Stochastic Blockmodels meet Graph Neural Networks

Anonymous Authors(s)
Affiliation
Address
email

Abstract

Stochastic Blockmodels (SBM) and their variants, e.g., mixed-membership and overlapping stochastic blockmodels are latent variable models for graphs, and have proven to be very successful for tasks, such as discovering the community structure and link prediction on graph-structured data. Recently, graph neural networks, e.g., graph convolutional networks, have also emerged as a promising approach to learn powerful representations (embeddings) for the nodes in the graph by exploiting various graph properties, such as locality, invariance, etc. In this work, we unify these two directions by developing a novel, *sparse* variational autoencoder for graphs, that retains the nice interpretability properties of SBMs, while also enjoying the excellent predictive performance of graph neural nets. Moreover, our framework is accompanied by a fast *recognition model* that enables fast inference of the node embeddings (which would be of independent interest even for inference in traditional SBMs). Although we develop this framework for a particular type of SBM, namely the *overlapping* stochastic blockmodel, our framework can be easily adapted for other types of SBMs as well. Experimental results on several benchmarks datasets demonstrate that our model outperforms various state-of-the-art methods for community discovery and link prediction.

Introduction

Learning the latent structure in graph-structured data (Fortunato 2010; Goldenberg et al. 2010; Schmidt and Morup 2013) is an important problem in a wide range of domains, such as social and biological network analysis, recommender systems, etc. These latent structures help discover the underlying communities in the network, as well as in predicting potential links between nodes. Latent space model (Hoff, Raftery, and Handcock 2002) and their structured extensions, such as stochastic blockmodel (Nowicki and Snijders 2001) and variants, such as infinite relational model (IRM) (Kemp et al. 2006), mixed-membership stochastic blockmodel (MMSB) (Airoldi et al. 2008), overlapping stochastic blockmodel (OSBM) (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011), etc., accomplish this by learning low-dimensional, interpretable node embeddings. These embeddings can be directly used to identify the community membership(s) of each node in the graph.

The overlapping stochastic blockmodels (OSBM), also known as the latent feature relational model (LFRM), is a particularly appealing model for relational data (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011; Zhu, Song, and Chen 2016). The OSBM models each node in the graph as belonging to one or more communities using a binary membership vector, and defines the link probability between any pair of nodes as a *bilinear* function of their community membership vectors. Despite its appealing properties, however, the OSBM has a number of limitations. In particular, although usually considered to be more expressive (Miller, Griffiths, and Jordan 2009) than models such as IRM and MMSB, a single layer of binary vector based node embedding, and the bilinear model for the link generation can still limit the expressiveness of OSBM. Moreover, OSBM has a challenging inference procedure, which primarily relies on MCMC (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011) or mean-field variational inference (Zhu, Song, and Chen 2016). Although some recent models have tried to improve the expressiveness of OSBM, e.g., by assuming a *deep* hierarchy of binary-vector based node embeddings (Hu, Rai, and Carin 2017), inference in such models remains intractable, requiring expensive MCMC based inference. It is therefore desirable to have a model that retains the basic spirit to OSBM (e.g., easy interpretability and strong link prediction performance), but has higher expressiveness, and a simpler and scalable inference procedure.

Motivated by these desiderata, we develop a deep generative framework for graph-structured data that inherits the easy interpretability of overlapping stochastic blockmodels, but is much more expressive, and enjoys a fast inference procedure. Our framework is based on a novel, *sparse* variant of the variational autoencoder (VAE) (Kingma and Welling 2013), designed to model graph-structured data. Our VAE based framework comprises a nonlinear generator/decoder for the graph and a nonlinear encoder based on graph convolutional network (GCN) (Kipf and Welling 2016a) (although other graph neural networks can also be used). Our framework posits each node of the graph to have an embedding in form of a sparse latent representation (modeled by a Beta-Bernoulli process (Griffiths and Ghahramani 2011),

which also enables *learning* the size of the embeddings). The generator/decoder part of the VAE models the probability of a link between two nodes via a nonlinear function (defined by a deep neural network) of their associated embeddings. The encoder part of the VAE consists of a fast *recognition* model that is designed leveraging reparameterization tricks for Beta and Bernoulli distributions (Maddison, Mnih, and Teh 2017; Nalisnick and Smyth 2017). The recognition model, based on stochastic gradient variational Bayes (SGVB) inference, enables fast inference of the node embeddings. In contrast, the traditional stochastic blockmodels rely on iterative MCMC or variational inference procedures for inferring the node embeddings. Consequently, the SGVB inference algorithm we develop is also of independent interest since the recognition model enables fast inference of the node embeddings in *single-layer* overlapping stochastic blockmodels.

Background

We first introduce some notation and then briefly describe the overlapping stochastic blockmodel (OSBM) (Latouche, Birmelé, and Ambroise 2011; Miller, Griffiths, and Jordan 2009; Zhu, Song, and Chen 2016). As we will describe subsequently in the next section, our deep generative model enriches OSBM by endowing it with a deep architecture based on a *sparse* variational autoencoder and a fast inference algorithm based on a recognition model. We assume that the graph is given as an adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$, where N denotes the number of nodes. We assume $A_{nm} = 1$ if there exist a link from node n and node m, and $A_{nm} = 0$ otherwise. In addition to $\mathbf{A}$, for each node, we may also be provided node features. These are given in form of an $N \times D$ matrix $\mathbf{X}$, with $\boldsymbol{x}_n \in \mathbb{R}^D$ being the node features for node n, and D being the number of observed features.

