Going Deeper and Wider: Unbounded Dirichlet Belief Networks

Abstract

The Dirichlet Belief Network (DBN) is recently proposed as a promising deep generative model to learn interpretable deep latent distributions for objects. However, the modelling capability of DBN is usually limited in two aspects: (1), DBN can only have a few number of layers since the observational information passing to next higher layer would decays in a $\mathcal{O}(\log)$ rate; (2), the lengths of latent distributions in all the layers are restricted to be the same and need to be fixed in advance. In this work, we propose Unbounded Dirichlet Belief Networks (U-DBN) to systematically address these two issues and thus enlarge the modelling capability. By inserting auxiliary Poisson random variables into the layerwise connections and appropriate design, our U-DBN can build deep archicture with arbitrary number of layers and make the latent distributions unbounded at the same time. We apply this U-DBN in the relational application and obtain competitive performance.

1. Introduction

The Dirichlet Belief Network (DBN) (Zhao et al., 2018) is an interesting approach in learning easy interpretable and meaningful deep latent distributions for objects. In comparison to existing deep generative models (e.g. Variational AutoEncoding (VAE) (), Generative Adversarial Networks (GAN) ()), which usually use parameterized functions to build deep architecture, DBN uses a multi-stochastic-layers () structure to complete the task. The latent distributions in each layer of DBN are generated as Dirichlet random variables and can thus be interpreted as categorical distributions. DBN can be correspondingly used for various situations, including Latent Dirichlet Allocation (Zhao et al., 2018), where the latent distribution is used to represent the word distribution for topics, and relational modelling (**?**), where the latent distribution refers to the nodes' membership distribution on communities.

. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

However, the current formulation of DBN suffers from two issues: (1), DBN's deep architecture is currently limited to only a few layers. In order to obtain efficient Gibbs sampling, DBN back-propagates the observed information from the output layer to each hidden layer. The cost of the information back-propagation is that the information would decay in a $\mathcal{O}(\log)$ rate on passing through one layer to its upper layer (Zhou et al., 2016). Therefore, little information might be available after a few layer back-propagations; (2), DBN assumes the lengths of all the latent distributions are the same and the value needs to be fixed in advance. Practise in deep learning might require flexible adjustment on the lengths (e.g. coarse-to-fine setting). Furthermore, appropriate value settings on the lengths of latent distributions obviously need careful and time consuming parameter tuning.

In this work, we try to systematically address these two issues. We name our proposed framework as Unbounded-Dirichlet Belief Network (U-DBN) and enlarge DBN's modelling capability from the number of layers and the length of latent distributions. One of our key contributions is the introduction of auxiliary Poisson random variables in the layerwise connection. These auxiliary variables are appropriately designed to facilitate the model inference. On one hand, they can be well organized to play as approximators of the latent distribution. On another hand, these auxiliary variables can circumvent the complicated bottom-up back-propagation way and enable full Gibbs sampling on all the latent variables. As a result, we may set arbitrary number of layers in the deep architecture of DBN.

Furthermore, we impose stick-breaking process prior on the length of latent distributions. The auxiliary variables, which can be regarded as approximators of the upper layers' latent distributions, are incorporated into the stick-breaking process to generate the unbounded latent distributions in the current layers. Since the counts incorporated in the construction are not actual output from the stick-breaking process, we name it as Pseudo-Counts-Stick-Breaking process. It enables the membership inheritance in latetn representation cross nearby layers and enable unbounded length of latent distributions at the same time.

We demonstrate the modelling advantages of our U-DBN in the applications of topic modelling and relational modelling.

2. Preliminary knowledge of Dirichlet Belief Networks

The Dirichlet Belief Networks (DBN)(Zhao et al., 2018) is proposed as an variant of the Gamma Belief Networks (GBN) (Zhou et al., 2016). When the GBN uses the shape parameter in the Gamma distribution to propagate information through different layers, the DBN chooses to propagate the information through the concentration parameters in the Dirichlet distribution. As a result, the latent distributions for objects in the DBN is restricted to be normalized and can be explained as distributions over latent structures.

More formally, $\forall l \in \{1, \dots, L-1\}$, the information propagation from l-th layer to the $l+1$-th layer $l+1$ will base on a linear combinations of latent distributions:

$$\boldsymbol{\pi}_i^{(l+1)} \sim \text{Dirichlet}(\sum_{i'} \beta_{i'i}^{(l)} \boldsymbol{\pi}_{i'}^{(l)}), \forall i = 1, \dots, N \tag{1}$$

where $\beta_{i'i}^{(l)} \in \mathbb{R}^+$, representing the propagation coefficient from node i' to node i at the l-th layer. In this way, we may be able describe high-order dependence between the nodes (**?**).

