ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200282

Struct-MMSB: Mixed Membership Stochastic Blockmodels with Interpretable Structured Priors

Yue Zhang and **Arti Ramesh** [1]

Abstract. The mixed membership stochastic blockmodel (MMSB) is a popular framework for community detection and network generation. It learns a low-rank mixed membership representation for each node across communities by exploiting the underlying graph structure. MMSB assumes that the membership distributions of the nodes are independently drawn from a Dirichlet distribution, which limits its capability to model highly correlated graph structures that exist in real-world networks. In this paper, we present a flexible richly structured MMSB model, *Struct-MMSB*, that uses a recently developed statistical relational learning model, hinge-loss Markov random fields (HL-MRFs), as a structured prior to model complex dependencies among node attributes, multi-relational links, and their relationship with mixed-membership distributions. Our model is specified using a probabilistic programming templating language that uses weighted first-order logic rules, which enhances the model's interpretability. Further, our model is capable of learning latent characteristics in real-world networks via meaningful latent variables encoded as a complex combination of observed features and membership distributions. We present an expectation-maximization based inference algorithm that learns latent variables and parameters iteratively, a scalable stochastic variation of the inference algorithm, and a method to learn the weights of HL-MRF structured priors. We evaluate our model on six datasets across three different types of networks and corresponding modeling scenarios and demonstrate that our models are able to achieve an improvement of 15% on average in test log-likelihood and faster convergence when compared to state-of-the-art network models.

1 INTRODUCTION

Modeling the complex and intricate interactions existing within a community is an important network science problem that has gained attention in the last decade. Perhaps one of the most commonly and widely used network generation and community detection model is mixed membership stochastic blockmodel (MMSB) [1], owing to its flexibility in modeling different kinds of networks and communities. MMSB models the node's membership in latent groups using a mixed-membership distribution, wherein the node can be part of multiple latent groups. This membership is used to generate the network structure.

Related Work MMSB has been adapted in various different ways to model complex network structures, such as hierarchical community structures [19, 33, 18, 41], dynamic networks [10, 15, 35, 20], multi-relational graph structures [37, 23, 30, 7], nodes participating in multiple communities [43, 36, 36, 9], and feature-rich networks [42, 12, 38]. MMSB has also recently been extended to heterogenous networks [21]. Other probabilistic blockmodels have also been developed, such as infinite relational model (IRM) [22], Bi-LDA and Multi-LDA with extension to hierarchical Dirichlet processes (Multi-HDP) [31], binary space partitioning-tree process (BSP-tree) [13], a generalized version of the Mondrian process [32], and random function priors [29].

Recent advances in deep learning produced more powerful models for graph data, such as graph neural networks (GNNs) [39, 6, 25], GraphRNN [40], graph attention networks (GAT) [34]. Bojchevski et al. developed NetGAN [4], which uses the generative adversarial networks (GAN) framework to generate random walks on graphs and De Cao et al. [8] developed MolGAN, which generates molecular graphs using the combination of a GAN framework and a reinforcement learning objective. These models differ from MMSB and its variants in that they are not hand-engineered but can learn from data. But, the downside is the opaqueness and lack of interpretability of these models [39], which Struct-MMSB seeks to address.

The above-mentioned blockmodels with the exception of a notable few ones such as the recently developed Copula-MMSB [11] assume that the membership indicator pairs are drawn independently; this limits the capability of the model to encode complex dependencies in the network structure. Even when the existing approaches relax this independence assumption, their applicability is restricted to only specific network structures (such as a-MMSB models assortative structure in networks) and are not solely versatile enough to handle the various kinds of dependencies present in real-world data, such as presence of additional features, multiple relationships, provision for learning meaningful latent variables. Further, as the models progress toward capturing complex dependencies, they are specified using complex functions that have limited interpretability. For example, the Copula-MMSB model uses the Copula function [11], latent mixed-membership groups (LMMG) uses the logistic function as priors [24], and the BSP-tree uses a complex variant of the Mondrian process [13]. In addition to only capturing a specific kind of relationship, these priors are also harder to specify and understand for domain experts who may have a limited knowledge of machine learning. Incorporating correlations among the latent groups and memberships that can be easily interpreted provides the models with modeling power and possibility of application in domains that require careful specification of domain knowledge such as computational social science, bioinformatics, and other evolving application areas for societal good.

Contributions In this paper, we introduce a scalable and general purpose MMSB, *Struct-MMSB*, by enhancing MMSB with a structured prior using a recently developed graphical model, hinge-loss Markov random fields (HL-MRFs). Our approach, Struct-MMSB, is

[1] SUNY Binghamton, USA, email: {yzhan202, artir}@binghamton.edu

Table 1: A comparison of general-purpose stochastic blockmodel frameworks

Model	Dependencies	Meaningful Latent Variables	Features	Scalability
IRM [22]	×	×	×	×
MMSB [1]	×	×	×	✓
LMMG [24]	×	×	✓	✓
a-MMSB [17]	×	×	×	✓
Copula-MMSB [11]	✓	×	×	×
Struct-MMSB (our approach)	✓	✓	✓	✓

inspired from latent topic networks (LTN), a general-purpose latent Dirichlet allocation (LDA) using structured priors [14]. Table 1 gives a comparison of our model Struct-MMSB with other popular variants of MMSB. Our model possesses the capability to encode: 1) the relational dependencies among membership distributions, 2) additional node features, 3) multi-relational information, and their impact on the membership distributions using a probabilistic programming templating language, thus remaining interpretable and easy to specify custom network relationships. Further, our approach also incorporates the ability to encode meaningful latent variables, which can be learned as a complex combination of observed features, membership distributions, and group-group interaction probabilities, thus catering to many latent-variable modeling scenarios in computational social science problems.

Next, we present algorithms for inference and learning in Struct-MMSB. We formulate an expectation maximization (EM) inference for inferring the expected value of latent variables, mixed-membership distributions of nodes in groups, and the group-group interaction probabilities. Then, we develop a scalable inference method using stochastic EM and show how to effectively incorporate relational dependencies, while remaining scalable to large networks. We then present a way to learn the weights of the first-order logical rules by maximizing the likelihood of the weights without additional ground truth data, thus allowing the model to learn the predictive ability of the logical rules in the data.

We demonstrate the versatility of our model in modeling different network modeling scenarios on data from six real-world networks and show that our model on average achieves a 15% better log-likelihood and a better prediction performance than the state-of-the-art MMSB variant, Copula-MMSB [11], IRM [22], and MMSB [1] and their multi-relational and online variants.

