Recurrent Dirichlet Belief Networks for Interpretable Dynamic Relational Data Modelling

Yaqiong Li[1] , **Xuhui Fan**[2*] , **Ling Chen**[1] , **Bin Li**[3] , **Zheng Yu**[4] and **Scott A. Sisson**[2]
[1]Centre for Artificial Intelligence, University of Technology Sydney
[2]School of Mathematics & Statistics, University of New South Wales, Sydney
[3]School of Computer Science, Fudan University
[4]Department of Electrical and Computer Engineering, University of Alberta
yaqiong.li@student.uts.edu.au, {xuhui.fan,scott.sisson}@unsw.edu.au, ling.chen@uts.edu.au

Abstract

The Dirichlet Belief Network (DirBN) has been recently proposed as a promising approach in learning interpretable deep latent representations for objects. In this work, we leverage its interpretable modelling architecture and propose a deep dynamic probabilistic framework – the Recurrent Dirichlet Belief Network (Recurrent-DBN) – to study interpretable hidden structures from dynamic relational data. The proposed Recurrent-DBN has the following merits: (1) it infers interpretable and organised hierarchical latent structures for objects within and across time steps; (2) it enables recurrent long-term temporal dependence modelling, which outperforms the one-order Markov descriptions in most of the dynamic probabilistic frameworks; (3) the computational cost scales to the number of positive links only. In addition, we develop a new inference strategy, which first upward-and-backward propagates latent counts and then downward-and-forward samples variables, to enable efficient Gibbs sampling for the Recurrent-DBN. We apply the Recurrent-DBN to dynamic relational data problems. The extensive experiment results on real-world data validate the advantages of the Recurrent-DBN over the state-of-the-art models in interpretable latent structure discovery and improved link prediction performance.

1 Introduction

Dynamic data is a common feature in many real-world applications, including relational data analysis [Mucha *et al.*, 2010; Phan and Airoldi, 2015; Yang and Koeppl, 2018] for learning time-varying node interactions, and text modelling [Guo *et al.*, 2018; Schein *et al.*, 2019] for exploring topic evolution. Modelling dynamic data has become a vibrant research topic, with popular techniques ranging from non-Bayesian methods, such as Collaborative Filtering with Temporal Dynamics (SVD++) [Koren, 2009], to Bayesian deep probabilistic frameworks such as Deep Poisson-Gamma Dynamical Systems (DPGDS) [Guo *et al.*, 2018]. The main advantage of Bayesian deep probabilistic frameworks is the flexible model design and the strong modelling performance. However, most of these frameworks are static so that they cannot account for the evolution of relationships over time. It would be highly beneficial if the frameworks can be extended to the dynamic setting to enjoy the modelling advantages.

The Dirichlet Belief Network (DirBN) [Zhao *et al.*, 2018] has been proposed recently as a promising deep probabilistic framework for learning *interpretable* deep latent structures. To date, the DirBN has mainly been used in two applications: (1) topic structure learning [Zhao *et al.*, 2018], where latent representations are used to model the word distribution for topics; and (2) relational models [Fan *et al.*, 2019a], where latent representations model the nodes' membership distribution over communities. By constructing a deep architecture for latent distributions, the DirBN can model high-order dependence between topic-word distributions (in topic models) and nodes' membership distributions (in relational models).

In this work, we propose a Recurrent Dirichlet Belief Network (Recurrent-DBN) to explore the complex latent structures in dynamic relational data. In addition to constructing an interpretable deep architecture for the data within individual time steps, we also study the temporal dependence in the dynamic relational data through (layer-to-layer) connections crossing consecutive time steps. Consequently, our Recurrent-DBN can describe long-term temporal dependence (i.e., the dependence between the current variables and those in the previous several time steps), improving over the one-order Markov structures that usually describe the dependence between the current variables and those in the previous one time step only.

For model inference, we further develop an efficient Gibbs sampling algorithm. Besides upward propagating latent counts as done by DirBN, we also introduce a backward step to propagate the counts from the current time step to the previous time steps. Our experiments on real-world dynamic relational data show significant advantages of the Recurrent-DBN over the state-of-the-art models in tasks of interpretable latent structure discovery and link prediction. Similar to DirBN that can be considered as a self-contained module [Zhao *et al.*, 2018], our Recurrent-DBN could be flexibly adapted to account for dynamic data other than evolving relational data,

[*] Corresponding Author

such as time-varying counts and dynamic drifting text data.

We summarise this paper's main merits as follows:

Model Recurrent structures are designed to model long term temporal dependence. Also, interpretable and organised latent structures are well explored;

Inference An efficient Gibbs sampling method is devised that first upward-backward propagates latent counts and then downward-forward samples variable;

Results Significantly improved model performance in real-world dynamic relational models compared to the state-of-the-art, including better link prediction performance and enhanced interpretable latent structure visualisation.

2 Background Information of DirBN

We first give a brief review of the DirBN model. In general, the DirBN constructs a *multi-stochastic* layered architecture to represent interpretable latent distributions for objects. We describe it within the relational data setting for illustrative purposes. Given a binary observed linkage matrix $\boldsymbol{R} \in \{0,1\}^{N\times N}$ for N nodes, where R_{ij} denotes whether node i has a relation to node j, the DirBN constructs an L-layer and K-length community membership distribution $\boldsymbol{\pi}_i = \{\boldsymbol{\pi}_i^{(l)}\}_{l=1}^L$ for each node i. The generative process for the membership distributions $\{\boldsymbol{\pi}_i^{(l)}\}_{l=1}^L$, as well as the observed matrix $\boldsymbol{R}$, can be briefly described as:

1. For $l = 1, \dots, L$
   (a) $\beta_{i'i}^{(l-1)} \sim \text{Gam}(c, \frac{1}{d}), \forall i, i' = 1, \dots, N$
   (b) $\boldsymbol{\pi}_i^{(l)} \sim \text{Dirichlet}(\boldsymbol{\alpha}^{1\times K}\mathbf{1}(l=1) + \sum_{i'} \beta_{i'i}^{(l-1)} \boldsymbol{\pi}_{i'}^{(l-1)})$
2. $\boldsymbol{X}_i \sim \text{Multinomial}(M; \boldsymbol{\pi}_i^{(L)}), \forall i = 1, \dots, N$;
3. $R_{ij} \sim \text{Bernoulli}\left(f(\boldsymbol{X}_i, \boldsymbol{X}_j)\right), \forall i, i' = 1, \dots, N$;

where $\boldsymbol{\alpha}^{1\times K}$ is a concentration parameter generating the membership distribution in the 1st layer, $\beta_{i'i}^{(l-1)}$ represents the information propagation coefficient from node i' to node i in the $(l-1)$th layer, c and d are the hyper-parameters generating these propagation coefficients, $\boldsymbol{X}_i$ is the latent count information for node i and M is the sum of these counts, and $f(\boldsymbol{X}_i, \boldsymbol{X}_j)$ represents the probabilistic function mapping a pair of membership distributions to a linkage probability. A larger value of $\beta_{i'i}^{(l-1)}$ indicates higher influence of $\boldsymbol{\pi}_{i'}^{(l-1)}$ on the generation of $\boldsymbol{\pi}_i^{(l)}$. Therefore, $\beta_{i'i}^{(l-1)}$ is set to 0 if node i' is not connected to node i in the observed data $\boldsymbol{R}$.

It is difficult to directly implement efficient Gibbs sampling for the DirBN because the prior and posterior distributions of the membership distributions $\boldsymbol{\pi}_i^{(l)}$ are not conjugate. To address this issue, a strategy of first upward propagating latent counts and then downward sampling variables has been developed in [Zhao *et al.*, 2018]. Given the count information $\boldsymbol{X}_i$ for node i, the DirBN upward propagates $\boldsymbol{X}_i$ to all the nodes in the $(L-1)$th layer through a Chinese Restaurant Table (CRT) distribution. Each node in the $(L-1)$th layer collects these propagated counts and uses their sum as its latent count $\boldsymbol{X}_i^{(L-1)}$ in the $(L-1)$th layer. This procedure is repeated until the counts have been assigned to all layers. Thus, conjugate constructions can be created for each variable and thereby used to construct efficient Gibbs samplers.

3 Recurrent-DBN for Dynamic Relational Data Modeling

To handle dynamic relational data, we attach an index t to variables to denote the corresponding time step. Thus, the observed dynamic relational data can be described as $\boldsymbol{R} \in \{0,1\}^{N\times N\times T}$ for N nodes at T time steps, where $R_{ij,t}$ denotes whether node i has relation to node j at the tth time step. Each matrix $\{\boldsymbol{R}_{-,t}\}_t \in \{0,1\}^{N\times N}$ can be either asymmetric (directional) or symmetric (non-directional) and we do not consider self-linkages $\{R_{ii,t}\}_{i,t}$.

3.1 Recurrent-DBN for Latent Structure Generation

In the Recurrent-DBN, we assume the time-dependent membership distribution of a node i in the l-th layer at time step t, $\boldsymbol{\pi}_{i,t}^{(l)}$, follows a Dirichlet distribution. Its generative process can be described as below, with the propagation of $\boldsymbol{\pi}_{i,t}^{(l)}$ illustrated in Fig. 1 (Left). For notation convenience, any parameters with index 0 are set to zero. It is noted that we have already used the observed data into the generative process.

1. For $t = 1, \dots, T, l = 1, \dots, L$
   (a) For $i', i = 1, \dots, N$
      i. $\beta_{i'i,t}^{(l-1)} \begin{cases} = 0, & \text{if } R_{i'i,t} = 0; \\ \sim \text{Gam}(c_c^{(l)}, \frac{1}{d_c}), & \text{if } i' = i; \\ \sim \text{Gam}(c_u^{(l)}, \frac{1}{d_c}), & \text{if } i' \neq i. \end{cases}$
      ii. $\gamma_{i'i,t-1}^{(l)} \begin{cases} = 0, & \text{if } R_{i'i,t-1} = 0; \\ \sim \text{Gam}(c_c^{(l)}, \frac{1}{d_c}), & \text{if } i' = i; \\ \sim \text{Gam}(c_u^{(l)}, \frac{1}{d_c}), & \text{if } i' \neq i. \end{cases}$
   (b) For $i = 1, \dots, N$
      i. Calculate concentration parameter $\boldsymbol{\psi}_{i,t}^{(l)}$:
      $$\boldsymbol{\psi}_{i,t}^{(l)} = \sum_{i'} \beta_{i'i,t}^{(l-1)} \boldsymbol{\pi}_{i',t}^{(l-1)} + \sum_{i'} \gamma_{i'i,t-1}^{(l)} \boldsymbol{\pi}_{i',t-1}^{(l)}. \quad (1)$$
      ii. $\boldsymbol{\pi}_{i,t}^{(l)} \sim \text{Dirichlet}(\boldsymbol{\alpha}^{1\times K}\mathbf{1}(t=1, l=1) + \boldsymbol{\psi}_{i,t}^{(l)})$.

Here, $\beta_{i'i,t}^{(l-1)} \in \mathbb{R}^+$ is the information propagation coefficient from node i' in the $(l-1)$-th layer to node i in the l-th layer at the same time t, $\gamma_{i'i,t-1}^{(l)} \in \mathbb{R}^+$ is the information propagation coefficient from node i' at time $t-1$ to node i at time t in the same layer l, $c_c^{(-)}, c_u^{(-)}, d_c$ are the corresponding hyper-parameters and $\boldsymbol{\alpha}^{1\times K}$ is the concentration parameter for the membership distribution in the first layer. The larger the value of these coefficients, the stronger the connections between the two corresponding latent representations (i.e., $\boldsymbol{\pi}_{i',t}^{(l-1)}$ and $\boldsymbol{\pi}_{i,t}^{(l)}$, $\boldsymbol{\pi}_{i',t-1}^{(l)}$ and $\boldsymbol{\pi}_{i,t}^{(l)}$).

