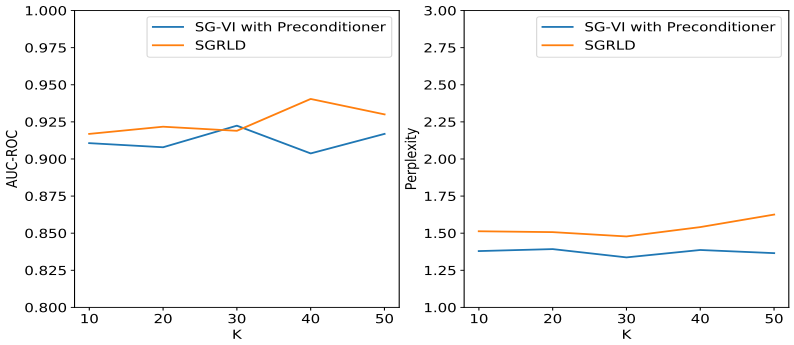

Figure 4: AUC-ROC and perplexity on *Net-AGM*.

the combination that gives best performance on a validation set and may be different for $\rho_\phi^{(t)}, \rho_\lambda^{(t)}$, and $\rho_\tau^{(t)}$.

We compare the missing links prediction with AUC-ROC and perplexity score on the hold-out test node pairs. The results for *Net-AGM* are shown in Figure 4 and for *Net-aMMSB* in Figure 5. The performance of both inference algorithms is similar. For *Net-AGM*, the AUC-ROC is above 0.9 and the perplexity between 1.2 and 1.5. There is a slight improvement in terms of predicting missing links for SGRLD while in terms of perplexity, SG-VI is performing better. We also observe for *Net-aMMSB* that the performance of SGRLD and SG-VI for S-AGM is similar. It is interesting to note that this is different to the results obtained in (Li et al., 2016), where an SGRLD inference on the aMMSB model clearly out-performs an SVI inference on this network.

In the case of *Net-AGM*, where the ground truth communities are known, we can measure performance with the NMI. We set $K = 4$, and set various thresholds on the values of w_{ik} to compute the hard community assignments required for the NMI. The best NMI over 5 random runs is considered. From Figure 6, we see that both algorithms obtain a perfect NMI score for an appropriate setting of the threshold. The NMI decreases more sharply for SGRLD when the threshold is high.

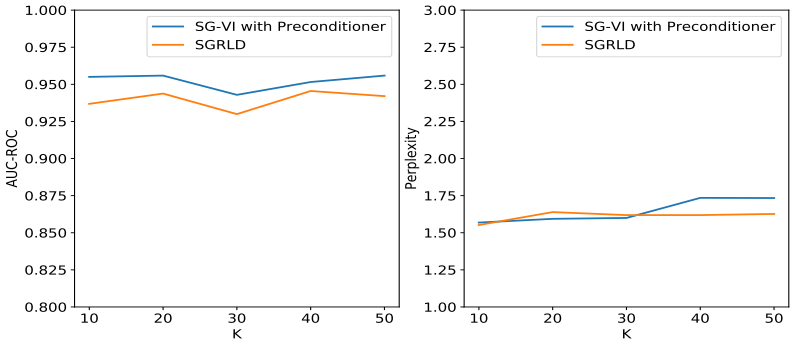

Figure 5: AUC-ROC and perplexity on *Net-aMMSB*.

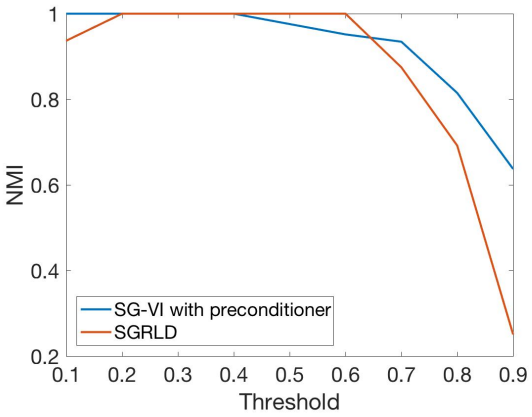

Figure 6: Best NMI of 5 random runs for various threshold on *Net-AGM*.