The OSBM (Latouche, Birmelé, and Ambroise 2011; Miller, Griffiths, and Jordan 2009; Zhu, Song, and Chen 2016) is a stochastic blockmodel for networks and assumes each node n to be associated with a binary vector (node embedding), also termed as *latent feature vector*, $\boldsymbol{z}_n \in \{0,1\}^K$ where $z_{nk} = 1$ denotes that node n belongs to cluster/community k, and $z_{nk} = 0$ otherwise. The OSBM allows each node to simultaneously belong to multiple communities and defines the link probability between two nodes via a bilinear function of their latent feature vectors

$$p(A_{nm} = 1|\boldsymbol{z}_n, \boldsymbol{z}_m, \mathbf{W}) = \sigma(\boldsymbol{z}_n^\top \mathbf{W} \boldsymbol{z}_m) \tag{1}$$

where the entry $w_{k\ell}$ in $\mathbf{W} \in \mathbb{R}^{K\times K}$ affects the probability of a link between node n and node m belonging to cluster k and cluster ℓ, respectively.

The nonparametric latent feature relational model (LFRM) is a specific type of OSBM that leverages the Indian Buffet Process (IBP) prior (Miller, Griffiths, and Jordan 2009) on the $N \times K$ binary matrix $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N]^\top$ of the node-community membership vectors, which enables learning the number of communities. Inference in LFRM/OSBM is typically performed via MCMC or variational inference (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011; Zhu, Song, and Chen 2016), which tends to be slow and usually cannot scale easily to more than a few hundred nodes.

Deep Generative OSBM

We now present our sparse VAE based deep generative framework for overlapping stochastic blockmodel. The proposed architecture, depicted in Fig. 1 (Left), associates each link $A_{nm} \in \{0,1\}$ with two latent embeddings $\boldsymbol{z}_n$ and $\boldsymbol{z}_m$ (of nodes associated with this link), and defines each link probability via a nonlinear function of the embeddings of its associated nodes. Unlike standard VAE which assumes dense, Gaussian-distributed embeddings, since we wish to use the node embeddings to also infer the community membership(s) of each node (as it is one of the goals of stochastic blockmodels), we impose sparsity on the node embeddings by modeling them as a sparse vector of the form $\boldsymbol{z}_n = \boldsymbol{b}_n \odot \boldsymbol{r}_n$ where $\boldsymbol{b}_n \in \{0,1\}^K$ is a binary vector modeled using a stick-breaking process prior (Teh, Grr, and Ghahramani 2007) and $\boldsymbol{r}_n \in \mathbb{R}^K$ is a real-valued vector with a Gaussian prior. Modeling $\boldsymbol{b}_n$ using the stick-breaking prior enables learning the node embedding size K from data. Note that, unlike the OSBM/LFRM, which assumes the node embedding $\boldsymbol{z}_n$ to be a strictly binary vector, our framework models it as a sparse real-valued vector, which provides a more flexible and informative representation for the nodes. In particular, this enables us to infer not just the node's membership into communities but also the *strength* of the membership in each of the communities the node belongs to. Specifically, $b_{nk} \in \{0,1\}$ denotes whether node n belongs to cluster k or not and $r_{nk} \in \mathbb{R}$ denotes the strength.

The VAE Generator/Decoder

Given the node embeddings $\boldsymbol{z}_n = \boldsymbol{b}_n \odot \boldsymbol{r}_n$, the VAE generator generates each link in the graph as $A_{nm} \sim p_\theta(\boldsymbol{z}_n, \boldsymbol{z}_m)$, where the probability distribution p_θ defines a *decoder* or generator model for the graph. This decoder can consist of one or more layers of deterministic *nonlinear* transformation of the node embeddings $\boldsymbol{z}_n$. Denoting the overall transformation for a node embedding $\boldsymbol{z}_n$ as $f(\boldsymbol{z}_n) = \boldsymbol{f}_n$, we model the probability of a link as $p(A_{nm} = 1|\boldsymbol{f}_n, \boldsymbol{f}_m) = \sigma(\boldsymbol{f}_n^\top \boldsymbol{f}_m)$, where the nonlinear function f can be modeled by a deep neural net (in our experiments, we use a deep neural net with each hidden layer having leaky ReLU nonlinearity). Fig. 1 (Left) depicts the generator.

We model the binary vector $\boldsymbol{b}_n \in \{0,1\}^K$, denoting node-community memberships, using the stick-breaking construction of the IBP (Teh, Grr, and Ghahramani 2007), which enables us to learn the *effective* K by specifying a sufficiently large truncation level K. The stick-breaking construction is given as follows

$$v_k \sim \text{Beta}(\alpha, 1), \quad k = 1, \ldots, K \tag{2}$$

$$\pi_k = \prod_{j=1}^{k} v_j, \quad b_{nk} \sim \text{Bernoulli}(\pi_k) \tag{3}$$

We further assume a Gaussian prior on membership strengths $\boldsymbol{r}_n \in \mathbb{R}^K$, i.e., $p(\boldsymbol{r}_n) = \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