We restrict the two nodes to have information propagated only if they are observed with positive relationship (step (a).i and (a).ii). This can reduce the computational cost of calculating $\boldsymbol{\beta}_{-,t}^{(l)}, \boldsymbol{\gamma}_{-,t}^{(l)}$ from $\mathcal{O}(N^2)$ to the scale of the number of

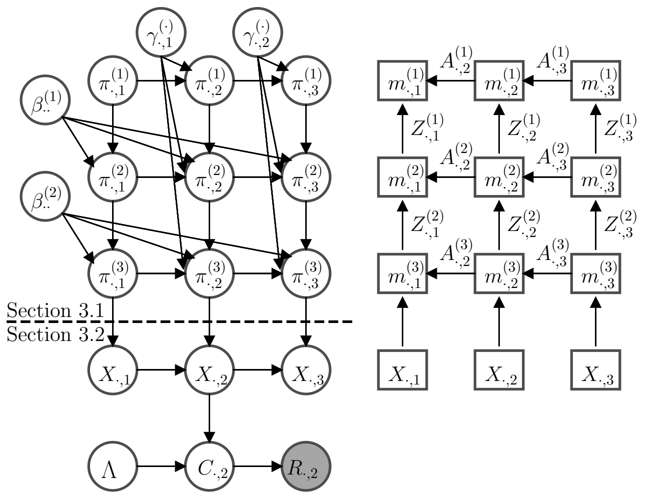

Figure 1: Left: a brief graphical model of Recurrent-DBN with 3-hidden-layers for a dynamic relational data with 3 time steps (Sections 3.1&3.2), where shaded nodes represent observed data. Hyper-parameters are ignored for concise presentation. Right: the upward-backward propagation of counts $\boldsymbol{X}$ to each hidden layers in each inference iteration (Section 3.3), where $\boldsymbol{m}^{(-)}_{-,-}$ represents the latent counts attached to nodes at each layer and time step, $\boldsymbol{Z}^{(-)}_{-,-}$ refers to the layer-wise propagated counts and $\boldsymbol{A}^{(-)}_{-,-}$ is the propagated counts between consecutive time steps.

positive relationships. Also, it encourages connected nodes to have more similar membership distributions and larger dependencies between each other.

The concentration parameter $\boldsymbol{\psi}^{(l)}_{i,t}$ for generating $\boldsymbol{\pi}^{(l)}_{i,t}$ comprises two parts: the information propagated from all other nodes' latent representations in the $(l-1)$-th layer at time t, $\sum_{i'}\beta^{(l-1)}_{i'i,t}\boldsymbol{\pi}^{(l-1)}_{i',t}$, and those in the l-th layer at time $(t-1)$, $\sum_{i'}\gamma^{(l)}_{i'i,t-1}\boldsymbol{\pi}^{(l)}_{i',t-1}$. In other words, $\boldsymbol{\psi}^{(l)}_{i,t}$ is a linear sum of all the previous-layers' information at the same time step and all the previous-time steps' information in the same layer. When the coefficients $\boldsymbol{\beta}$ dominate over $\boldsymbol{\gamma}$, the hierarchical structure plays a more important role. Otherwise, the temporal dependence has higher influence.

3.2 Application to Dynamic Relational Data

After generating the membership distributions $\{\boldsymbol{\pi}^{(l)}_{i,t}\}$, we use the Bernoulli-Poisson link function [Dunson and Herring, 2005; Zhou, 2015; Fan *et al.*, 2019a] to generate the relational data at each time step:

1. $\Lambda_{k_1k_2}\sim\text{Gamma}(\lambda_1,\lambda_0),\forall k_1,k_2$
2. $M_{i,t}\sim\text{Poisson}(M),\forall i,t$
3. $\boldsymbol{X}_{i,t}\sim\text{Multinomial}(M_{i,t};\boldsymbol{\pi}^{(L)}_{i,t}),\forall i,t;$
4. For $t=1,\ldots,T,i,j=1,\ldots,N,$
   (a) $C_{ij,k_1k_2,t}\sim\text{Poisson}(X_{i,k_1,t}\Lambda_{k_1k_2}X_{j,k_2,t}),\forall k_1,k_2$
   (b) $R_{ij,t}=\mathbf{1}(\sum_{k_1,k_2}C_{ij,k_1k_2,t}>0),$

where $\Lambda_{k_1k_2}$ is a community compatibility parameter such that a larger value of $\Lambda_{k_1k_2}$ indicates a larger possibility of generating the links between communities k_1 and k_2, λ_1,λ_0,M are hyper-parameters, $M_{i,t}$ is a scaling parameter for generating the related counting information for node i at time t, and $C_{ij,k_1k_2,t}$ is a community-to-community latent integer for linkage R_{ij} at time t.

Through the Multinomial distributions with $\boldsymbol{\pi}^{(L)}_{i,t}$ as event probabilities, $\boldsymbol{X}_{i,t}$ can be regarded as an estimator of $\boldsymbol{\pi}^{(L)}_{i,t}$. Since the sum $M_i\sim\text{Poisson}(M)$, according to the Poisson-Multinomial equivalence, each $X_{i,k,t}$ is equivalently distributed as $X_{i,k,t}\sim\text{Poisson}(M\pi^{(L)}_{i,k,t})$. Therefore, both the prior distribution for generating $X_{i,k,t}$ and the likelihood based on $X_{i,k,t}$ are Poisson distributions. We may form feasible categorical distribution on its posterior inference. This trick is inspired by the recent advances in data augmentation and marginalisation techniques [Fan *et al.*, 2019a], which allows us to implement posterior sampling for $X_{i,k,t}$ efficiently.

The counts $\boldsymbol{X}_{i,t}$ lead to the generation of the $K\times K$ integer matrix $\boldsymbol{C}_{ij,t}$. Based on the Bernoulli-Poisson link function [Dunson and Herring, 2005; Zhou, 2015], the observed $R_{ij,t}$ is mapped to the latent Poisson count random variable matrix $\boldsymbol{C}_{ij,t}$. It is shown in [Fan *et al.*, 2019a] that $\{C_{ij,k_1k_2,t}\}_{k_1,k_2}=0$ if $R_{ij,t}=0$. That is, only the non-zero links are involved during the inference for $\boldsymbol{C}_{ij,k_1k_2,t}$, which largely reduces the computational complexity, especially for large and sparse dynamic relational data.