2 BACKGROUND

In this section, we provide background on MMSB and HL-MRFs, and then proceed to show how to incorporate HL-MRFs as a structured prior in MMSB in Struct-MMSB. For ease of reference, we present a list of notations and their meanings to be used in the equations and derivations in this paper in Table 2.

2.1 Mixed Membership Stochastic Blockmodels (MMSB)

The generative process for MMSB is defined as follows.

- For each node $p \in N$:
  - Draw a K dimensional mixed membership vector $\vec{\pi}_p \sim$ Dirichlet $(\vec{\alpha})$
- For each pair of nodes $(p, q) \in N \times N$
  - Draw membership indicator for the initiator, $\vec{z}_{p\to q} \sim$ Multinomial $(\vec{\pi}_p)$
  - Draw membership indicator for the receiver, $\vec{z}_{q\to p} \sim$ Multinomial $(\vec{\pi}_q)$

Table 2: Notations for STRUCT-MMSB

N	number of nodes
p, q	specific nodes
K	number of communities
k, k_1, k_2	specific communities
$\alpha, \beta^{(1)}, \beta^{(2)}$	hyperparameters of MMSB
$Y_{p,q}$	link between nodes p and q
$z_{p\to q}$	membership indicator for sender node p and receiver node q
Π	membership distributions of all the nodes
π_p	membership distribution of node p across communities
$\pi_k^{(p)}$	value in the membership distribution of node p for group k
B	matrix capturing interactions between communities
B_{k_1,k_2}	interaction between communities k_1 and k_2
Λ, λ	HL-MRF rule weights
ψ_r	HL-MRF potential function for rule r
$M(1)$	number of HL-MRF potentials for Π
$M(2)$	number of HL-MRF potentials for B
$H^{(1)}, H^{(2)}$	latent variables in the HL-MRF prior
X	observed features
η	Lagrange coefficient

  - Sample the value of their interaction, $Y_{p,q} \sim$ Bernoulli$(\vec{z}_{p\to q}^{\top} B \vec{z}_{q\to p})$, where $B \sim Beta(\vec{\beta^{(1)}}, \vec{\beta^{(2)}})$

The independence assumptions implicit in the Dirichlet and Beta priors are what prevents MMSB from capturing complex dependencies. The priors are therefore our point of attack in developing a rich, flexible, and easy-to-encode stochastic blockmodel. In Struct-MMSB, we replace the flat Dirichlet and Beta priors with more expressive HL-MRF priors, we discuss HL-MRFs and their suitability as priors in MMSB below.

2.2 Hinge-loss Markov Random Fields

HL-MRFs are a recently developed scalable class of continuous, conditional graphical models [3]. HL-MRFs can be specified using *Probabilistic Soft Logic (PSL)* [3], a first-order logic templating language. In PSL, random variables are represented as logical atoms and weighted rules define dependencies between them of the form: $\lambda : P(a) \wedge Q(a, b) \rightarrow R(b)$, where *P*, *Q*, and *R* are predicates, *a* and *b* are variables, and λ is the weight associated with the rule. The weight of the rule r indicates its importance in the HL-MRF model, which is defined as

$$P(Y|X) = \exp\Big(-\sum_{r=1}^{M} \lambda_r \psi_r(Y, \mathbf{X})\Big)/Z(\lambda); \quad (1)$$

$$Z(\lambda) = \int_Y \exp\Big(-\sum_{r=1}^{M} \lambda_r \psi_r(Y, X)\Big)$$

$$\psi_r(Y, X) = (\max\{l_r(\mathbf{Y}, X), 0\})^{\rho_r} \quad (2)$$

where $P(Y|X)$ is the probability density function of a subset of logical atoms Y given observed logical atoms X, $\psi_r(Y, X)$ is a *hinge-loss potential* corresponding to an instantiation of a rule r, and is specified by a linear function l_r and optional exponent $\rho_r \in \{1, 2\}$, and $Z(\lambda)$ is the partition function. The logical conjunction of Boolean variables $X \wedge Y$ can be generalized to continuous variables using the hinge function $max\{X + Y - 1, 0\}$, which is known as the Lukasiewicz t-norm. HL-MRFs admit tractable MAP inference regardless of the graph structure of the graphical model, making it feasible to reason over complex dependencies. This is possible because HL-MRFs operate on continuous random variables and encode dependencies using convex potential functions, so MAP inference in these models is a convex optimization problem. Giannini et al. [16] provide theoretical insight for understanding the convex characterization of the Lukasiewicz t-norms and Horn clauses in PSL [3].

Our resulting model, Struct-MMSB, upon replacing the priors with HL-MRFs possess the following desirable qualities. They are structured and can capture complex relational dependencies among the nodes, their features, their membership distributions and group-group interactions using graphical model templates. These templates are simple weighted rules that capture the general characteristics of the network. The lucid nature of these rules make them inherently interpretable and intuitive and hence pave the way for ease of specification by domain experts. Further, these priors can be enriched using meaningful latent variables that further enhance their modeling power and intuitiveness.

3 STRUCT-MMSB

Our goal in designing Struct-MMSB is to create a lucid, easy-to-specify, and expressive generative model. We discuss the technical details of Struct-MMSB below.

3.1 Struct-MMSB graphical model

To construct Struct-MMSB, we replace the Dirichlet priors and Beta priors in the MMSB generative process with HL-MRF potential function ψ. Figure 1 shows the plate diagram of Struct-MMSB. The HL-MRF priors are indicated by ψ_π, which captures dependencies between the membership distribution π of two nodes in the graph and ψ_B, which captures dependencies between groups in the group interaction matrix B. Before delving into the mathematical details,

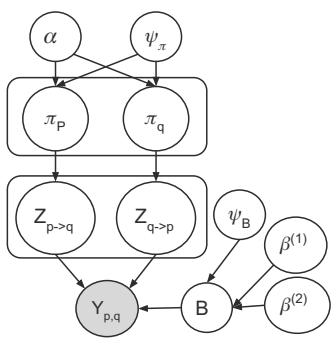


Figure 1: Plate Diagram of Struct-MMSB

we provide some example dependencies that can be encoded using Struct-MMSB:

Correlated nodes: correlated(p, q) $\wedge$ $\pi(p, K) \rightarrow \pi(q, K)$, if two nodes p and q are correlated, then their corresponding membership distributions (π) are correlated.

Latent characteristics modeled as latent variables: link(p, r) $\wedge$ link(q, r) $\rightarrow$ similar(p, q), if two nodes p and q share common neighbor r, then they are similar.

Inter-group and Intra-group links: correlated(p, q) $\wedge$ $\pi(p, K_1) \wedge$ B(K_1, K_2) $\rightarrow \pi(q, K_2)$, if two nodes p and q are correlated and the group interaction between K_1 and K_2 is high, then the value of the membership distribution π(p, K_1) is similar to π(q, K_2).