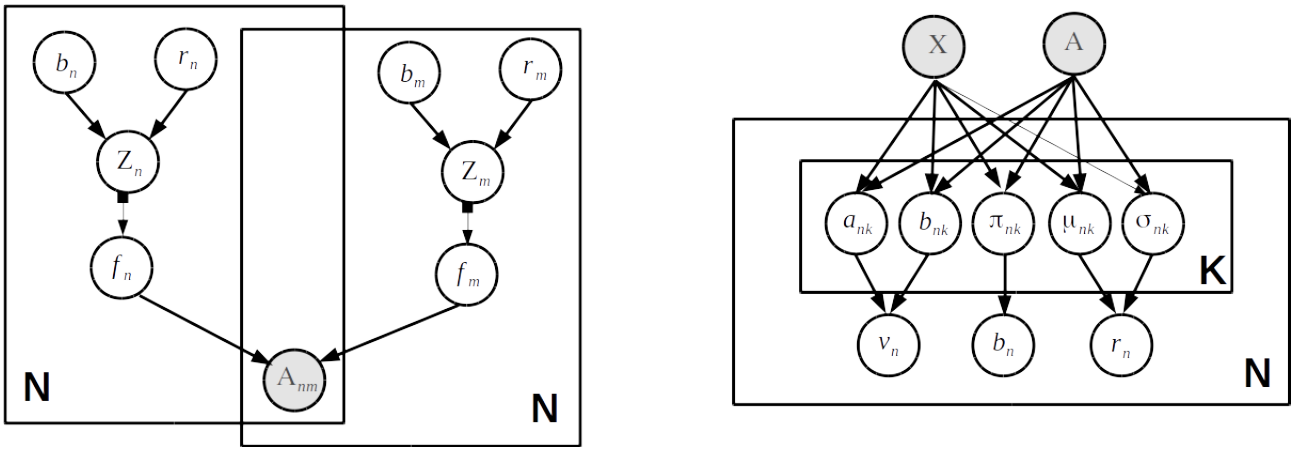

Figure 1: (Left) The generator/decoder model in the plate notation. Note that the mapping from $\boldsymbol{z}_n$ to $\boldsymbol{f}_n$ is a deterministic nonlinear transformation modeled by a deep neural network, (Right) The encoder model, which is defined by a graph convolutional network (Kipf and Welling 2016a) that takes as input the network $\mathbf{A}$ and node features $\mathbf{X}$ (if available) and outputs the parameters of the variational distributions of the model parameters

The VAE Encoder

Our framework consists of a *nonlinear* encoder to infer the node embedding $\boldsymbol{z}_n$ for each node using a fast, non-iterative *recognition model* (Kingma and Welling 2013). Denoting the parameters of the variational posterior for the embeddings of all the nodes, collectively, as $\{\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r}\}$, we consider an approximation to model's true posterior $p(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r}|\mathbf{A}, \mathbf{X})$ with a variational posterior of the form $q_\phi(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r})$. For simplicity, we consider mean-field approximation, which allows us to factorize the posterior as $q_\phi(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r}) = \prod_{k=1}^{K}\prod_{n=1}^{N} q_\phi(v_{nk})q_\phi(b_{n,k})q_\phi(r_{n,k})$. Our nonlinear encoder, as shown in Fig. 1 (Right), assumes variational distributions on the local variables of each node, i.e., $\boldsymbol{v}_n$, $\boldsymbol{b}_n$ and $\boldsymbol{r}_n$, and defines the variational parameters of these distributions as the outputs of a graph convolutional network (GCN) (Kipf and Welling 2016a), which takes as input the network $\mathbf{A}$ and the node feature matrix $\mathbf{X}$. GCN has recently emerged as a flexible encoder of graph-structured data (similar in spirit to convolutional neural networks for images) which makes it an ideal choice of the encoder in our VAE based generative model for graphs. Although here we have used the vanilla GCN in our architecture, more advanced variants of GCN, such as GraphSAGE (Hamilton, Ying, and Leskovec 2017) can also be used as the encoder. The variational distributions have the following forms

$$q_\phi(v_{nk}) = \text{Beta}(c_{nk}, d_{nk}) \quad k = 1, \ldots, K \tag{4}$$
$$q_\phi(b_{nk}) = \text{Bernoulli}(\pi_{nk}) \quad k = 1, \ldots, K \tag{5}$$
$$q_\phi(\boldsymbol{r}_n) = \mathcal{N}(\boldsymbol{\mu}_n, \text{diag}(\boldsymbol{\sigma}_n^2)) \tag{6}$$

where $c_{nk}, d_{nk}, \pi_{nk}, \mu_n$, and σ are outputs of a GCN, i.e., $\{c_k, d_k, \pi_k, \mu_k, \sigma_k\}_{n=1}^{n=N} = \text{GCN}(\mathbf{A}, \mathbf{X})$. We use the stochastic gradient variational Bayes (SGVB) algorithm (Kingma and Welling 2013) to infer the parameters of the above variational distributions. Our SGVB algorithm is based on a reparameterization trick for Beta and Bernoulli distributions (Maddison, Mnih, and Teh 2017; Nalisnick and Smyth 2017). We provide details of the reparameterization and loss formulation in the inference section.

Some Special Cases

Some existing models for graph-structured data can be seen as special cases of our framework. Recall that we model the node embeddings as $\boldsymbol{z}_n = \boldsymbol{b}_n \odot \boldsymbol{r}_n$ and our generative model is of the form $A_{nm} \sim p_\theta(\boldsymbol{z}_n, \boldsymbol{z}_m)$. If we ignore the community strength latent variable $\boldsymbol{r}_n$, i.e., $\boldsymbol{z}_n$ is defined simply as $\boldsymbol{z}_n = \boldsymbol{b}_n$ (just a binary vector) and further define p_θ as a Bernoulli distribution with its probability being a bilinear function of the embeddings $\boldsymbol{z}_n$ and $\boldsymbol{z}_m$, then we recover the OSBM/LFRM (Latouche, Birmelé, and Ambroise 2011; Miller, Griffiths, and Jordan 2009). Note, however, that while OSBM/LFRM typically rely on MCMC or variational inference, our framework can leverage SGVB for efficient inference.