Recurrent structure. Before describing the inference of Recurrent-DBN, we discuss the characteristic of the recurrent structure of our model. Instead of using the one-order Markov property to describe the temporal dependence (assuming the state at time t depends on the states at time $t-1$ only), which is adopted by most probabilistic dynamic models, the deep structure of the Recurrent-DBN allows the latent variables at time t depend on those at time steps from $t-1$ to $t-L$. For example, by using the law of total expectations, we can have the expectation of the latent count $\boldsymbol{X}_{-,t}$ in a 2-layered Recurrent-DBN as (We use the notation $-$ to denote a related parameter or variable hereafter):

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{X}_{-,t}|-\right]=&\boldsymbol{\beta}^{(L-1)}_{-,t-1}\boldsymbol{\gamma}^{(L-1)}_{-,t-1}\boldsymbol{\pi}^{(L)}_{-,t-1}\\&+\boldsymbol{\beta}^{(L-1)}_{-,t-1}\boldsymbol{\beta}^{(L-2)}_{-,t-1}\boldsymbol{\gamma}^{(L-2)}_{-,t-1}\boldsymbol{\gamma}^{(L-2)}_{-,t-2}\boldsymbol{\pi}^{(L-1)}_{-,t-2}.\end{aligned}\quad(2)$$

In Eq. (2), $\mathbb{E}\left[\boldsymbol{X}_{-,t}|-\right]$ depends on both $\boldsymbol{\pi}_{-,t-1}$ and $\boldsymbol{\pi}_{-,t-2}$. This format can be extended straightforwardly to L-layers and involve more previous membership distributions. Such recurrent structures allow us to summarise and abstract those random variables, capturing both the hierarchical latent structures and the dynamic dependencies.

3.3 Inference

The joint distribution of the latent variables is expressed as:

$$P(\boldsymbol{\pi},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{X},\boldsymbol{C},\boldsymbol{\Lambda},\boldsymbol{R}|-)=\prod_{i',i,t}P(R_{i',i,t}|\boldsymbol{C})P(\boldsymbol{\Lambda}|\lambda_1,\lambda_0)$$

$$\cdot\prod_{i,l,t}\left[P(\boldsymbol{\pi}^{(l)}_{i,t}|\boldsymbol{\pi}^{(l)}_{-,t-1},\boldsymbol{\pi}^{(l-1)}_{-,t},\boldsymbol{\beta}^{(l)}_{-i,t},\boldsymbol{\gamma}^{(l)}_{-i,t})\cdot P(\beta^{(l)}_{-i,t}|-)P(\gamma^{(l)}_{-i,t}|-)\right]$$

$$\cdot\prod_{i,t}\left[P(X_{i,t}|\boldsymbol{\pi}_{i,t},M)\prod_{i',k_1,k_2}P(C_{i'i,k_1k_2,t}|X_{i,t},X_{j,t},\Lambda_{k_1k_2})\right]$$

While the DirBN only has upward-propagation for the latent counts and downward-sampling for the latent variables, for the Recurrent-DBN we develop an upward-backward propagation and forward-downward Gibbs sampling algorithm for count propagation and latent variable sampling. Posterior simulation for the Recurrent-DBN involves two key steps in each sampling iteration: (1) propagating the counts $\boldsymbol{X}$ upward and backward to the upper layers and previous time steps via a latent count variable $\boldsymbol{m}$; (2) forward and downward sampling $\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ given the propagated latent counts $\boldsymbol{m}$. Full updates for the other variables are similar to those in [Fan *et al.*, 2019a].

Upward-Backward Propagating the Latent Counts

Figure 1 (right) illustrates the upward-backward propagation of counts $\boldsymbol{X}$ to the latent count variable $\boldsymbol{m}$ at each hidden layers. Generally speaking, for $i, i' = 1, \ldots, N, l = 1, \ldots, L, t = 1, \ldots, T, k = 1, \ldots, K$, the latent variable $\boldsymbol{\psi}$ is generated as Eq. (1). $m_{i,k,t}^{(l)}$ refers to the latent counts for the node i in layer l at time t for the k-th community. By integrating the $m_{i,k,t}^{(l)}$, the likelihood term of $\boldsymbol{\psi}_{i,t}^{(l)}$ can be calculated as:

$$\mathcal{L}(\boldsymbol{\psi}_{i,t}^{(l)}) \propto \frac{\Gamma(\sum_k \psi_{i,k,t}^{(l)})}{\Gamma(\sum_k \psi_{i,k,t}^{(l)} + \sum_k m_{i,k,t}^{(l)})} \prod_k \frac{\Gamma(\psi_{i,k,t}^{(l)} + m_{i,k,t}^{(l)})}{\Gamma(\psi_{i,k,t}^{(l)})}$$

where $\Gamma(-)$ is a Gamma function.

By introducing the auxiliary variables $q_{i,t}^{(l)}$ and $y_{i,k,t}^{(l)}$, the likelihood term of $\boldsymbol{\psi}_{i,t}^{(l)}$ can be further augmented as:

$$\mathcal{L}(\boldsymbol{\psi}_{i,t}^{(l)}, q_{i,t}^{(l)}, y_{i,k,t}^{(l)}) \propto \prod_{k=1}^{K} \left(q_{i,t}^{(l)}\right)^{\psi_{i,k,t}^{(l)}} \left(\psi_{i,k,t}^{(l)}\right)^{y_{i,k,t}^{(l)}}$$

where the $q_{i,t}^{(l)}$ and $y_{i,k,t}^{(l)}$ can be generated as:

$$y_{i,k,t}^{(l)} \sim \text{CRT}(m_{i,k,t}^{(l)}, \psi_{i,k,t}^{(l)}), q_{i,t}^{(l)} \sim \text{Beta}(\sum_k \psi_{i,k,t}^{(l)}, \sum_k m_{i,k,t}^{(l)})$$