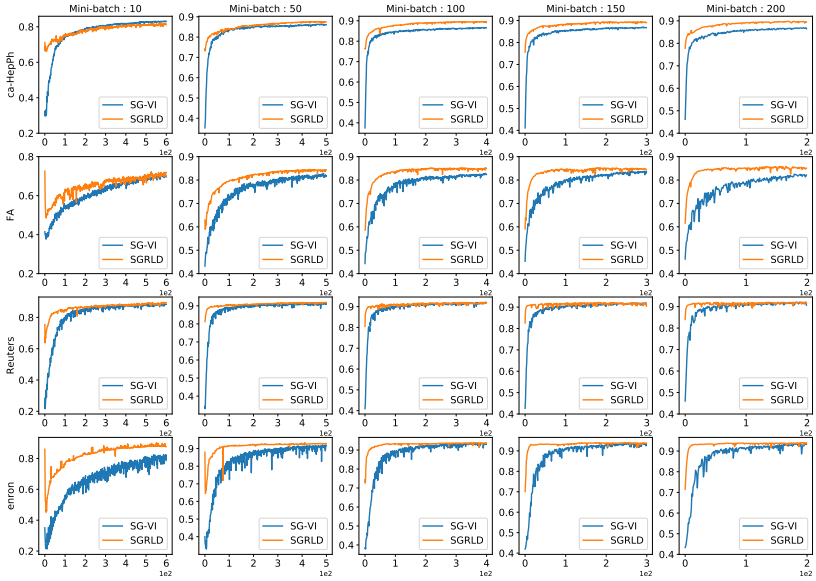

Figure 7: AUC-ROC vs iterations on real world networks with SG-VI and SGRLD

4.2. Real world networks

In this section, we examine performance on a number of real world networks that have more than $10,000$ nodes; namely ca-HepPh (Leskovec and Krevl, 2014) (12,008 nodes and 118,521 edges), FA (Nelson et al., 2004) (10,299 nodes and 61,677 edges), Reuters (Corman et al., 2002) (13,314 nodes and 148,038) and Enron (Leskovec and Krevl, 2014) (36,692 nodes and 183,831 edges). For the experiments, we take $K = 50$ and use a hold out test set consisting of 10% of the links in the network with an equal number of non-links.

Table 1: Comparison of time taken (in sec) for fixed number of iterations between SGRLD and SG-VI.

	#Iterations (Mini-Batch)	60,000 (10)	50,000 (50)	40,000 (100)	30,000 (150)	20,000 (200)
ca-HepPh	SGRLD	629.48	600.17	621.76	641.46	**573.68**
	SG-VI	821.20	1158.11	1834.25	2569.99	**2919.47**
FA	SGRLD	587.91	533.74	558.08	574.50	**526.14**
	SG-VI	726.64	1013.15	1742.50	2354.10	**2747.16**
Reuters	SGRLD	691.95	665.90	672.03	670.97	**596.79**
	SG-VI	970.09	1326.09	1981.01	2693.40	**2945.52**
Enron	SGRLD	1393.75	1256.64	1150.45	1029.25	**848.63**
	SG-VI	1757.21	1964.01	2410.69	2923.33	**3281.47**

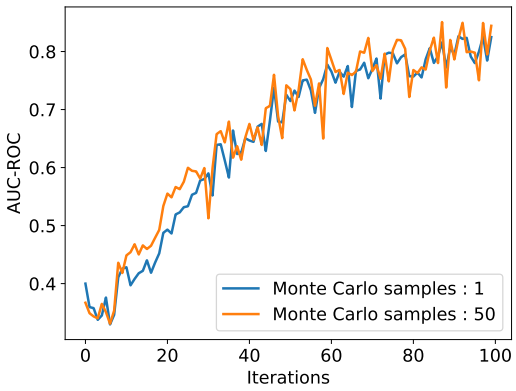

Figure 8: AUC-ROC vs iterations on Enron network with SG-VI

Monte Carlo Estimation in SG-VI We use Enron to examine the impact of the Monte Carlo sample size used in estimating $\hat{g}(\phi)$ and $\hat{g}(\lambda)$. For this experiment we fix the mini-batch size to 50. In Figure 8, the convergence plot of AUC-ROC over $10,000$ iterations with the AUC-ROC computed every 100 iterations is shown, where the estimation has been carried out using a single sample and using 50 samples. We can see that for a network of this size, a single sample is sufficient (note that with a single sample per estimate the simulation takes 378.87 sec, while with 50 samples, it takes 3226.82 sec). Hence, for the remaining experiments using SG-VI, we consider single Monte Carlo estimates in Generalized Re-parameterization for updating $\overline{\lambda}$ and $\overline{\phi}$.

Impact of Mini-batch Size For the SG-VI algorithm, we adjust the number of iterations according to mini-batch size as follows. The number of iterations is chosen as $60,000$ for mini-batch size 10; $50,000$ iterations for batch size 50; $40,000$ iterations for batch size 100; $30,000$ iterations for batch size 150; and $20,000$ iterations for batch size 200; as convergence occurs with fewer iterations as the mini-batch size increases. For this experiment, we have chosen the initial step-size as $a = 1.0$ for $\overline{\lambda}$, $\overline{\phi}$ and τ.

Table 2: AUC-ROC score for various mini-batch sizes with SG-VI.

Mini-Batch size	10	50	100	150	200
ca-HepPh	0.8295	0.8610	0.8653	0.8674	0.8646
FA	0.6960	0.8172	0.8238	0.8357	0.8213
Reuters	0.8839	0.9093	0.9190	0.9105	0.9068
Enron	0.7865	0.9161	0.9302	0.9348	0.9333

In Table 2, the AUC-ROC of missing links prediction of the final iteration by SG-VI is given for various mini-batch sizes. We can see there is an increase in AUC-ROC with increase in mini-batch sizes with the smallest AUC-ROC scores occurring at mini-batch size of 10 and mini-batch sizes above 100 having similar AUC-ROC scores.