3.2 Combining MMSB with HL-MRF priors

In Struct-MMSB, the priors given by HL-MRFs define probability densities as follows:

$$P(\Pi, H^{(1)}|X^{(1)}) \propto exp\Big(- \sum_{j=1}^{M^{(1)}} \lambda_j^{(1)} \psi_j^{(1)}(\Pi, X^{(1)}, H^{(1)}) \Big)$$

$$P(B, H^{(2)}|X^{(2)}) \propto exp\Big(- \sum_{j=1}^{M^{(2)}} \lambda_j^{(2)} \psi_j^{(2)}(B, X^{(2)}, H^{(2)}) \Big)$$

where $\psi_j^{(a)}$ are hinge-loss potentials, $M^{(a)}$ are number of hinge-loss potentials, and $X^{(a)}$ are observed features that are conditioned on, and $H^{(a)}$ are additional latent variables introduced by Struct-MMSB, and Π and B are MMSB parameters.

The membership distributions Π are constrained to sum to 1 for each node as in MMSB. For membership distributions that do not come from the HL-MRF priors, the default Dirichlet priors are used. We also add smoothed Dirichlet and Beta potentials for Π, B, and $1 - B$, $\prod_{p,k} \pi_k^{p^{\alpha-1}}$, $\prod_{k_1,k_2} B_{k_1,k_2}^{\beta^{(1)}-1}$, and $\prod_{k_1,k_2}(1 - B_{k_1,k_2})^{\beta^{(2)}-1}$, respectively. Incorporating the HL-MRF structured priors together with the smoothed Dirichlet and Beta potentials, the log-posterior of the variables of interest (parameters Π and B, latent variables $H^{(1)}$ and $H^{(1)}$) is,

$$\begin{aligned}
&logP(\Pi, B, H^{(1)}, H^{(2)}|Y, \alpha, \beta^{(1)}, \beta^{(2)}, X^{(1)}, X^{(2)}, \Lambda) = \\
&\quad \sum_{p \in N, q \in N} log\Big(\sum_{z_p \in K, z_q \in K} P(Y_{p,q}, z_p, z_q|\pi_p, \pi_q, B) \Big) \\
&\quad + \sum_{p \in N, k \in K} (\alpha - 1) log \pi_k^p \\
&\quad - \sum_{j=1}^{M^{(1)}} \lambda_j^{(1)} \psi_j^{(1)}(\Pi, X^{(1)}, H^{(1)}) \\
&\quad + \sum_{k_1,k_2} \Big((\beta^{(1)} - 1) log B_{k_1,k_2} + (\beta^{(2)} - 1) log(1 - B_{k_1,k_2}) \Big) \\
&\quad - \sum_{j=1}^{M^{(2)}} \lambda_j^{(2)} \psi_j^{(2)}(B, X^{(2)}, H^{(2)}) + const. \qquad (3)
\end{aligned}$$

3.3 Training via EM

Our EM algorithm is motivated from latent variable learning in HL-MRFs [2] and LTN [14]. We train the model by maximum a posteriori (MAP) estimation, optimizing Eqn 3 with respect to Π, B, $H^{(1)}$, $H^{(2)}$. Since Eqn 3 cannot be optimized directly due to the sum inside the logarithm, we develop an EM algorithm that iteratively optimizes a lower bound arising from Jensen's inequality.

Applying Jensen's inequality, we get the objective function to be, $R(\Pi, B, X^{(1)}, X^{(2)}, H^{(1)}, H^{(2)}) \leq$

$logP(\Pi, B, H|Y, \alpha, \beta^{(1)}, \beta^{(2)}, X, \Lambda)$, where,

$$R(\Pi, B, X^{(1)}, X^{(2)}, H^{(1)}, H^{(2)}) = -\sum_{j=1}^{M^{(1)}} \lambda_j^{(1)} \psi_j^{(1)}(\Pi, X^{(1)}, H^{(1)})$$
$$- \sum_{j=1}^{M^{(2)}} \lambda_j^{(2)} \psi_j^{(2)}(B, X^{(2)}, H^{(2)})$$
$$+ \sum_{p,k_1} \left(\sum_{q,k_2} (\gamma_{p,q,k_1,k_2} + \gamma_{q,p,k_2,k_1}) + \alpha - 1 \right) log \pi_{k_1}^p$$
$$+ \sum_{k_1,k_2} \left(\sum_{p,q} \gamma_{p,q,k_1,k_2} Y_{p,q} + \beta^{(1)} - 1 \right) log B_{k_1,k_2}$$
$$+ \sum_{k_1,k_2} \left(\sum_{p,q} \gamma_{p,q,k_1,k_2} (1 - Y_{p,q}) + \beta^{(2)} - 1 \right) log(1 - B_{k_1,k_2})$$
$$- \sum_{p,q,k_1,k_2} \gamma_{p,q,k_1,k_2} log \gamma_{p,q,k_1,k_2} + const \quad (4)$$

We define $\gamma_{p,q,k_1,k_2} \triangleq P(\vec{z}_{p\to q} = k_1, \vec{z}_{q\to p} = k_2 | \Pi^{(t)}, B^{(t)}, Y_{p,q})$, which encodes the distribution over the membership indicators $\vec{z}_{p\to q}$ and $\vec{z}_{q\to p}$ based on the previous parameter values of Π and B. The algorithm consists of an E-step and an M-step, which are iterated until convergence. We first present the batch version of the algorithm, that iterates on all the data points for each E and M-step update and then present a stochastic variation that is scalable to large datasets.

3.3.1 E-step

To perform the E-step of the EM algorithm at iteration t, we first compute γ_{p,q,k_1,k_2} in Equation 4.

$$\gamma_{p,q,k_1,k_2} \propto P(Y_{p,q}|z_p, z_q, \Pi^{(t)}, B^{(t)}) P(z_p, z_q | \Pi^{(t)}, B^{(t)})$$
$$= B_{k_1,k_2}^{Y_{p,q}} (1 - B_{k_1,k_2})^{1 - Y_{p,q}} \pi_{k_1}^{(p)} \pi_{k_2}^{(q)}$$

Since the inference algorithm is iterative, the superscript (t) is used to refer to the previous/initialized values of Π and B.