Consequently, $y_{i,k,t}^{(l)}$ can be considered as the 'derived latent counts' for node i derived from the latent counts $m_{i,k,t}^{(l)}$. Each $y_{i,k,t}^{(l)}$ can then be upward and backward distributed based on the probabilities of $\psi_{i,k,t}^{(l)}$ as follows:

$$\begin{aligned}&(Z_{i1,k,t}^{(l-1)}, \ldots, Z_{iN,k,t}^{(l-1)}, A_{i1,t-1,k}^{(l)}, \ldots, A_{i1,t-1,k}^{(l)})\\ &\sim\text{Multinomial}(y_{i,k,t}^{(l)}; \frac{\boldsymbol{\beta}_{-i,t}^{(l-1)} \boldsymbol{\pi}_{-,k,t}^{(l-1)}}{\psi_{i,k,t}^{(l)}}, \frac{\boldsymbol{\gamma}_{-i,t-1}^{(l)} \boldsymbol{\pi}_{-,t-1,k}^{(l)}}{\psi_{i,k,t}^{(l)}}) \quad (3)\end{aligned}$$

Here, the $y_{i,k,t}^{(l)}$ is divided into two parts: one is delivered to each i' at time t of layer $l-1$ (($Z_{i1,k,t}^{(l-1)}, \ldots, Z_{iN,k,t}^{(l-1)}$)), and the other to each i' at time $t-1$ of layer l ($A_{i1,t-1,k}^{(l)}, \ldots, A_{i1,t-1,k}^{(l)}$)). We denote them as $\boldsymbol{Z}_{i-,k,t}^{(l-1)}$ and $\boldsymbol{A}_{i-,t-1,k}^{(l)}$ respectively. The latent counts of lower layers and previous time steps can thus be calculated respectively as:

$$\begin{aligned} m_{i,k,t}^{(l-1)} &= \sum_{i'} Z_{i'i,k,t}^{(l-1)} + \sum_{i'} A_{i'i,k,t}^{(l-1)} \\ m_{i,t-1,k}^{(l)} &= \sum_{i'} Z_{i'i,t-1,k}^{(l)} + \sum_{i'} A_{i'i,t-1,k}^{(l)} \quad (4)\end{aligned}$$

Let $\boldsymbol{m}_{i,T}^{(L)} = \boldsymbol{X}_{i,T}$, for $t = T-1, \ldots, 2, i, i' = 1, \ldots, N$, the specification in terms of layer L is as follows,

$$m_{i,t-1,k}^{(L)} = X_{i,t-1,k} + \sum_{i'} A_{i'i,t-1,k}^{(L)} \quad (5)$$

To summarize, upward and backward propagation derives $\boldsymbol{y}_{i,t}^{(l)}$ from the latent counts $\boldsymbol{m}_{i,t}^{(l)}$. Then, $\boldsymbol{y}_{i,t}^{(l)}$ is distributed to all i' at time t of layer $l-1$ and time $t-1$ of layer l respectively as $\boldsymbol{Z}_{i-,k,t}^{(l-1)}$ and $\boldsymbol{A}_{i-,t-1,k}^{(l)}$. Lastly, $\boldsymbol{Z}_{-i,k,t}^{(l-1)}$ and $\boldsymbol{A}_{-i,t-1,k}^{(l)}$ contribute to the generation of $\boldsymbol{m}_{i,t}^{(l-1)}$ and $\boldsymbol{m}_{i,t-1}^{(l)}$ respectively. By repeating this process through layers and crossing time steps, we propagate the $\boldsymbol{X}$ to the $\boldsymbol{m}^{(l)}$ upward and backward sequentially.

Forward-Downward Sampling Latent Variables

The generated $\boldsymbol{\psi}, \boldsymbol{q}, \boldsymbol{m}^{(l)}, (\boldsymbol{Z}, \boldsymbol{A})$ can enable to form closed Gibbs sampling algorithm for the following variables:

Sampling $\{\boldsymbol{\pi}_{i,t}^{(l)}\}_{i,t,l}$ After obtaining the latent counts $\boldsymbol{m}_{i,t}^{(l)}$ for each layer and each time step, the posterior inference of $\boldsymbol{\pi}_{i,t}^{(l)}$ can be proceeded as:

$$\boldsymbol{\pi}_{i,t}^{(l)} \sim \text{Dirichlet}(\boldsymbol{\psi}_{i,t}^{(l)} + \boldsymbol{m}_{i,t}^{(l)})$$

Sampling $\{\beta_{i'i,t}^{(l)}, \gamma_{i'i,t}^{(l)}\}_{i',i,l,t}$ The likelihood term of $\beta_{i'i,t}^{(l)}$ can be represented as:

$$\mathcal{L}(\beta_{i'i,t}^{(l)}) \propto e^{\log q_{i,t}^{(l)} \beta_{i'i,t}^{(l)}} \left(\beta_{i'i,t}^{(l)}\right)^{\sum_k Z_{i'i,k,t}^{(l)}}$$

The prior of $\beta_{i'i,t}^{(l)}, \beta_{i'i,t}^{(l)}$ is Gam($\gamma_i^{(l)}, \frac{1}{c^{(l)}}$). Their posterior distribution is

$$\begin{aligned}\beta_{i'i,t}^{(l)} &\sim\text{Gam}(\gamma_i^{(l)} + \sum_k Z_{i'i,k,t}^{(l)}, \frac{1}{c^{(l)} - \log q_{i',t}^{(l)}})\\ \gamma_{i'i,t}^{(l)} &\sim\text{Gam}(\gamma_i^{(l)} + \sum_k A_{i'i,k,t}^{(l)}, \frac{1}{c^{(l)} - \log q_{i',t}^{(l)}})\end{aligned}$$

4 Related Work

Several Bayesian deep probabilistic frameworks have been proposed to capture the temporal dependence in dynamic data [Gan *et al.*, 2015; Gong, 2017; Henao *et al.*, 2015]. The Deep Dynamic Sigmoid Belief Network [Gan *et al.*, 2015] sequentially stacks models of sigmoid belief networks and uses the binary-valued hidden variables to depict the log-range dynamic dependence. The Deep Dynamic Poisson Factor Analysis (DDPFA) [Gong, 2017] incorporates the Recurrent Neural Networks (RNN) into the Poisson Factor Analysis (PFA) to depict the long-range dynamic dependence. However, in DDPFA, the parameters in RNN and the latent variables in PFA are optimized separately. Poisson Gamma Dynamic Systems (PGDS) [Schein *et al.*, 2016] are developed to model the counting data through a "shallow" modelling strategy. Dynamic-PGDS (DPGDS) [Guo *et al.*, 2018] is probably the closest work to our approach. Compared with DPGDS, our Recurrent-DBN differs in three aspects: (1) our Recurrent-DBN generates normalized latent representations and thus