Due to the inconjugate property between these latent distributions, direct efficient Gibbs sampling over these random variables is difficult. Instead, (Zhao et al., 2018)(Zhou et al., 2016) choose to first back propagates the observed counting information into each layer and then proceed forward variable Gibbs sampling. That is to say, for latent counts $X_{ik}^{(l+1)}$, they would use Chinese Restaurant Table (CRT) distribution to generate the counts information to the l-th layer through $X_{ik}^{(l)} \sim \text{CRT}(X_{ik}^{(l+1)}, \sum_{i'} \beta_{i'i}^{(l)} \boldsymbol{\pi}_{i'}^{(l)}))$. As the CRT distribution counts the number of clusters only, the information decays in a $\mathcal{O}(\log)$ rate ().

3. Unbounded-Dirichlet Belief Networks

We assume there are N objects to be modelled in this work, where $\boldsymbol{\pi}_i^{(l)}$ is used to denote the i-th latent distribution in the l-th layer. There are L hidden layers in the deep architecture. $\pi_i^{(l)}$ may have different lengths $K^{(l)}$ across different layers and it can be explained as node i's distribution over $K^{(l)}$ features. Since $\boldsymbol{\pi}_i^{(l)}$ can also be explained as the distribution over latent features, the observation is generated from the Multinomial counts vector $\boldsymbol{X}_i$ (with $\boldsymbol{\pi}_i^{(l)}$ as event probabilities).

3.1. Going Deeper

In U-DBN, we do not use the the backward counts propagation and forward variable sampling strategy for model inference. Instead, we introduce layers of counts $\boldsymbol{C}$ into the layerwise Dirichlet-Dirichlet connections as:

$$\boldsymbol{C}_{i\cdot\cdot}^{(l)} \sim \text{Multinomial}\left(M^{(l)}; \boldsymbol{\pi}_{i\cdot}^{(l)} \bigotimes \boldsymbol{w}_{i\cdot}^{(l)}\right), \forall i \tag{2}$$

$$\boldsymbol{\pi}_i^{(l+1)} \sim \text{Dirichlet}(\alpha_d + \sum_{i'} \boldsymbol{C}_{i'\cdot i}^{(l)}), \alpha_d > 0, \forall i \tag{3}$$

where $M^{(l)} \sim \text{Poisson}(M)$ and where $\boldsymbol{\omega}_{i\cdot}^{(l)} \sim \text{Dirichlet}(\boldsymbol{\alpha}_\omega)$ represents the representation sharing parameters. Since $\sum_k \omega_{ik}^{(l)} = 1$, we can further get $\sum_{i'} \boldsymbol{C}_{i\cdot i'} \sim \text{Multinomial}(M^{(l)}; \boldsymbol{\pi}_i^{(l)})$. That is to say, the the normalized proportion vector of $\sum_{i'} \boldsymbol{C}_{i\cdot i'}$ can be regarded as an approximator of $\boldsymbol{\pi}_i^{(l)}$. In Eq. (3), we re-organize the counting variables and calculate the "received" counts information for each node i. For any node i, it will receive latent counts from all the nodes in this layer. These received counts can then be used to consistute the concentration parameter vector of the receiving node i's Dirichlet distribution.

The number $M_i^{(l)}$ of events records the total intensity of relating the latent representation to the related counts. Larger values of $M_i^{(l)}$ will remind a closer approximation to $\boldsymbol{\pi}_i^{(l)}$. Further, the introduction of Poisson counts help to decompose the additive effect of the weights, which is quite difficult to have closed Gibbs sampling format.

This key contribution needs to be emphasized again. The introduction of the counts variable $\boldsymbol{C}$ enables us to use a layerwise sampling method for all the variables and thus avoid the complicated strategy of backward counts propagation and forward variable sampling. $\boldsymbol{C}_i^{(l)}$ plays the role of "likelihood" variable for each latent distribution $\boldsymbol{\pi}_i^{(l)}$ and we can thus easily obtain Gibbs sampling for $\boldsymbol{\pi}_i^{(l)}$. Based on the Poisson-Multinomial equivalence, each inserted count variable $C_{iki'}^{(l)}$ follows a Poisson distribution as $C_{iki'}^{(l)} \sim \text{Poisson}(M\pi_{ik}^{(l)}\omega_{ii'}^{(l)})$. Combining its likelihood, we can easily obtain all its potential posterior proportional values.