3.3.2 M-step

We then perform the maximization step for Π and B parameters that are not involved in the HL-MRF prior, for which the update is identical to the M-step of the EM algorithm for standard MMSB. For Π, we add Lagrange terms $-\sum_p \eta_k^{\Pi} (\sum_k \pi_k^p - 1)$ to constrain the parameter vectors to sum to one, where η_k^{Π} is the Lagrange coefficient. To derive updates for B, we define B and $1 - B$ as $B^{(i)}, i \in \{0, 1\}$, respectively, and add the Lagrange term $-\sum_{k_1 k_2} \eta_{k_1 k_2}^B (\sum_i B_{k_1 k_2}^{(i)} - 1)$ to constrain the $B_{k_1 k_2}^{(i)}$ to sum to one, where $\eta_{k_1 k_2}^B$ is the Lagrange coefficient. Taking derivatives and setting them to zero, we obtain the updates,

$$\pi_{k_1}^p \propto \sum_{q,k_2} (\gamma_{p,q,k_1,k_2} + \gamma_{q,p,k_2,k_1}) + \alpha - 1;$$
$$B_{k_1,k_2}^{(0)} = \frac{\sum_{p,q} \gamma_{p,q,k_1,k_2} Y_{p,q} + \beta^{(1)} - 1}{\sum_{p,q} \gamma_{p,q,k_1,k_2} + \beta^{(1)} + \beta^{(2)} - 2}$$

Then, we optimize the lower bound jointly over the remaining Π and B parameters by fixing the parameters above that are not updated using HL-MRF priors. The hinge-loss potentials for Π and B can be optimized separately as evident from Eqn 4. We minimize the negative of each of these two subproblems $-R(\Pi, H^{(1)}, X^{(1)})$, $-R(B, H^{(2)}, X^{(2)})$ using a consensus-optimization algorithm based on the alternating direction method of multipliers (ADMM).

To accomplish this, we extend the consensus optimization algorithm by Bach et al. [3] for MAP inference in HL-MRFs. The algorithm creates a local copy for each of the variables, thus dividing the original problem into independent subproblems that may be solved in parallel. The algorithm iteratively solves the independent subproblems and then updates the global consensus variables to be the average of the local copies. This is guaranteed to find the global optimum solution of the objective function. For more details, we refer the reader to [3] and [5].

We extend this algorithm to include the objective function terms in the lower bound R in Eqn 4 for Π and B. For Π, the terms 1 and 3 in Eqn 4 are included and for B, the terms 2, 4, and 5 in Eqn 4 are included. The entropy term (term 6, $-\sum_{p,q,k_1,k_2} \gamma_{p,q,k_1,k_2} log \gamma_{p,q,k_1,k_2}$) and the constant term (term 7) are not included in the M-step objective as they do not have Π or B in them. To cast this as a consensus optimization problem, we create local copies of our variables of interest (Π and B, denoted by C_i in the ADMM update equation below), which we refer to as c_i, where i refers to each π_k^p and B_{k_1,k_2}, in the respective consensus optimization equations. The consensus optimization problem entails solving the subproblems with local copies c_i independently for both the variables and then iterating till a consensus is reached, which is enforced via an equality constraint in the Lagrange term with coefficient η_i. For convenience, we refer to our objective terms as $A_i log c_i$, where A_i denotes the coefficient in term 3 in Equation 4, which equals to $(\sum_{q,k_2} (\gamma_{p,q,k_1,k_2} + \gamma_{q,p,k_2,k_1}) + \alpha - 1)$, and c_i denotes the local copy of the variable.

The consensus ADMM update [5] for c_i is given by,

$$c_i = \arg\min_{c_i'} (-A_i log c_i' + \eta_i (c_i' - C_i) + \frac{\rho}{2} (c_i' - C_i)^2)$$

where ρ is an ADMM step-size parameter, c_i is the local copy of the variable used by the ADMM consensus optimization algorithm, and C_i refers to the original variable. In order to solve the ADMM update equation, we take its derivative with respect to c_i and set it equal to zero, $\frac{\partial J}{\partial c_i'} = \rho c_i'^2 + c_i' (\eta_i - \rho C_i) - A_i = 0$.

Solving this quadratic equation, we get two solutions. One of them is negative and can be discarded; c_i is set to the positive solution. The Lagrange coefficients are updated using ADMM updates for Lagrangian dual, $\eta_i \leftarrow \eta_i + \rho(c_i' - C_i)$.

3.4 Stochastic EM

To scale the batch model to large networks, we present a stochastic EM inference method. Our algorithm is motivated from the online EM algorithm [28] and the stochastic optimization for a-MMSB [17]. We compute the expected sufficient statistics of Π and B and update it using randomly sampled node-pair (p, q) from distribution $g(p, q)$ instead of the entire sample, by computing the stochastic natural gradient in the space of sufficient statistics.

To obtain the stochastic EM algorithm, we first subsample the graph and compute the expected sufficient statistics for the sample, given the current settings of the Π and B. We then update Π and B using these sufficient statistics. We introduce three global variables: θ_k^p, $\phi_{k_1 k_2}$, and $\phi'_{k_1 k_2}$ to derive the stochastic EM updates. The membership distribution π_k^p is equal to the normalized θ_k^p. The affinity blockmodel $B_{k_1 k_2}^{(0)}$ (which refers to B) is given by $\phi_{k_1 k_2}/(\phi_{k_1 k_2} + \phi'_{k_1 k_2})$, and $B_{k_1 k_2}^{(1)}$ (which refers to $1 - B$) equals $\phi'_{k_1 k_2}/(\phi_{k_1 k_2} + \phi'_{k_1 k_2})$.

To derive the expected sufficient statistics of π^p_k for a specific node p and community k_1, we have,

$$\Big(\sum_{q,k_2}(\gamma_{p,q,k_1,k_2}+\gamma_{q,p,k_2,k_1})+\alpha-1\Big)log\pi^p_{k_1}$$
$$=\Big(\sum_{q,k_2}g(p,q)\frac{1}{g(p,q)}(\gamma_{p,q,k_1,k_2}+\gamma_{q,p,k_2,k_1})+\alpha-1\Big)log\pi^p_{k_1}$$
$$=\Big(E_g[\frac{1}{g(p,q)}(\gamma_{p,q,k_1,k_2}+\gamma_{q,p,k_2,k_1})]+\alpha-1\Big)log\pi^p_{k_1}$$

The intuition behind the above expected sufficient statistics comes from online EM for unsupervised models [28] and stochastic variational EM for a-MMSB [17]. Here, we sample a single node pair (p, q) and compute the γ if our entire graph consisted of p and q, repeated $1/g(p,q)$ times. In practice, instead of a single node a mini-batch is sampled. Liang et al. [28] specify that updating on multiple data instances using a mini-batch adds more stability than updating after each sample. This translates to all q in the mini-batch for a specific node of interest p.