	AUC (mean and standard deviation)				
Model	Coleman	Mining reality	Hypertext	Infectious	Student net
MMSB	0.875 ± 0.013	0.883 ± 0.009	0.869 ± 0.008	0.969 ± 0.004	0.916 ± 0.001
T-MBM*	0.886 ± 0.012	0.863 ± 0.005	0.797 ± 0.009	0.833 ± 0.018	0.886 ± 0.015
fcMMSB*	0.909 ± 0.005	0.932 ± 0.006	0.909 ± 0.005	0.980 ± 0.002	0.958 ± 0.003
BPTF*	0.907 ± 0.003	0.923 ± 0.004	0.871 ± 0.006	0.845 ± 0.001	0.905 ± 0.011
DRGPM*	$\cdots$	0.935 ± 0.013	0.906 ± 0.002	0.988 ± 0.001	0.825 ± 0.004
CN	0.871 ± 0.008	0.863 ± 0.014	0.786 ± 0.016	0.889 ± 0.004	0.849 ± 0.019
SVD++*	$\cdots$	0.843 ± 0.016	0.725 ± 0.014	0.617 ± 0.001	$\cdots$
MNE	0.893 ± 0.004	0.823 ± 0.004	0.869 ± 0.008	0.898 ± 0.007	0.942 ± 0.001
DeepWalk	0.916 ± 0.008	0.762 ± 0.014	0.826 ± 0.015	0.915 ± 0.010	0.915 ± 0.008
Recurrent-DBN,K=30	$\mathbf{0.919} \pm 0.012$	$\mathbf{0.969} \pm 0.000$	$\mathbf{0.944} \pm 0.004$	$\mathbf{0.995} \pm 0.000$	$\mathbf{0.976} \pm 0.002$
Recurrent-DBN,K=20	0.909 ± 0.019	$\mathbf{0.965} \pm 0.001$	$\mathbf{0.932} \pm 0.003$	$\mathbf{0.995} \pm 0.000$	$\mathbf{0.971} \pm 0.002$
Recurrent-DBN,K=10	0.899 ± 0.011	$\mathbf{0.961} \pm 0.002$	$\mathbf{0.926} \pm 0.002$	$\mathbf{0.989} \pm 0.000$	$\mathbf{0.964} \pm 0.010$

	Precision (mean and standard deviation)				
Model	Coleman	Mining reality	Hypertext	Infectious	Student net
MMSB	0.289 ± 0.025	0.126 ± 0.009	0.121 ± 0.019	0.233 ± 0.065	0.238 ± 0.017
T-MBM*	0.199 ± 0.015	0.443 ± 0.016	0.142 ± 0.010	0.393 ± 0.065	0.168 ± 0.007
fcMMSB*	0.344 ± 0.017	0.835 ± 0.017	0.505 ± 0.012	0.326 ± 0.011	0.304 ± 0.007
BPTF*	0.385 ± 0.057	0.701 ± 0.013	0.297 ± 0.010	0.371 ± 0.016	0.309 ± 0.080
DRGPM*	$\cdots$	0.855 ± 0.007	$\mathbf{0.525} \pm 0.022$	0.226 ± 0.001	0.284 ± 0.017
CN	0.189 ± 0.035	0.426 ± 0.006	0.121 ± 0.009	0.333 ± 0.065	0.138 ± 0.017
SVD++*	$\cdots$	0.423 ± 0.026	0.135 ± 0.008	0.214 ± 0.016	$\cdots$
MNE	0.315 ± 0.018	0.269 ± 0.004	0.227 ± 0.014	0.262 ± 0.009	0.347 ± 0.037
DeepWalk	0.167 ± 0.068	0.191 ± 0.009	0.117 ± 0.015	0.252 ± 0.019	0.192 ± 0.054
Recurrent-DBN,K=30	$\mathbf{0.569} \pm 0.022$	$\mathbf{0.881} \pm 0.003$	0.509 ± 0.017	$\mathbf{0.543} \pm 0.022$	$\mathbf{0.373} \pm 0.016$
Recurrent-DBN,K=20	$\mathbf{0.476} \pm 0.081$	$\mathbf{0.869} \pm 0.003$	0.468 ± 0.013	$\mathbf{0.469} \pm 0.026$	$\mathbf{0.361} \pm 0.016$
Recurrent-DBN,K=10	$\mathbf{0.457} \pm 0.042$	0.853 ± 0.007	0.450 ± 0.014	0.369 ± 0.010	$\mathbf{0.356} \pm 0.016$

Table 1: Links prediction performance comparison. Note:* represents a dynamic model.

provides more interpretable structures; (2) the count information is propagated in a different way; (3) our Recurrent-DBN is devised in the setting of relational modelling, while DPGDS is for the topic modelling setting.

For modelling dynamic network data, many of the existing works are "shallow" probabilistic modelling. The dynamic Tensorial Mixed Membership Stochastic Block model (T-MBM) [Tarrés-Deulofeu *et al.*, 2019] and the Fragmentation Coagulation Based MMSB (fcMMSB) [Yu and Fan, 2020] combine the notable mixed-membership stochastic block model with a dynamic setting. The Bayesian Poisson Tensor Factorization (BPTF) [Schein *et al.*, 2015] and the Dependent Relational Gamma Process model (DRGPM) [Yang and Koeppl, 2018] are the representative works that use Poisson matrix factorization techniques to address dynamic counting data. There are also some models using the collaborative filtering techniques such as SVD++. Some methods are not developed for dynamic network data originally, but they have later been applied to the dynamic scenario, such as structure-based models like Common Neighbor (CN) [Newman, 2001], and network embedding models, including Scalable Multiplex Network Embedding (MNE) [Zhang *et al.*, 2018] and DeepWalk [Perozzi *et al.*, 2014]. It is noted that there is a recent trend in using the graphon theory [Lloyd *et al.*, 2012; Orbanz and Roy, 2014] to model the network data [Fan *et al.*, 2016; 2018b; 2018a; 2019b; 2020].