We also compare SGRLD with varying mini-batch sizes. Here, again accounting for faster convergence when the mini-batch size is larger, we adjust the length of burn in with the mini-batch size. In particular, we have a burn-in of $50,000$ iterations for mini-batch size 10; $40,000$ iterations for mini-batch size 50; $30,000$ iterations for mini-batch size 100; $20,000$ iterations for mini-batch size 150; and $10,000$ iterations for mini-batch size 200. After burn-in we collect $10,000$ samples with a lag of 100. An initial step size of $a = 0.001$ is chosen for sampling both π and w.

In Table 3, the AUC-ROC of missing links prediction from the samples collected after burn-in at lags of 100 by SGRLD is given for various mini-batch sizes. Again, we can see that there is a slight increase in AUC-ROC with increase in mini-batch size with the smallest AUC-ROC observed at mini-batch size of 10 and largest at mini-batch size of 200.

Comparison between SG-VI and SGRLD In Figure 7, the convergence of AUC-ROC for missing links prediction on real world networks after every 100 iterations is shown for SG-VI and SGRLD. From Figure 7, we can see that although both the algorithms converge to the same value, the convergence of SGRLD is faster than SG-VI. From the figure we can see that choosing a mini-batch size of 100 is almost same as choosing larger mini-batch size.

Since SG-VI is computationally more expensive than SGRLD, from Table 1, the time taken for a fixed number of iterations is less for SGRLD than SG-VI. With the increase in mini-batch size, SG-VI takes much more time when compared to SGRLD.

5. Discussion

The goal of this paper has been to compare two approaches to scalable inference on a complex Bayesian model, which exhibits non-conjugacy. A general conclusion can be reached

Table 3: AUC-ROC score for various mini-batch sizes with SGRLD.

Mini-Batch size	10	50	100	150	200
ca-HepPh	0.8195	0.8762	0.8969	0.8938	0.8990
FA	0.7222	0.8470	0.8541	0.8550	0.8616
Reuters	0.8942	0.9208	0.9231	0.9275	0.9306
Enron	0.8908	0.9336	0.9445	0.9489	0.9513

that, for this model, SGRLD is a better option than SG-VI. This agrees to some extent with the findings of (Li et al., 2016) who developed an SGRLD inference for the aMMSB model. However, we do not see the general superiority of the SGRLD over SG-VI in terms of quality upon convergence. It is noteworthy that both approaches have a similar computational framework, requiring parameter updates over mini-batches of nodes and edges, which we have constructed in the same way for each approach. For SG-VI, there is the additional task of selecting the number of Monte-Carlo samples in the gradient estimation and our results show that even a single sample is sufficient. Although there is no such sampling in SGRLD, an appropriate burn-in must be chosen.

6. CONCLUSION

We have presented a scalable inference algorithm for the *Soft-Affiliation Graph Model (S-AGM)* using stochastic gradient variational inference (SG-VI) which maximizes the evidence lower bound of the model. For the non-conditional conjugate terms of the model, we have introduced a preconditioner for the update which improves convergence and developed a generalised re-parameterisation scheme to estimate the required gradients. This scalable variational inference has been compared with a scalable MCMC inference that uses Stochastic Gradient Riemannian Langevin Dynamics (SGRLD). Both algorithms converge similarly for various real world and synthetic networks, although for larger networks, convergence of SGRLD is faster than SG-VI.

Acknowledgments

This research is supported by Science Foundation Ireland grant SFI/12/RC/2289_P2.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. doi: 10.1080/01621459.2017.1285773. URL https://doi.org/10.1080/01621459.2017.1285773.

Steven R Corman, Timothy Kuhn, Robert D McPhee, and Kevin J Dooley. Studying complex discursive systems. centering resonance analysis of communication. *Human communication research*, 28(2):157–206, 2002.

Peter W Glynn and Pierre L'ecuyer. Likelihood ratio gradient estimation for stochastic recursions. *Advances in applied probability*, 27(4):1019–1053, 1995.

Prem K Gopalan, Sean Gerrish, Michael Freedman, David M Blei, and David M Mimno. Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2012.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Nishma Laitonjam, Wěipéng Huáng, and Neil J. Hurley. A soft affiliation graph model for scalable overlapping community detection. In *ECML PKDD*, 2019.

Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3): 033015, 2009.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, June 2014.

Wenzhe Li, Sungjin Ahn, and Max Welling. Scalable mcmc for mixed membership stochastic blockmodels. In *Artificial Intelligence and Statistics*, pages 723–731, 2016.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407, 2004.

Sam Patterson and Yee Whye Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

Robert Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, 1958.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. *arXiv preprint arXiv:1401.0118*, 2013.

Francisco R Ruiz, Michalis Titsias RC AUEB, and David Blei. The generalized reparameterization gradient. In *Advances in neural information processing systems*, pages 460–468, 2016.

Tim Salimans, David A Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979, 2014.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1170–1175. IEEE, 2012.