3.2. Going Wider

Given these latent counts $\boldsymbol{C}^{(l)}$, we can further obtain infinite length extensions for the latent distributions. Instead of the Dirichlet distribution, we introduce a counts-based-Stick-Breaking process to generate the latent distributions as:

$$V_{ik}^{(l+1)} \sim \text{Beta}(1 + \sum_{i'} C_{i'ki}^{(l)}, \alpha + \sum_{i', k'=k+1} C_{i'k'i}^{(l)}), \tag{4}$$

$$\pi_{ik}^{(l+1)} = V_{ik}^{(l+1)} \prod_{k'=1}^{k-1} (1 - V_{ik'}^{(l+1)}), \forall i, k \tag{5}$$

When $\boldsymbol{C}_{i\cdot\cdot}^{(l+1)}$ is the actual observations for $\pi_i^{(l+1)}$, we let $\boldsymbol{C}_{\cdot\cdot i}^{(l)}$ play the similar roles. In this way, the generated $\pi_i^{(l+1)}$

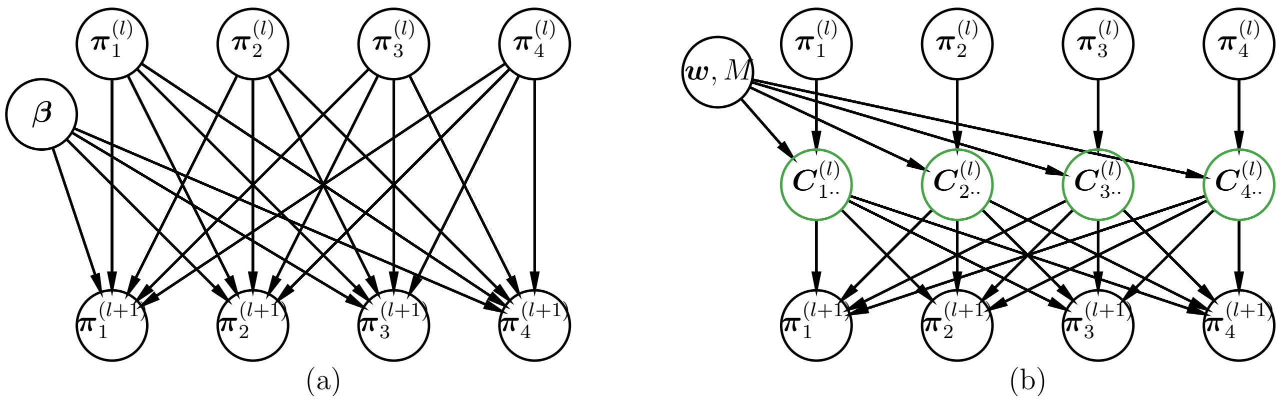

Figure 1. Graphical Models of (a) the Dirichlet Belief Networks and (b) our Unbounded Dirichlet Belief Networks.

can approximate to the weights of $\boldsymbol{C}_{\cdot\cdot i}^{(l)}$, which presents to be a inheritance, and have unbounded length at the same time.

The lengths of latent distributions can help us automatically determine the representation power in each layer in the deep architecture.

Combining with the stick-breaking process on generating $\boldsymbol{\pi}_i^{(l)}$, each element of $V_{ik}^{(l)}$ can be generated in a conjugate way. In this way, we do not need to engage the complicate bottom-up way to back propagate the counts to each layer. More importantly, we won't have $\mathcal{O}(\log^{L-l})$ scale counting information for the "observations" in the l-th layer. Arbitrary number of layers can be placed in the deep architecture.

3.3. Unbounded-DBN

In our U-DBN, we use the following constructions to replace the way of information propagation (Eq. (1)):

$$\boldsymbol{C}_{iki'} \sim \text{Poisson}\,(M^{(l)}\boldsymbol{\pi}_{ik}^{(l)}\boldsymbol{w}_{ii'}^{(l)}), \forall i, i', k \tag{6}$$

$$V_{ik}^{(l+1)} \sim \text{Beta}(1+\sum_{i'} C_{i'ki}^{(l)}, \alpha + \sum_{i',k'=k+1} C_{i'k'i}^{(l)}), \tag{7}$$

$$\pi_{ik}^{(l+1)} = V_{ik}^{(l+1)} \prod_{k'=1}^{k-1}(1-V_{ik'}^{(l+1)}), \forall i, k \tag{8}$$

where $\boldsymbol{w}_{i\cdot} \sim \text{Dirichlet}(\mathbf{1}_{1\times N})$, $C_{i'ki}^{(l)}$ represents the count value propagates from node i' to node i for the k-th latent features in the l-th layer. Figure. 1 displays the graphical models of these random variables.