5 Experiments

We evaluate the performance of our proposed Recurrent-DBN on five real-world data sets, by comparing with nine baseline methods: Mixed Membership Stochastic Block model (MMSB) [Airoldi *et al.*, 2008], T-MBM, fcMMSB, BPTF, DRGPM, SVD++, CN, MNE and DeepWalk. Except MMSB, all of the other eight baseline models are implemented with the released code. For MMSB, we use Gibbs sampling for the inference of all variables.

5.1 Data Set and Experimental Setting

The real-world relational data sets used in this paper are: Coleman [Coleman, 1964], Mining Reality [Eagle and Pentland, 2006], Hypertext [Isella *et al.*, 2011], Infectious [Isella *et al.*, 2011] and Student Net [Fan *et al.*, 2014]. The summarized statistics are detailed in Table 2. For the hyper-parameters, we specify $M \sim \text{Gamma}(N, 1)$ for all data sets, $\{c_c^{(l)}, c_u^{(l)}\}_l, d, d_c$ and $\Lambda_{k1,k2}$ are all given $\text{Gamma}(1, 1)$ priors and $L = 3$. For MMSB, we set the membership distribution according to Dirichlet$(\mathbf{1}^{1\times K})$.

5.2 Link Prediction

For link prediction, we randomly extract a proportion of 10% of relational data entries (either links or non-links) at each time step as the test set. The remaining 90% is used for training. The test relational data are not used to construct the

Data set	N	T	N_E	$S\%$
Coleman	73	2	506	4.75
Mining reality	96	10	15580	16.9
Hypertext	113	10	6996	5.48
Infectious	410	10	7112	0.42
Student net	1005	11	62041	0.56

Table 2: Data set information. N is the number of nodes, T is the number of time steps, N_E is the number of positive links and $S\%$ is the ratio of the number of positive links to the total number of links.

information propagation matrix (i.e., we set $\{\beta_{i'i,t}^{(l)}\}_{l,t} = 0$ if $R_{i'i}$ is the testing data). We estimate the posterior mean of $e^{-\sum_{k_1,k_2} X_{i,k_1,t}\Lambda_{k_1k_2}X_{j,k_2,t}}$ as the linkage probability for each test data. These linkage probabilities are then used to calculate two evaluation metrics: the area under the curve of the receiver operating characteristic (AUC) and the precision-recall (precision). Higher values of AUC and precision indicate better model performance.

The detail results are shown in Table 1. We report the average evaluation results for each model over 16 runs. Each run uses 3000 MCMC iterations with the first 1500 discarded as burn-in. Overall, Recurrent-DBN outperforms the baseline models for both metrics on almost all data sets. As might be expected, the value of AUC and precision increase with higher model complexity of Recurrent-DBN (i.e., larger values of K). For the other methods, fcMMSB is competitive with DRGPM and outperforms the other baselines. However, they all perform worse than Recurrent-DBN, especially for data sets with large numbers of N or T. We can see that the Recurrent-DBN has clear advantages in learning dynamic relational data, thanks to the deep hierarchical structure and recurrent long-term temporal dependence modelling.

5.3 Latent Variable Visualization

To gain further insights, we visualize the latent variables in Figure 2. It can be observed from the top part that: (1) for the same time step, the membership distributions change gradually with the increase of layers; (2) the membership distributions share some similarities for consecutive time steps and the similarities slowly shift along with the time. For example, the left bottom area of $\{\boldsymbol{\pi}_{i=1:30}^{(3)}\}$ seems to have 3 different patterns: time steps $t = 1$, $t = 2 \sim 4$, and $t = 5 \sim 10$. The bottom part of Figure 2 visualizes propagation coefficients. It is reasonable to see the values of $\overline{\boldsymbol{\beta}}$ in the first layer and $\overline{\boldsymbol{\gamma}}$ in the first several time steps are small, since less information is propagated in these cases. The values become larger when more information is propagated. Also, the layer-wise propagation seems to have a larger influence than the cross-time propagation, with an average value of $\overline{\boldsymbol{\beta}}/\overline{\boldsymbol{\gamma}} = 1.2 \sim 1.4$.

6 Conclusion

We have presented a probabilistic deep hierarchical structure named Recurrent Dirichlet Belief Networks (Recurrent-DBN) for learning dynamic relational data. Through Recurrent-DBN, the evolution of the latent structure is characterized by both the cross-layer and the cross-time dependencies. We also develop an upward-backward–forward-downward information propagation to enable efficient Gibbs sampling for all variables. The experimental results on a variety of real data sets demonstrate the excellent predictive performance of our model, and the inferred latent structure provides a rich interpretation for both hierarchical and dynamic information propagation. Our Recurrent-DBN can be applied to tasks like dynamic topic models [Guo *et al.*, 2018; Zhao *et al.*, 2018]) and dynamic collaborative filtering. We keep these potential applications as the future work.

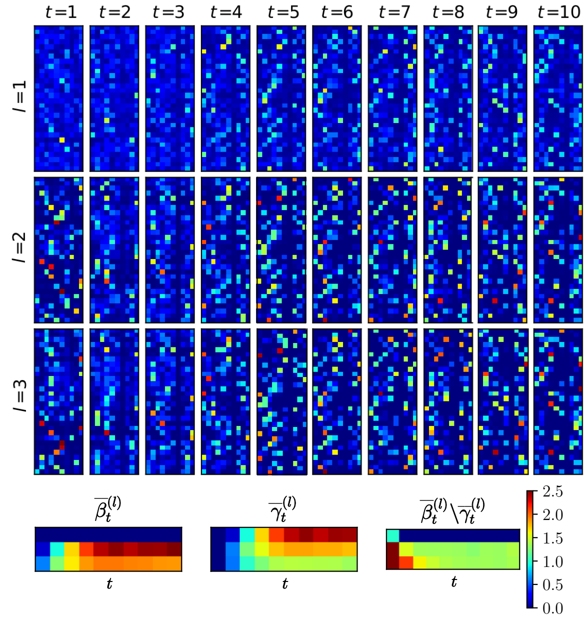

Figure 2: Top: visualizations of the membership distributions ($\{\boldsymbol{\pi}_{i=1:30}^{(l)}\}_{l=1}^{3}$) for the *Infectious* data set. Rows represent the nodes and columns represent the communities (with $K = 10$); Bottom: visualizations of average propagation coefficients $\overline{\boldsymbol{\beta}}_t^{(l)}, \overline{\boldsymbol{\gamma}}_t^{(l)}$ and their ratio. $\overline{\boldsymbol{\beta}}_t^{(l)}, \overline{\boldsymbol{\gamma}}_t^{(l)}$ are re-scaled for visualization convenience.