Hence, under each mini-batch, without HL-MRF priors, the expected sufficient statistics of $\theta^p_{k_1}$, $s_{p,k_1}=\frac{1}{g}\sum_{q,k2}(\gamma_{p,q,k_1,k_2}+\gamma_{q,p,k_2,k_1})+\alpha-1$. With HL-MRF priors, we first use ADMM to find local optimal value $\pi^{*p}_{k_1}$ under the current mini-batch, then the expected sufficient statistics is calculated as $s^{PSL}_{p,k_1}=\pi^{*p}_{k_1}\sum_{k_1}s_{p,k_1}$. Similarly, we calculate the expected sufficient statistics for B. As B_{k_1,k_2} and $1-B_{k_1,k_2}$ factorize into separate terms in Eqn 4, we treat them as two different variables and calculate the expected sufficient statistics separately. For B_{k_1,k_2}, we have,

$$\sum_{p,q}\gamma_{p,q,k_1,k_2}Y_{p,q}logB_{k_1,k_2}$$
$$=\sum_{p,q}g(p,q)\Big(\frac{1}{g(p,q)}\gamma_{p,q,k_1,k_2}Y_{p,q}\Big)logB_{k_1,k_2}$$
$$=E_g[\frac{1}{g(p,q)}\gamma_{p,q,k_1,k_2}Y_{p,q}]logB_{k_1,k_2}$$

Under mini-batch, without HL-MRF priors, expected sufficient statistics of ϕ_{k_1,k_2} equals to, $s_{k_1k_2}=\frac{1}{g}\sum_{p,q}\gamma_{pqk_1k_2}Y_{p,q}+\beta^{(1)}-1$. Similarly, for $B'_{k_1,k_2}=1-B_{k_1,k_2}$, we have,

$$\sum_{p,q}\gamma_{p,q,k_1,k_2}(1-Y_{p,q})log(B'_{k_1,k_2})$$
$$=\sum_{p,q}g(p,q)\Big(\frac{1}{g(p,q)}\gamma_{p,q,k_1,k_2}(1-Y_{p,q})\Big)log(B'_{k_1,k_2})$$
$$=E_g[\frac{1}{g(p,q)}\gamma_{p,q,k_1,k_2}(1-Y_{p,q})]log(B'_{k_1,k_2})$$

Under mini-batch, without HL-MRF priors, expected sufficient statistics of ϕ'_{k_1,k_2} equals to $s'_{k_1k_2}=\frac{1}{g}\sum_{p,q}\gamma_{p,q,k_1,k_2}(1-Y_{p,q})+\beta^{(2)}-1$. With HL-MRF priors, to compute the expected sufficient statistics for $\phi_{k_1k_2}$ and $\phi'_{k_1k_2}$, first we need to use ADMM to find local optimal value $B^*_{k_1k_2}$ under current mini-batch. Then, the expected sufficient statistics for $\phi_{k_1k_2}$ becomes $s^{PSL}_{k_1k_2}=B^*_{k_1k_2}(s_{k_1k_2}+s'_{k_1k_2})$ and for $\phi'_{k_1k_2}$, $s'^{PSL}_{k_1k_2}=(1-B^*_{k_1k_2})(s_{k_1k_2}+s'_{k_1k_2})$. Intuitively speaking, the expected sufficient statistics with HL-MRF priors re-assign the total mass without HL-MRF priors, $\sum s$, and make it proportional to the optimal value using the estimated $\pi^{*p}_{k_1}$ and $B^*_{k_1k_2}$ values from ADMM.

In our experiments, we use mini-batch instead of just one node pair. Evaluating the rewritten above equations for a node pair sampled from the graph gives a noisy but unbiased estimate of the batch model. We scale the expected sufficient statistics estimated from the subsample by $\frac{1}{g(p,q)}$ so that they are unbiased estimates of the true sufficient statistics as illustrated by Gopalan et al. [17]. For instance, if we sample nodes uniformly at random, then $g(p,q)=\text{miniBatch}/N^2$.

We calculate this as the difference between expected sufficient statistics and previous value as follows

$$\begin{aligned}\partial\theta^t_{pk}&=s^{PSL}_{pk}-\theta^{t-1}_{pk}\\ \partial\phi^t_{k_1k_2}&=s^{PSL}_{k_1k_2}-\phi^{t-1}_{k_1k_2}\\ \partial\phi'^t_{k_1k_2}&=s'^{PSL}_{k_1k_2}-\phi'^{t-1}_{k_1k_2}\end{aligned}\tag{5}$$

In the next step, we calculate the stochastic natural gradients computed from a sampled node pair (p,q) or mini-batch and update the global variables,

$$\begin{aligned}\theta^p_k&\leftarrow\theta^p_k+\rho_t*\partial\theta^t_{pk}\\ \phi_{k_1k_2}&\leftarrow\phi_{k_1k_2}+\rho_t*\partial\phi^t_{k_1k_2}\\ \phi'_{k_1k_2}&\leftarrow\phi'_{k_1k_2}+\rho_t*\partial\phi'^t_{k_1k_2}\end{aligned}\tag{6}$$

where ρ is global step size. Results from the stochastic approximation literature requires that $\sum_t\rho^2_t<\infty$ and $\sum_t\rho_t=\infty$ to guarantee convergence to a local optimum. We set $\rho_t\triangleq(\tau_0+t)^{-\kappa}$, where $\kappa\in(0.5,1]$ is the learning rate and $\tau_0\geq 0$ downweights early iterations.

Distributed Implementation We implement a distributed computing environment using Apache Thrift framework for the stochastic EM inference. This further helps in scaling our models to larger datasets. We perform stochastic EM inference at each client side using subsampled mini-batch and then update the global parameters at server side.

3.5 Weight learning

The weights of the HL-MRF first-order-logic rules, Λ, can be selected based on domain knowledge. The challenge to learning weights arises because often there is no ground truth data for our variables of interest, Π and B. Following Bach et al. [3], we present a maximum-likelihood estimation by maximizing the log-likelihood of the training data given parameter values to learn the weights.

$$\frac{\partial R}{\partial\Lambda_q}=\frac{\partial logP(\Pi,H|X)}{\partial\Lambda_q}=E_\Lambda[\Psi_q(X,H,\Pi)]-\Psi_q(X,H,\Pi)$$
$$E_\Lambda[\Psi_q(X,H,\Pi)]=\int_\Pi\Psi_q(X,H,\Pi)P(\Pi|X,H)$$

We approximate the expectation using the value of the potential functions at the most probable setting of Π with the current parameters. During weight learning, we treat latent variables H as observations using the inferred value from the EM step.