From Eq. (6)(8) and Figure. 1, it is easy to see that we have inserted the auxiliary counting variables into the layerwise connections.

3.4. Inference

Sampling $\boldsymbol{C}_i^{(l)}$ Each element $C_{iki'}^{(l)}$ is distributed as $C_{iki'}^{(l)} \sim \text{Poisson}(M\pi_{ik}^{(l-1)}w_{ii'})$, which is

$$P(C_{iki'}^{(l)}|\cdots) \propto \frac{\left(M\pi_{ik}^{(l-1)}w_{ii'}\right)^{C_{iki'}^{(l)}}}{C_{iki'}^{(l)}!} \tag{9}$$

The likelihood term relates to $C_{iki'}^{(l)}$ are:

$$\begin{aligned} &P(V_{i'1}^{(l)}, \ldots, V_{i'k}^{(l)}|C_{iki'}^{(l)}, \ldots) \propto \left[\pi_{i'k}^{(l)}\right]^{C_{iki'}^{(l)}} \\ &\cdot \frac{\Gamma(1+\alpha+\sum_{i'',k} C_{i''ki'}^{(l)})}{\Gamma(1+\sum_{i''} C_{i''ki'}^{(l)})} \prod_{k'=2}^{k}(\alpha + \sum_{i'',k''=k'} C_{i''ki'}^{(l)}) \end{aligned} \tag{10}$$

The posterior distribution of $C_{iki'}^{(l)}$ is:

$$\begin{aligned} &P(C_{iki'}^{(l)}|\cdots) \propto \frac{\left(M\pi_{ik}^{(l-1)}w_{ii'}\pi_{i'k}^{(l)}\right)^{C_{iki'}^{(l)}}}{C_{iki'}^{(l)}!} \\ &\cdot \frac{\Gamma(1+\alpha+\sum_{i'',k} C_{i''ki'}^{(l)})}{\Gamma(1+\sum_{i''} C_{i''ki'}^{(l)})} \cdot \prod_{k'=2}^{k}(\alpha + \sum_{i'',k''=k'} C_{i''ki'}^{(l)}) \end{aligned} \tag{11}$$

Using the ratio test for Eq. (11), it is easy to see that this series converges.

$$\begin{aligned} &\lim_{n\to\infty}\left(\frac{\prod_{l=0}^{C^*}(n+1+l)}{(n+1)!}\right) \Big/ \left(\frac{\prod_{l=0}^{C^*}(n+l)}{(n)!}\right) \\ &= \lim_{n\to\infty}\frac{\prod_{l=0}^{C^*}(1+\frac{1}{n+l})}{n+1} = 0 \end{aligned} \tag{12}$$

where $C^* = \sum_{k'\neq k} C_{k'} - 1$.

Sampling $\boldsymbol{\pi}_i^{(l)}$ $\boldsymbol{\pi}_i^{(l)}$'s posterior distribution is a Dirichlet distribution as

$$V_{ik}^{(l)} \sim \text{Beta}\left(1 + \sum_{i'} C_{i'ki}^{(l)} + \sum_{i'} C_{iki'}^{(l+1)}, \right.$$
$$\left. \alpha + \sum_{k'=k+1}^{K} (\sum_{i'} C_{i'k'i}^{(l)} + \sum_{i'} C_{ik'i'}^{(l+1)}) \right) \quad (13)$$

where $\pi_{ik}^{(l)} = V_{ik}^{(l)} \prod_{k'=1}^{k-1}(1 - V_{ik'}^{(l)})$.

Sampling $\boldsymbol{w}$ Let $\boldsymbol{w}_{i\cdot} \sim \text{Dirichlet}(\mathbf{1}_{1\times N})$, the posterior distribution of $\boldsymbol{w}_{i\cdot}$ is obtained as:

$$\boldsymbol{w}_{i\cdot} \sim \text{Dirichlet}(1 + \sum_{i',l,k'} \boldsymbol{C}_{ik'i'}^{(l)}) \quad (14)$$

4. Application of U-DBN to relational modelling

4.1. Model

Our U-DBN is first applied to the relational modelling setting (**?**), which we named it as U-DBN-RM. The relational data $\boldsymbol{R}$ is represented as a binary matrix $\boldsymbol{R} \in \{0,1\}^{N\times N}$, where N is the number of nodes and the element R_{ij} $(\forall i, j)$ indicates whether node i relates to node j ($R_{ij} = 1$ if the relation exists, otherwise $R_{ij} = 0$), with the self-connection relation R_{ii} not considered here. The matrix $\boldsymbol{R}$ can be symmetric (i.e. undirected) or asymmetric (i.e. directed). The network's feature information is denoted by a non-negative matrix $\boldsymbol{F} \in \{\mathbb{R}^+ \cup 0\}^{N\times D}$, where D denotes the number of features, and where each element F_{id} $(\forall i, d)$ takes the value of the d-th feature for the i-th node.