Acknowledgements

This work is partly supported by ARC Discovery Project DP180100966. Yaqiong Li is a recipient of UTS Research Excellence Scholarship. Xuhui Fan and Scott A. Sisson are supported by the Australian Research Council through the Australian Centre of Excellence in Mathematical and Statistical Frontiers (ACEMS, CE140100049). Bin Li is supported by Shanghai Municipal Science & Technology Commission (16JC1420401) and the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

References

[Airoldi *et al.*, 2008] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic block models. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

[Coleman, 1964] James S. Coleman. Introduction to mathematical sociology. *Introduction to mathematical sociology*, 1964.

[Dunson and Herring, 2005] David B. Dunson and Amy H. Herring. Bayesian latent variable models for mixed discrete outcomes. *Biostatistics*, 6(1):11–25, 2005.

[Eagle and Pentland, 2006] Nathan Eagle and Alex Sandy Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[Fan *et al.*, 2014] Xuhui Fan, Longbing Cao, and Richard Yi Da Xu. Dynamic infinite mixed-membership stochastic block model. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2072–2085, 2014.

[Fan *et al.*, 2016] Xuhui Fan, Bin Li, Yi Wang, Yang Wang, and Fang Chen. The Ostomachion Process. In *AAAI*, pages 1547–1553, 2016.

[Fan *et al.*, 2018a] Xuhui Fan, Bin Li, and Scott Sisson. Rectangular bounding process. In *NeurIPS*, pages 7631–7641, 2018.

[Fan *et al.*, 2018b] Xuhui Fan, Bin Li, and Scott A. Sisson. The binary space partitioning-tree process. In *AISTATS*, volume 84, pages 1859–1867, 2018.

[Fan *et al.*, 2019a] Xuhui Fan, Bin Li, Caoyuan Li, Scott SIsson, and Ling Chen. Scalable deep generative relational model with high-order node dependence. In *NeurIPS*, pages 12637–12647, 2019.

[Fan *et al.*, 2019b] Xuhui Fan, Bin Li, and Scott A. Sisson. The binary space partitioning forests. In *AISTATS*, volume 89, pages 3022–3031, 2019.

[Fan *et al.*, 2020] Xuhui Fan, Bin Li, and Scott A. Sisson. Online binary space partitioning forests. In *AISTATS*, 2020.

[Gan *et al.*, 2015] Zhe Gan, Chunyuan Li, Ricardo Henao, David E Carlson, and Lawrence Carin. Deep temporal sigmoid belief networks for sequence modeling. In *NeurIPS*, pages 2467–2475, 2015.

[Gong, 2017] Chengyue Gong. Deep dynamic poisson factorization model. In *NeurIPS*, pages 1666–1674, 2017.

[Guo *et al.*, 2018] Dandan Guo, Bo Chen, Hao Zhang, and Mingyuan Zhou. Deep poisson gamma dynamical systems. In *NeurIPS*, pages 8442–8452, 2018.

[Henao *et al.*, 2015] Ricardo Henao, Zhe Gan, James Lu, and Lawrence Carin. Deep poisson factor modeling. In *NeurIPS*, pages 2800–2808, 2015.

[Isella *et al.*, 2011] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, 2011.

[Koren, 2009] Yehuda Koren. Collaborative filtering with temporal dynamics. In *KDD*, pages 447–456, 2009.

[Lloyd *et al.*, 2012] 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*, pages 1007–1015, 2012.

[Mucha *et al.*, 2010] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[Newman, 2001] Mark EJ. Newman. Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102, 2001.

[Orbanz and Roy, 2014] Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):437–461, 2014.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[Phan and Airoldi, 2015] Tuan Q. Phan and Edoardo M. Airoldi. A natural experiment of social network formation and dynamics. *Proceedings of the National Academy of Sciences*, 112(21):6595–6600, 2015.

[Schein *et al.*, 2015] Aaron Schein, John Paisley, David M. Blei, and Hanna Wallach. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *KDD*, pages 1045–1054, 2015.

[Schein *et al.*, 2016] Aaron Schein, Hanna Wallach, and Mingyuan Zhou. Poisson-gamma dynamical systems. In *NeurIPS*, pages 5005–5013, 2016.

[Schein *et al.*, 2019] Aaron Schein, Scott Linderman, Mingyuan Zhou, David Blei, and Hanna Wallach. Poisson-randomized gamma dynamical systems. In *NeurIPS*, pages 781–792, 2019.

[Tarrés-Deulofeu *et al.*, 2019] Marc Tarrés-Deulofeu, Antonia Godoy-Lorite, Roger Guimerà, and Marta Sales-Pardo. Tensorial and bipartite block models for link prediction in layered networks and temporal networks. *Physical Review E*, 99(3):032307, 2019.

[Yang and Koeppl, 2018] Sikun Yang and Heinz Koeppl. Dependent relational gamma process models for longitudinal networks. In *ICML*, pages 5547–5556, 2018.

[Yu and Fan, 2020] Zheng Yu and Xuhui Fan. Fragmentation coagulation based mixed-membership stochastic blockmodel. *Arxiv*, 2020.

[Zhang *et al.*, 2018] Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In *IJCAI*, volume 18, pages 3082–3088, 2018.

[Zhao *et al.*, 2018] He Zhao, Lan Du, Wray Buntine, and Mingyuan Zhou. Dirichlet belief networks for topic structure learning. In *NeurIPS*, pages 7955–7966, 2018.

[Zhou, 2015] Mingyuan Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, pages 1135–1143, 2015.