4 EXPERIMENTS

We conduct experiments to answer the questions:

1) *How our models perform on different network modeling scenarios?* To demonstrate the versatility of Struct-MMSB and its ability to model different real-world scenarios, we evaluate the performance of our model on a total of 6 datasets and three network modeling scenarios: a) presence of additional features (*case 1*), b) presence

of multiple links (*case 2*), and c) neighborhood similarity (*case 3*), [27]. We induce three different structured priors that correspond to the characteristics of each of these modeling scenarios. We compare the training and test log-likelihood and area under the ROC curve (AUC-ROC) of our models at convergence with MMSB [1] or its online/stochastic variant, IRM [22], and the state-of-the-art variant of MMSB (Copula-MMSB [11]), which has been compared to some other variants of MMSB and network models (MMSB, IRM, LFRM [30], and iMMM [26]) and shown to be better. Since Copula-MMSB in the present form is not capable of handling multi-relational data, we extend Copula-MMSB to handle multi-relational data wherever appropriate; we refer to this model as Multi-Copula. We show that our model achieves better performance and faster convergence both under batch and stochastic inference when compared to the above-mentioned models.

2) *How informative are the latent variables in our intuitive HL-MRF priors?* While performance is one important aspect, the more interesting contribution of our approach is the interpretable and intuitive nature of Struct-MMSB models, succinctly capturing domain knowledge in the different modeling scenarios. This interpretable nature is further enhanced by the presence of meaningful latent variables that can abstract the relationship between the features, nodes, and network connections. We show how to encode meaningful latent variables in all the three modeling scenarios: presence of additional features, presence of multiple links, and neighborhood similarity. We present qualitative analysis of the values learned by our latent variables and show that our models are able to encode and learn meaningful latent variable values that enhance the modeling power and interpretability of our model.

4.1 Experimental settings

In all our experiments, our models use same initializations as standard MMSB. In the stochastic inference setting, our models also use the same mini-batches as online MMSB, so that they are exactly comparable. We initialize the hyper-parameters $\beta^{(1)}$ and $\beta^{(2)}$ of parameter B by setting $\beta^{(1)}/(\beta^{(1)}+\beta^{(2)}) = \textit{link-rate}$, where $\textit{link-rate} = link/(link + non\text{-}link)$. We set number of communities to 5 for small datasets (feature-based similarity (case 1) and multi-relational experiments (case 2)), and to 8 for the large datasets in link-based similarity (case 3) experiment. And, we randomly initialize the Π and B values. Statistically significant values with a rejection threshold of p = 0.05 are typed in **bold**.

4.2 Case 1: feature-based similarity

Here, we design models that take additional node features into account to guide community discovery and network generation.

4.2.1 Model

In this modeling scenario, we specify HL-MRF priors to utilize feature commonality between nodes to model their community membership. The priors in Table 3 capture that if two nodes have many common features, then there is a greater chance of these nodes being grouped together. The first rule in Table 3 means if node p and node q have the same features (multiple instantiations of Rule 1), then we infer that node p and node q are similar. *similarity(p,q)* is a latent variable that helps us model the degree of similarity between two nodes. The second rule captures that if two nodes are similar, then we can infer that they have similar membership distributions.

Table 3: HL-MRF priors to guide community discovery based on presence of multiple common features measured using latent variable *similarity(p, q)*

Feature-Based Similarity Model
feature(p, T) $\wedge$ feature(q, T) $\wedge$ link(p, q) $\rightarrow$ similarity(p, q) similarity(p, q) $\wedge$ π(p, K) $\rightarrow$ π(q, K) similarity(p, q) $\wedge$ $\neg\pi$(p, K) $\rightarrow$ $\neg\pi$(q, K)

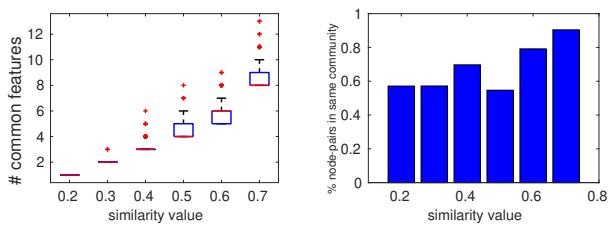
4.2.2 Results

We evaluate this model on two Facebook Ego datasets: 1) Ego-414 dataset containing 159 nodes and 3386 links, and 2) Facebook Ego-686 dataset containing 170 nodes and 3312 links. Table 4 shows that our model achieves a better training and test log-likelihood and AUC on two Facebook ego datasets when compared with standard MMSB, Copula-MMSB, and IRM.

Our latent variables, apart from providing the model with modeling power, also bring interpretability to the model. Figures 2(a) and 2(b) illustrate the correlation between latent variable *similarity* and the number of common features and membership distributions. This conforms with our structured prior that a commonality in features between a pair of nodes can indicate that they belong to the same communities. We also observe that our latent variable values act as a proxy to the relationship between similarity in features and their membership distributions and helps us interpret them better. For example, in Ego-414 dataset, we get the value for the latent variable *similarity* for nodes 74 and 88 to be 0.714 and we observe that they both have similar membership distributions after training, having a high value for the same community.

Table 4: Results on Facebook Ego-414 and Ego-686 datasets comparing Struct-MMSB with Copula-MMSB, Standard MMSB, and IRM.

Dataset	Model	Log-Likelihood	Test Log-Likelihood	AUC
Ego414	Struct-MMSB	**-1042.986**	**-787.406**	**0.954**
	Copula-MMSB	-1909.378	-1404.075	0.844
	MMSB	-1309.271	-986.321	0.9220
	IRM	-2205.484	-1521.684	0.746
Ego686	Struct-MMSB	**-1628.858**	**-1169.224**	**0.892**
	Copula-MMSB	-2002.216	-1407.204	0.854
	MMSB	-1690.593	-1221.328	0.875
	IRM	-2840.591	-1866.839	0.659



(a) Correlation between similarity value and number of common features (b) Correlation between similarity value and community membership distributions

Figure 2: Correlation between latent variable values and membership distributions

4.3 Case 2: multi-relational graphs

In our next experiment, we consider a multi-relational graph setting, where there are multiple different relationships between a single pair of nodes. Here, we experiment on two different kinds of multi-relational data: i) the presence of multiple different types of relationships between a pair of nodes, and ii) the presence of the same type of relationship between a pair of nodes at different timestamps, each timestamp capturing a single relationship.