Likewise, if we define $\boldsymbol{z}_n = \boldsymbol{r}_n$, i.e., a *dense* vector and define p_θ as a Bernoulli distribution with its probability being a bilinear function of the embeddings, we recover the Eigenmodel or latent-space model (LSM) (Hoff, Raftery, and Handcock 2002). Note that this model cannot infer K since the binary vector $\boldsymbol{b}_n$ is not present. Finally, if p_θ as a Bernoulli distribution with its probability being a *nonlinear* function of the embeddings, then we recover the recently proposed graph VAE (VGAE) model (Kipf and Welling 2016b), which can also be seen as a nonlinear extension of LSM. Moreover, note that a key limitation of LSM and VGAE is that these cannot be used to infer the community structure (due to the non-sparse nature of $\boldsymbol{z}_n$) and usually can only be used for link-prediction tasks.

Inference

Our variational posterior $q_\phi(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r}) = \prod_{k=1}^{K}\prod_{n=1}^{N} q_\phi(v_{nk})q_\phi(b_{n,k})q_\phi(r_{n,k})$, with the individual posteriors over the latent variables defined as

$$
\begin{aligned}
q_\phi(v_{nk}) &= \text{Beta}(v_{nk}|c_k(\mathbf{A},\mathbf{X}), d_k(\mathbf{A},\mathbf{X})) \\
q_\phi(b_{nk}) &= \text{Bernoulli}(b_{nk}|\pi_k(\mathbf{A},\mathbf{X})) \\
q_\phi(\boldsymbol{r}_n) &= \mathcal{N}(\boldsymbol{\mu}_n(\mathbf{A},\mathbf{X}), \text{diag}(\boldsymbol{\sigma}_n^2(\mathbf{A},\mathbf{X})))
\end{aligned}
$$

where $c_k, d_k, \pi_k, \boldsymbol{\mu}_n$and $\boldsymbol{\sigma}_n$ are a function of the GCN encoder with inputs $\mathbf{A}$ and $\mathbf{X}$.

We define the loss function $\mathcal{L}$ parameterized by inference network (encoder) parameters (ϕ) and generator parameters (θ) by minimizing the *negative* of the evidence lower bound (ELBO)

$$
\begin{aligned}
\mathcal{L} = \sum_{n=1}^{N} \Big(&\text{KL}[q_\phi(\boldsymbol{b}_n|\boldsymbol{v}_n) \,||\, p_\theta(\boldsymbol{b}_n|\boldsymbol{v}_n)] + \text{KL}[q_\phi(\boldsymbol{r}_n) \,||\, p_\theta(\boldsymbol{r}_n)] \\
&+ \text{KL}[q_\phi(\boldsymbol{v}_n) \,||\, p(\boldsymbol{v}_n)]\Big) - \sum_{n=1}^{N}\sum_{m=1}^{N} \Big(\mathbb{E}_q[\log p_\theta(A_{nm}|\boldsymbol{z}_n, \boldsymbol{z}_m)]\Big)
\end{aligned} \tag{7}
$$

where $\text{KL}[q(\cdot)||p(\cdot)]$ is the Kullback-Leibler divergence between $q(\cdot)$ and $p(\cdot)$. For the encoder and decoder parameters we infer the point estimates, while we learn the distribution over the latent variables $\boldsymbol{b}, \boldsymbol{v},$ and $\boldsymbol{r}$

Our variational autoencoder for link generation is trained using Stochastic Gradient Variational Bayes (SGVB) (Kingma and Welling 2013). SGVB can be used to perform inference for a broad class of non-conjugate models and is therefore quite appealing to Bayesian nonparametric models such as those based on stick-breaking priors that we use in our framework. SGVB uses differentiable Monte Carlo (MC) expectations to learn the model parameters. Specifically, it requires *differentiable, non-centered parameterization (DNCP)* (Kingma and Welling 2014) to allow backpropagation. However, our model has expectations over Beta and Bernoulli distributions, neither of which permit easy reparameterization as required by SGVB. We leverage the recent developments on reparameterizing these distributions (Maddison, Mnih, and Teh 2017; Nalisnick and Smyth 2017), which consequently leads to a simple inference procedure.

Following (Nalisnick and Smyth 2017), we alleviate the issue by first approximating the Beta posterior in Eq. 4 with the Kumaraswamy distribution which is defined as: $Kumar(x; a, b) = abx^{a-1}(1-x^a)^{b-1}$ for $x \in (0, 1)$ and $a, b > 0$. The closed-form inverse CDF allows easy reparameterization and samples for v_{nk} (with parameters c_{nk} and d_{nk}) can now be drawn using:

$$
\begin{aligned}
u &\sim \text{Uniform}(0, 1) \\
v_{nk} &\overset{d}{=} (1 - u^{\frac{1}{d_{nk}}})^{\frac{1}{c_{nk}}}
\end{aligned} \tag{8}
$$

We compute the KL divergence between Kumaraswamy $q(\boldsymbol{v})$ and the Beta distribution $p(\boldsymbol{v})$ by taking the finite approximation of the infinite sum as mentioned in (Nalisnick and Smyth 2017).

For the Bernoulli random variable, we use the Binary Concrete distribution (Maddison, Mnih, and Teh 2017; Jang, Gu, and Poole 2017) at the time of training as a continuous relaxation to get the biased low-variance estimates of the gradient. The KL between two Bernoullis is relaxed using two Binary Concrete distributions.