4.3.1 Model

To guide the generation of the model in the multi-relational setting, our structured HL-MRF prior captures that if a pair of two nodes p and q have multiple links between them (Rule 1 in Table 5), then, they are "closer" than other pairs of nodes. This closeness in their relationship is modeled using the latent variable *close(p, q)*. Rule 2 in Table 5 captures that if two nodes are closer (i.e., have a higher value of latent variable *close(p, q)*), then there is a higher probability of a link existing between them in the network. Note that in Rule 2, we connect the blockmodel B that captures the interaction between communities, membership distributions of nodes p and q with the latent variable *close* induced by the structured prior to guide the generation process.

Table 5: Capturing proximity between nodes based on the presence of multiple links between them using latent variable *close(p, q)*

Node Proximity in Multi-Relational Graphs
link(p, q, T) $\rightarrow$ close(p, q)
close(p, q) $\wedge$ B(K1,K2) $\wedge$ π(p, K1) $\rightarrow$ π(q, K2)

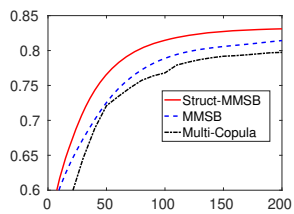
4.3.2 Results

We evaluate the performance of our model on two datasets: 1) Unified Medical Language System dataset (UMLS) consisting of 135 nodes, 49 relations, and a total of 6752 links, and 2) Email-Temporal dataset consisting of 142 nodes, 525 timestamps, and a total of 48,141 links. Table 6 shows the comparison of the performance scores of our model with standard MMSB and Multi-Copula (Copula extended to the multi-relational case to enable a fair comparison) at training and testing time on UMLS and Email-Temporal datasets, where our model achieves better log-likelihood scores on training and test.

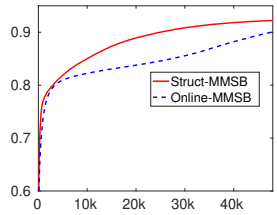
Also, as the priors are expected to guide the model in the right direction in the beginning, a better measure of the effectiveness of the structured priors is evaluating the progression toward convergence. Figure 3(a) shows the progression of the models toward convergence. We observe that Struct-MMSB progresses faster toward convergence and gets a higher AUC value right from the early iterations.

Table 6: Comparing Struct-MMSB with Standard and Multi-Copula MMSB on UMLS and Email-Temporal datasets.

Dataset	Model	Log-Likelihood	Test Log-Likelihood	AUC
UMLS	Struct-MMSB	**-11868.051**	**-2945.953**	**0.831**
	Multi-Copula	-12242.841	-3153.354	0.785
	MMSB	-12061.241	-2998.037	0.814
Email	Struct-MMSB	**-1523.488**	**-175.871**	**0.997**
	Multi-Copula	-1777.112	-210.090	0.995
	MMSB	-1562.353	-180.620	0.995



(a) Progression toward convergence of batch Struct-MMSB in UMLS dataset



(b) Progression toward convergence of stochastic Struct-MMSB in Facebook Ego107

Figure 3: Struct-MMSB progresses faster toward convergence in both batch and stochastic scenarios

4.4 Case 3: community discovery using neighborhood similarity

Here, we consider the problem of discovering communities in graphs using neighborhood similarity. Note that this is the setting for which MMSB was originally conceived. In this experiment, we evaluate the performance of the stochastic Struct-MMSB with the online variant of MMSB on performance and the ability to converge faster.

4.4.1 Model

The mixed membership π is a low rank representation of node's neighborhood. We guide π by capturing that if two nodes have multiple common neighbors (Rule 1 in Table 7), then they have similar membership distributions (Rules 2 and 3 in Table 7). We capture the neighborhood similarity using the latent variable *similarity(p, q)*.

Table 7: HL-MRF priors capturing neighborhood similarity between nodes based on the presence of multiple neighbors using latent variable *similarity(p, q)*

Community Discovery using Neighborhood Similarity
link(p, r) $\wedge$ link(q, r) $\rightarrow$ similarity(p, q)
similarity(p, q) $\wedge$ π(p, K) $\rightarrow$ π(q, K)
similarity(p, q) $\wedge$ $\neg\pi$(p, K) $\rightarrow$ $\neg\pi$(q, K)

Table 8: Results on Ego-107 and Email-Eu-Core datasets comparing stochastic Struct-MMSB with online MMSB

Dataset	Model	Log-Likelihood	Test Log-Likelihood	AUC
Ego107	Struct-MMSB	**-45202.096**	**-11441.823**	**0.922**
	Online-MMSB	-48534.366	-12262.921	0.900
Email	Struct-MMSB	**-31425.953**	**-8059.303**	**0.862**
	Online-MMSB	-31562.378	-8133.698	0.861

4.4.2 Results

We evaluate performance on the Facebook Ego-107 dataset, which contains 1045 nodes and 53498 links and Email-Eu-core dataset, which contains 1005 nodes and 25571 links. We implement the stochastic optimization for both Struct-MMSB and standard MMSB and compare our stochastic Struct-MMSB with stochastic/online MMSB. For each mini-batch iteration, we randomly sample 10% of the nodes from all the data. We add all links that exist between

the sampled nodes to the mini-batch. In total, we run 48,000 mini-batch iterations. Our stochastic model achieves better log-likelihood at training and test time when compared to online-MMSB as evident from Table 8. We also observe a similar convergence trend in the stochastic model as well (Figure 3(b)), where our model progresses faster toward convergence even while only observing a small subset of all the nodes and relationships during each iteration.

5 Conclusion

We presented a versatile general-purpose MMSB, *Struct-MMSB*, that is capable of encoding multi-relational data, additional features, and dependencies among the membership distributions, community interaction matrix, and learn meaningful latent variables in the process. We present a batch inference algorithm using EM and a scalable variant using stochastic EM and a method to learn the weights of the structured HL-MRF priors. Our experimental evaluation demonstrates the ability of our model to achieve a superior log-likelihood on training and held-out test data and faster convergence on three different modeling scenarios across 6 datasets and the interpretable nature of our learned latent variables.

REFERENCES

[1] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing, 'Mixed membership stochastic blockmodels', *JMLR*, **9**(Sep), 1981–2014, (2008).

[2] Stephen Bach, Bert Huang, Jordan Boyd-Graber, and Lise Getoor, 'Paired-dual learning for fast training of latent variable hinge-loss mrfs', in *ICML*, pp. 381–390, (2015).

[3] Stephen H Bach, Matthias Broecheler, Bert Huang, and Lise Getoor, 'Hinge-loss markov random fields and probabilistic soft logic', *JMLR*, **18**(109), 1–67, (2017).

[4] Aleksandar Bojchevski, Oleksandr Shchur, Daniel Zugner, and Stephan Gunnemann, 'Netgan: Generating graphs via random walks', in *ICML*, (2018).

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al., 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Foundations and Trends® in Machine learning*, **3**, 1–122, (2011).

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann Lecun, 'Spectral networks and locally connected networks on graphs', in *ICLR*, (2014).

[7] Bei Chen, Ning Chen, Jun Zhu, Jiaming Song, and Bo Zhang, 'Discriminative nonparametric latent feature relational models with data augmentation.', in *AAAI*, (2016).

[8] Nicola De Cao and Thomas Kipf, 'Molgan: An implicit generative model for small molecular graphs', *arXiv preprint arXiv:1805.11973*, (2018).

[9] Ismail El-Helw, Rutger Hofman, Wenzhe Li, Sungjin Ahn, Max Welling, and Henri Bal, 'Scalable overlapping community detection', in *Parallel and Distributed Processing Symposium Workshops*, (2016).

[10] Xuhui Fan, Longbing Cao, and Richard Yi Da Xu, 'Dynamic infinite mixed-membership stochastic blockmodel', *IEEE transactions on neural networks and learning systems*, **26**, 2072–2085, (2015).

[11] Xuhui Fan, Richard Yi Da Xu, and Longbing Cao, 'Copula mixed-membership stochastic blockmodel.', in *IJCAI*, (2016).

[12] Xuhui Fan, Richard Yi Da Xu, Longbing Cao, and Yin Song, 'Learning nonparametric relational models by conjugately incorporating node information in a network', *IEEE transactions on cybernetics*, **47**, 589–599, (2017).

[13] Xuhui Fan, Bin Li, and Scott Sisson, 'The binary space partitioning-tree process', in *AISTATS*, (2018).

[14] James Foulds, Shachi Kumar, and Lise Getoor, 'Latent topic networks: A versatile probabilistic programming framework for topic models', in *ICML*, (2015).

[15] Wenjie Fu, Le Song, and Eric P Xing, 'Dynamic mixed membership blockmodel for evolving networks', in *ICML*, (2009).

[16] Francesco Giannini, Michelangelo Diligenti, Marco Gori, and Marco Maggini, 'Characterization of the convex łukasiewicz fragment for learning from constraints', in *AAAI*, (2018).

[17] Prem K Gopalan, Sean Gerrish, Michael Freedman, David M Blei, and David M Mimno, 'Scalable inference of overlapping communities', in *NeurIPS*, (2012).

[18] Qirong Ho, Ankur Parikh, Le Song, and Eric Xing, 'Multiscale community blockmodel for network exploration', in *AISTATS*, (2011).

[19] Qirong Ho, Ankur P Parikh, Le Song, and Eric P Xing, 'Infinite hierarchical mmsb model for nested communities/groups in social networks', *Statistics*, **1050**, 9, (2010).

[20] Qirong Ho, Le Song, and Eric Xing, 'Evolving cluster mixed-membership blockmodel for time-evolving networks', in *AISTATS*, (2011).

[21] Weihong Huang, Yan Liu, and Yuguo Chen. Mixed membership stochastic blockmodels for heterogeneous networks, 2018.

[22] Charles Kemp, Joshua B Tenenbaum, Thomas L Griffiths, Takeshi Yamada, and Naonori Ueda, 'Learning systems of concepts with an infinite relational model', in *AAAI*, (2006).

[23] Dae Il Kim, Michael Hughes, and Erik Sudderth, 'The nonparametric metadata dependent relational model', in *ICML*, (2012).

[24] Myunghwan Kim and Jure Leskovec, 'Latent multi-group membership graph model', in *ICML*, (2012).

[25] Thomas N Kipf and Max Welling, 'Semi-supervised classification with graph convolutional networks', in *ICLR*, (2017).

[26] Phaedon-Stelios Koutsourelakis and Tina Eliassi-Rad, 'Finding mixed-memberships in social networks.', in *AAAI Spring Symposium: Social Information Processing*, (2008).

[27] Jure Leskovec and Rok Sosič, 'Snap: A general-purpose network analysis and graph-mining library', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **8**(1), 1, (2016).

[28] Percy Liang and Dan Klein, 'Online em for unsupervised models', in *NAACL-HLT*, (2009).

[29] James Lloyd, Peter Orbanz, Zoubin Ghahramani, and Daniel M Roy, 'Random function priors for exchangeable arrays with applications to graphs and relational data', in *NIPS*, (2012).

[30] Kurt Miller, Michael I Jordan, and Thomas L Griffiths, 'Nonparametric latent feature models for link prediction', in *NeurIPS*, (2009).

[31] Ian Porteous, Evgeniy Bart, and Max Welling, 'Multi-hdp: A non parametric bayesian model for tensor factorization.', in *AAAI*, (2008).

[32] Daniel M Roy, Yee Whye Teh, et al., 'The mondrian process.', in *NIPS*, pp. 1377–1384, (2008).

[33] Tracy M Sweet, Andrew C Thomas, and Brian W Junker, 'Hierarchical mixed membership stochastic blockmodels for multiple networks and experimental interventions', *Handbook on mixed membership models and their applications*, 463–488, (2014).

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio, 'Graph attention networks', *ICLR*, (2018).

[35] Eric P Xing, Wenjie Fu, Le Song, et al., 'A state-space mixed membership blockmodel for dynamic network tomography', *The Annals of Applied Statistics*, **4**, 535–566, (2010).

[36] Yunfeng Xu, Hua Xu, Dongwen Zhang, and Yan Zhang, 'Finding overlapping community from social networks based on community forest model', *Knowledge-Based Systems*, **109**, 238–255, (2016).

[37] Hongxia Yang and Aurelie Lozano, 'Multi-relational learning via hierarchical nonparametric bayesian collective matrix factorization', *Journal of Applied Statistics*, **42**, 1133–1147, (2015).

[38] Jaewon Yang, Julian McAuley, and Jure Leskovec, 'Community detection in networks with node attributes', in *ICDM*, (2013).

[39] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec, 'Gnn explainer: A tool for post-hoc explanation of graph neural networks', in *NeurIPS*, (2019).

[40] Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec, 'Graphrnn: Generating realistic graphs with deep auto-regressive models.', in *ICML*, (2018).

[41] Yuan Zhang, Tianshu Lyu, and Yan Zhang, 'Hierarchical community-level information diffusion modeling in social networks', in *SIGIR*, (2017).

[42] He Zhao, Lan Du, and Wray L. Buntine, 'Leveraging node attributes for incomplete relational data', in *ICML*, (2017).

[43] Mingyuan Zhou, 'Infinite edge partition models for overlapping community detection and link prediction', in *AISTATS*, (2015).