We reparameterize b_{nk} defined by a Bernoulli with probability π_{nk} (in Eq. 3 and Eq. 5) with reparameterization:

$$
\begin{aligned}
L &= \log\Big(\frac{u}{1-u}\Big) \\
b_{nk} &\overset{d}{=} \sigma\left(\frac{\text{logit}(\pi_{nk}) + L}{\lambda}\right)
\end{aligned} \tag{9}
$$

where $\sigma(\cdot)$ is the sigmoid function, $logit(\cdot)$ is the inverse-sigmoid function, λ is the relaxation temperature and $u \sim \text{Uniform}(0, 1)$.

Structured Mean-Field: Since the vanilla mean-field variational inference ignores the posterior dependencies among the latent variable, we also considered Structured Stochastic Variational Inference (SSVI) (Hoffman 2014; Hoffman et al. 2013), which allows global-local parameter dependency and improves upon the mean-field approximation. We considered $\boldsymbol{v}$ (and its variational parameters c and d) as global parameters and imposed a hierarchical structure on $\boldsymbol{b}_n$ by conditioning it on $\boldsymbol{v}$. The variational posterior of our framework using SSVI can be factorized as $q_\phi(\boldsymbol{v}, \boldsymbol{b}, \boldsymbol{r}) = \prod_{k=1}^{K} q_\phi(v_k) \prod_{n=1}^{N} q_\phi(b_{n,k}|\boldsymbol{v}) q_\phi(r_{n,k})$ with $q_\phi(v_k) = \text{Beta}(c_k, d_k); q_\phi(b_{nk}) = \text{Bernoulli}(\pi_k); \pi_k = \prod_{j=1}^{K} v_k$, where c_k, d_k are parameters to be learned. In practice, we found structured mean-field to perform better than the mean-field, and our model implementation uses the former.

Related Work

Our work can be seen as bridging two strands of research on modeling graphs: (1) structured latent variable models for graphs, such as stochastic blockmodels and its variants (Kemp et al. 2006; Airoldi et al. 2008; Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011); and (2) deep learning models for graphs, such as graph convolutional networks (Kipf and Welling 2016a). Our effort is motivated by the goal is to harness their complementary strengths in order to develop a deep generative stochastic blockmodel for graphs that also enjoys an efficient inference procedure.

The most prominent methods in stochastic blockmodels include models that associate each node to a single community (Nowicki and Snijders 2001; Kemp et al. 2006), a mixture of communities (Airoldi et al. 2008), and an overlapping set of communities (Miller, Griffiths, and Jordan 2009; Latouche, Birmelé, and Ambroise 2011; Yang and Leskovec 2012; Zhou 2015). While stochastic blockmodels have nice interpretability, these models usually assume the links of the networks to modeled as a simple bilinear function of the node embeddings, which may not be able to capture the nonlinear interactions between the nodes (Yan, Xu, and Qi 2011). An approach to model such nonlinear interactions was proposed in (Yan, Xu, and Qi 2011), using matrix-variate Gaussian process. However, despite the modeling flexibility, inference in this model is considerably challenging and the model is usually infeasible to run even on moderate-sized networks with more than 100 nodes.

There is also a significant recent interest in non-probabilistic deep learning models for graphs. Some of the

prominent works in this direction include DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) and graph autoencoders (GAE) (Kipf and Welling 2016a; Hamilton, Ying, and Leskovec 2017). DeepWalk is inspired by the idea of word embeddings. It treats each node as a "document" by starting a random walk at that node and taking the nodes encountered in the path taken as the word in that document. It uses document/word embedding methods to the learn embedding of each node. In contrast, the GAE approaches are based on the idea of graph convolutional networks (GCN) (Kipf and Welling 2016a). This line of work nicely complements our contribution since modules like GCN can be effectively used to design the encoder model for our deep generative framework. In particular, as noted in the model description, our encoder is essentially a GCN. We believe that such advances in graph encoding can be used as modules to design new deep generative models for relational data.

Despite the resounding success of deep generative models for images and text data, there has been relatively much less work on deep generative models for relational data (Hu, Rai, and Carin 2017; Wang, Shi, and Yeung 2017; Kipf and Welling 2016b). Among the existing methods, (Hu, Rai, and Carin 2017) proposed an extension of the LFRM via a deep hierarchy of binary latent features for each node. However, this model relies on expensive batch MCMC inference, which precludes its applicability to large-scale networks. Another deep latent variable model was proposed recently in (Wang, Shi, and Yeung 2017). However, this model too has a difficult inference procedure, requiring a model-specific inference procedure. Moreover, the node embeddings are not interpretable. Perhaps the closest in spirit to our work is the recent work on variational graph autoencoders (VGAE) (Kipf and Welling 2016b) and Graphite (Grover, Zweig, and Ermon 2018). However, these are built on top of standard VAE and consequently do not have direct interpretability of node embeddings as desired by stochastic blockmodels. This leads to a model with significantly different properties and a significantly different inference procedure as compared to (Kipf and Welling 2016b). Moreover, our VAE architecture is nonparametric in nature and can infer the node embedding size.

Experiments

We report our experimental results on several synthetic and real world datasets to demonstrate the efficacy of our model. Our experiment results include quantitative comparisons on the task of link prediction as well as qualitative results, such as using the embeddings to discover the underlying communities in the network data. Our qualitative results are meant to demonstrate the expressiveness of the latent space that our model can infer. The expressive nature of our model is the result of the sparse and interpretable embedding for each node of the graph. In particular, we show that these sparse embeddings can be interpreted as the memberships and strength of memberships of each node in one or more communities. The hyperparameter settings used for all the experiments are included in supplementary section.

First we evaluate the performance of our framework on link-prediction and compare it with various baselines on several benchmark datasets on moderate (about 2000 nodes) to large-scale (about 20,000 nodes) datasets. We then analyze the latent structure $\boldsymbol{z}_n$ learned by our model on a synthetic and a real-world co-authorship dataset. We compare the latent structure with the embeddings learned by the variational graph autoencoder (VGAE) (Kipf and Welling 2016b). We also inspect the community structure on the real-world co-authorsip dataset and show that the framework proposed is able to readily capture the underlying communities. We refer to our framework as DGLFRM (Deep Generative Latent Feature Relational Model), which refers to our most general model with sparse embeddings $\boldsymbol{z}_n = \boldsymbol{b}_n \odot \boldsymbol{r}_n$ with nonlinear generator and nonlinear encoder. We also consider a variant of DGLFRM with binary embeddings $\boldsymbol{z}_n = \boldsymbol{b}_n$, which we refer to as DGLFRM-B (the 'B' here denotes "binary").

Baselines

For link prediction, we compare our model with four baselines, one of which is a simplified variant of DGLFRM and akin to the LFRM (Miller, Griffiths, and Jordan 2009) which is an overlapping stochastic blockmodel. The original LFRM which uses MCMC based inference was infeasible to run on the datasets we have used in our experiments. On the other hand, DGLFRM with $\boldsymbol{z}_n = \boldsymbol{b}_n$ and bilinear decoder (link generation model) is similar to LFRM, but with a much faster SGVB based inference (we will refer to this simplified variant of DGLFRM simply as LFRM).

Among the other three baselines, two of the baselines - Spectral Clustering (SC) and DeepWalk (DW) (Perozzi, Al-Rfou, and Skiena 2014) - learn node embeddings, which we use to compute the link probability as $\sigma(\boldsymbol{z}_n^\top \boldsymbol{z}_m)$. The third baseline is the recently proposed variational autoencoder on graphs (VGAE) (Kipf and Welling 2016b). Note that none of these baselines can be used for community detection since the real-valued embeddings learned by these baselines are not interpretable (unlike our model which learns sparse embeddings, with nonzeros denoting community memberships).

Datasets

In our experiments, we consider five real-world datasets, with three datasets consisting of side information in form of the node features (which our model can leverage), and the other two datasets having only the link information. For the link prediction experiments, all the models are provided a partially complete network (with unknown part to be predicted). The node features (when available) are provided to all the models. The description of each data set is provided below:

- **NIPS12**: The NIPS12 coauthor network (Zhou 2015) includes all the 2037 authors in NIPS papers vols 0-12, with 3134 edges. This network has no side information.
- **Yeast**: We also consider the Yeast protein interaction network (Zhou 2015) with 2361 nodes and 6646 non-self edges. This network also has no side information.
- **Cora**: Cora network is a citation network consisting of 2708 documents. The datasets contain sparse bag-of-words feature vectors of length 1433 for each document.

Table 1: AUC ROC. The - denotes that the result is not available

Method	NIPS12	Yeast	Cora	Citeseer	Pubmed
SW	-	-	$0.8460 \pm .0001$	$0.8050 \pm .0002$	$0.8420 \pm .0002$
DW	-	-	$0.8310 \pm .0001$	$0.8050 \pm .0001$	$0.8440 \pm .0000$
VGAE	$0.9029 \pm .0031$	$\mathbf{0.8840} \pm .0004$	$0.9260 \pm .0001$	$0.9200 \pm .0002$	$0.9470 \pm .0002$
LFRM	$0.8806 \pm .0065$	$0.8628 \pm .0012$	$0.9096 \pm .0026$	$0.8965 \pm .0035$	$0.9152 \pm .0041$
DGLFRM-B	$\mathbf{0.9156} \pm .0023$	$0.8735 \pm .0061$	$0.9207 \pm .0047$	$0.9007 \pm .0045$	$0.9396 \pm .0052$
DGLFRM	$0.8969 \pm .0053$	$0.8672 \pm .0051$	$\mathbf{0.9355} \pm .0039$	$\mathbf{0.9254} \pm .0066$	$\mathbf{0.9595} \pm .0032$

Table 2: Average Precision (AP). The - denotes that the result is not available

Method	NIPS12	Yeast	Cora	Citeseer	Pubmed
SC	-	-	$0.8850 \pm .0000$	$0.8500 \pm .0100$	$0.8780 \pm .0100$
DW	-	-	$0.8500 \pm .0100$	$0.8360 \pm .0100$	$0.8440 \pm .0000$
VGAE	$0.9111 \pm .0025$	$0.8831 \pm .0021$	$0.9328 \pm .0001$	$0.9200 \pm .0002$	$0.9470 \pm .0002$
LFRM	$0.9119 \pm .0025$	$0.8642 \pm .0028$	$0.9060 \pm .0033$	$0.9118 \pm .0031$	$0.9197 \pm .0054$
DGLFRM-B	$\mathbf{0.9363} \pm .0011$	$\mathbf{0.8897} \pm .0052$	$0.9219 \pm .0041$	$0.9153 \pm .0031$	$0.9454 \pm .0050$
DGLFRM	$0.9233 \pm .0038$	$0.8711 \pm .0098$	$\mathbf{0.9377} \pm .0037$	$\mathbf{0.9356} \pm .0055$	$\mathbf{0.9570} \pm .0035$

Table 3: Example of communities inferred by our model on NIPS data. Authors ordered by strength of membership in these communities.

Cluster	Authors
Probabilistic Modeling	**Sejnowski T**, Hinton G, Dayan P, Jordan M, Williams C
Reinforcement Learning	Barto A, Singh S, Sutton R, Connolly C, Precup D
Robotics/Vision	Shibata T, Peper F, Thrun S, Giles C, Michel A
Computational Neuroscience	Baldi P, Stein C, Rinott Y, Weinshall D, Druzinsky R
Neural Networks	Pearlmutter B Abu-Mostafa Y, LeCun Y, **Sejnowski T**, Tang A

These are used as node features. The network has total 5278 links.

- **Citeseer**: Citeseer is a citation network consisting of 3312 scientific publications from six categories: agents, AI, databases, human computer interaction, machine learning, and information retrieval. The side information for the dataset is the category label for each paper which is converted into a one-hot representation. These one-hot vectors are used as node features. The network has total 4552 links.
- **Pubmed**: A citation network consisting of 19717 nodes. The dataset contains sparse bag-of-words feature vectors of length 500 for each document. These are used as node features. The network has total 44324 links.

Link Prediction

We use Area Under the ROC Curve (AUC) and Average Precision (AP) to compare our model with the other baselines on the task of link prediction. For all the datasets, we hold out 10% and 5% links as our test set and validation set, respectively and use the validation set to fine tune the hyperparameters. We take the average of AUC-ROC and AP scores by running our model on 10 random splits of each dataset to compare with the other baselines. The AUC-ROC scores of our models and the various baselines are shown in Table 1 and AP scores are shown in Table 2. As shown in the tables, our models outperform the baselines on almost all the datasets. We would again like to highlight that unlike the baselines such as VGAE that cannot learn interpretable embeddings, our model also learns embeddings that can be easily interpreted as memberships of nodes into communities. The superior results of DGLFRM and DGLRFM-B demonstrate the benefit of our deep generative models. The significantly better results of these as compared to LFRM also show the benefit of endowing LFRM with a deep architecture with nonlinear decoder and nonlinear encoder. We also performed an experiment to investigate the model's ability to leverage node features. As expected, our model when using the features performs better compared to the case when it ignores features. This experiment is included in the supplementary section.

Qualitative Analysis on Learned Embeddings

To demonstrate the interpretable nature of the embeddings learned by our model, we generate a synthetic dataset with 100 nodes and 10 communities. The dataset is generated by fixing the ground-truth communities (by creating a binary vector for each node) such that some of the nodes belong to same communities. The adjacency matrix is then generated using a random bilinear function, which takes a ground-truth vector pair as the input, followed by the sigmoid opera-

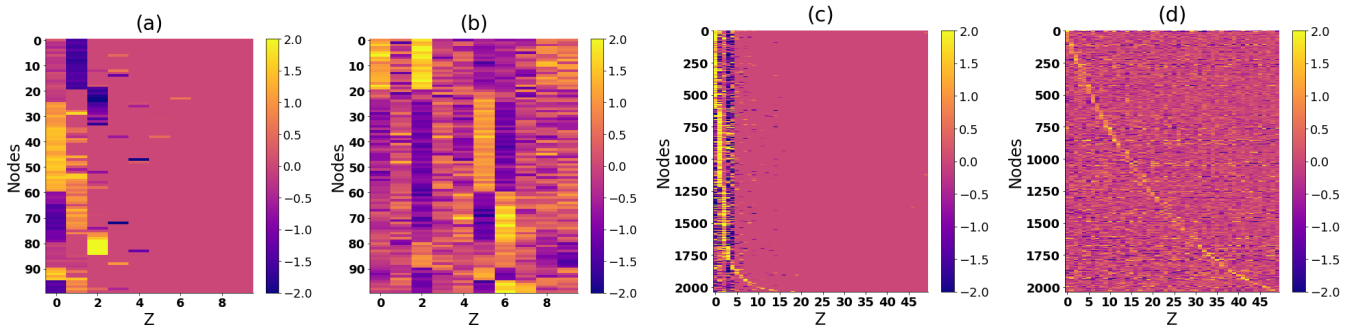

Figure 2: The latent structure (node-community associations) on synthetic data using (a) DGLFRM (b) VGAE. The latent structure on the NIPS12 using (c) DGLFRM (d) VGAE. In the latent structure plots for NIPS12, the communities are ordered to make community with more nodes have a smaller index.

tion. We use 85% of the synthetic adjacency matrix for link-prediction and visualize the latent structure that our model learns. In our experiments, we saw that the latent structure learned is in fact close to the ground-truth community assignments we started with. In Figure 2 (a), we plot the latent structure obtained using DGLFRM. As it can be seen, our model discovers the overlapping communities (Multiple communities, depicted by Z, are active for each node). By inspection, it can be seen that our model readily captures close to 10 overlapping-communities for nodes. We compare the community structure from our model with the latent structure learned by VGAE Figure 2 (b). Note that the Gaussian latent structure in VGAE is dense and therefore, fails to learn community memberships which are readily interpretable.

We also do qualitative analysis on the NIPS12 dataset Fig. 2 (c-d). Again we used 85% of the adjacency matrix with DGLFRM and VGAE. Table shows five of the inferred communities by DGLFRM. The authors shown under each community are ordered by the strength of their community memberships (in decreasing order). As Table shows, each of the communities represent a sub-field with authors working on similar topics. Moreover, note that some authors (e.g., Sejnowski T) are inferred as belonging to more than one community (which makes sense). This qualitative experiment demonstrates that our model can learn interpretable embeddings that can be used for tasks such as (overlapping) clustering. Note that our model can infer the number of communities naturally because of the stick-breaking prior. The stick-breaking prior requires specifying a large truncation level on the number of communities. Our model can effectively infer the "active" communities for a given truncation level. As shown in Fig. 2 (c), the posterior inference in our model is able to "turn off" the unnecessary columns in $\mathbf{Z}$. Although we do not know the ground truth for the number of communities, the number of inferred active communities is similar to what is reported in prior work on nonparametric Bayesian overlapping stochastic blockmodels (Miller, Griffiths, and Jordan 2009). Note that VGAE emeddings require an additional step (such as K-Means clustering) to cluster nodes. Moreover, a method such as K-means can not detect overlapping communities and is also sensitive to the initialization of K (estimated number of communities). For reference, we have included clustering results on the VGAE embeddings on NIPS12 data in the supplementary section.

Conclusion and Discussion

We have presented a deep generative framework for overlapping community discovery and link prediction. Our work combines the interpretability of stochastic blockmodels, such as the latent feature relational model, with the modeling power of deep generative models. Moreover, leveraging a nonparametric Bayesian prior on the node embeddings enables learning the node embedding size (i.e., the number of communities) from data. Our framework is fairly modular and a wide variety of decoder and encoder models can be used. In particular, it can leverage recent advances in non-probabilistic autoencoders for graphs, such as the graph convolutional network (Kipf and Welling 2016a) or its extensions (Hamilton, Ying, and Leskovec 2017). Inference in the model is based on SGVB which does not require conjugacy; this further widens the applicability of our framework to model different types of networks (e.g., weighted, count-valued edges, power-law degree distribution of node degrees, etc). We believe this combination of discrete latent variables based stochastic blockmodels and graph neural network will help leverage their respective strengths, and will fuel further research and advance the state-of-the-art in (deep) generative modeling of graph-structured data.

Although SGVB inference makes our model fairly efficient, it can be scaled up further by using mini-batch based inference. Another possibility to scale up the model is to replace the Bernoulli-logistic likelihood model by a Bernoulli-Poisson link (Zhou 2015), which enable scaling up the model in the number of nonzeros (i.e., number of edges) in the network. Given that our framework can work with a wide variety of decoder/generator models, such modifications can be done without much difficulty.

Finally, in this work we model each node as having a single binary vector denoting its memberships in one or more communities. Another interesting extension would be to consider multiple layers of latent variables, which can model a node's membership into a hierarchy of communities (Ho et al. 2011; Blundell and Teh 2013; Hu, Rai, and Carin 2017).

References

Airoldi, E. M.; Blei, D. M.; Fienberg, S. E.; and Xing, E. P. 2008. Mixed membership stochastic blockmodels. *JMLR*.

Blundell, C., and Teh, Y. W. 2013. Bayesian hierarchical community discovery. In *NIPS*.

Fortunato, S. 2010. Community detection in graphs. *Physics reports* 486(3):75–174.

Goldenberg, A.; Zheng, A. X.; Fienberg, S. E.; and Airoldi, E. M. 2010. A survey of statistical network models. *Foundations and Trends® in Machine Learning*.

Griffiths, T. L., and Ghahramani, Z. 2011. The indian buffet process: An introduction and review. *JMLR*.

Grover, A.; Zweig, A.; and Ermon, S. 2018. Graphite: Iterative generative modeling of graphs. *arXiv preprint arXiv:1803.10459*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NIPS*.

Ho, Q.; Parikh, A.; Song, L.; and Xing, E. 2011. Multiscale community blockmodel for network exploration. In *AISTATS*.

Hoff, P. D.; Raftery, A. E.; and Handcock, M. S. 2002. Latent space approaches to social network analysis. *JASA*.

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

Hoffman, M. D. 2014. Stochastic structured mean-field variational inference. *CoRR* abs/1404.4114.

Hu, C.; Rai, P.; and Carin, L. 2017. Deep generative models for relational data with side information. In *International Conference on Machine Learning*, 1578–1586.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.

Kemp, C.; Tenenbaum, J. B.; Griffiths, T. L.; Yamada, T.; and Ueda, N. 2006. Learning systems of concepts with an infinite relational model. In *Proceedings of the national conference on artificial intelligence*.

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

Kingma, D. P., and Welling, M. 2014. Efficient gradient-based inference through transformations between bayes nets and neural nets. *CoRR* abs/1402.0480.

Kipf, T. N., and Welling, M. 2016a. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kipf, T. N., and Welling, M. 2016b. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Latouche, P.; Birmelé, E.; and Ambroise, C. 2011. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*.

Miller, K.; Griffiths, T.; and Jordan, M. 2009. Nonparametric latent feature models for link prediction. *NIPS*.

Nalisnick, E., and Smyth, P. 2017. Stick-breaking variational autoencoders. In *ICLR*.

Nowicki, K., and Snijders, T. A. B. 2001. Estimation and prediction for stochastic blockstructures. *JASA*.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Schmidt, M. N., and Morup, M. 2013. Nonparametric bayesian modeling of complex networks: An introduction. *Signal Processing Magazine, IEEE* 30(3).

Teh, Y. W.; Grr, D.; and Ghahramani, Z. 2007. Stick-breaking construction for the indian buffet process. In *AISTATS*.

Wang, H.; Shi, X.; and Yeung, D.-Y. 2017. Relational deep learning: A deep latent variable model for link prediction. In *AAAI*, 2688–2694.

Yan, F.; Xu, Z.; and Qi, Y. 2011. Sparse matrix-variate gaussian process blockmodels for network modeling. In *UAI*.

Yang, J., and Leskovec, J. 2012. Community-affiliation graph model for overlapping network community detection. In *ICDM*.

Zhou, M. 2015. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*.

Zhu, J.; Song, J.; and Chen, B. 2016. Max-margin nonparametric latent feature models for link prediction. *arXiv preprint arXiv:1602.07428*.