The generative of U-DBN-RM is as follows:

1. $C_{ikd}^F \sim \text{Poisson}(M^{(0)} F_{id} T_{dk}), \forall i, d, k, \boldsymbol{\pi}_i^{(1)} \sim \text{Dirichlet}(\sum_d \boldsymbol{C}_{i\cdot d}^F), \forall i$
2. For $l = 2, \ldots, L-1; i, i' = 1, \cdots, N, k = 1, \ldots$
   (a) $\boldsymbol{C}_i^{(l)} \in \mathbb{N}^{K\times N} \sim \text{Poisson}\left(M^{(l)} \boldsymbol{\pi}_i^{(l)} \boldsymbol{w}_{i\cdot}\right)$
   (b) $V_{ik}^{(l+1)} \sim \text{Beta}(1 + \sum_{i'} \boldsymbol{C}_{i'ki}^{(l)}, \alpha + \sum_{k'=k+1}^{K} \sum_{i'} \boldsymbol{C}_{i'k'i}^{(l)}), \pi_{ik}^{(l+1)} = V_{ik}^{(l+1)} \prod_{k'=1}^{k-1}(1 - V_{ik'}^{(l+1)})$
3. $\forall i, j = 1, \cdots, N$
   (a) $\forall k_1, k_2, Z_{ij,k_1k_2} \sim \text{Poisson}\left((\sum_{i'} C_{ik_1i'}^{(L)}) \Lambda_{k_1k_2} (\sum_{j'} C_{jk_2j'}^{(L)})\right)$
   (b) $R_{ij} = \mathbf{1}\left(\sum_{k_1,k_2} Z_{ij,k_1k_2} > 0\right)$

where $T_{dk} \in \mathbb{R}^+$ is a feature-to-community transition coefficient, transfering the d-th feature into the k-th community; C_{ikd}^F is a feature-to-community counts variable (which presents as a Poisson random variable), referring to the information of node i's d-th feature to its k-th community; $\{M^{(l)}\}_{l=0}^L$ refers to the scaling parameter in generating the related counting information; Λ_{k_1,k_2} is a community-versus-community compability parameter, and larger value of Λ_{k_1,k_2} indicates larger possiblility of generating the link betwen community k_1 and k_2; Z_{ij,k_1k_2} is a latent integer, recording the decomposed (k_1, k_2)-th latent components in the relation R_{ij}.

Incorporating Feature information Steps 1 incorporates the feature information and then generates the membership distribution $\boldsymbol{\pi}$ at the 0-th layer. Similar as the above information propagation method, latent counting variable $\boldsymbol{C}^F$ is also used to enable effcient Gibbs sampling. Based on the additive property of the Poisson distribution, each element in the Dirichlet distribution's concentration parameter vector follows a Poisson distribution as $\sum_d C_{ikd}^F \sim \text{Poisson}(M^{(0)} \sum_d F_{id} T_{dk})$, with parameter being the linear sum of node i's feature information.

Builidng the deep architecture through U-DBN Step 2 generates the membership distribution $\boldsymbol{\pi}$ in a layerwise manner. When most of construction follows the setting of Section **??**, we follow the setting of (**?**) and restriction the elements of $\boldsymbol{w}_i$ be 0 if there is no observed link from node i to the particular node.

Generating the relational data The generation of relational data also follows (**?**). Each relation R_{ij} is decomposed into community-to-community latent integers and only the Relation with the summation of its latent integers larger than 0 would be taken as observed ones. In this way, the computation cost scales to the number of positive links only.

References

Zhao, H., Du, L., Buntine, W., and Zhou, M. Dirichlet belief networks for topic structure learning. In *NeurIPS*, pp. 7966–7977, 2018.

Zhou, M., Cong, Y., and Chen, B. Augmentable gamma belief networks. *Journal of Machine Learning Research*, 17(163):1–44, 2016.

A. Sampling for finite version

Sampling $\{T_{kd}\}_{k,d}$

$$T_{kd} \sim \text{Gam}\left(k_T + \sum_i C_{ikd}^F, \frac{1}{\theta_T + M^{(0)} \sum_i F_{id}}\right) \quad (15)$$

Sampling C^F_{ikd}

$$P(C^F_{ikd}|\cdots)$$
$$\propto \frac{\left(M^{(F)}F_{id}T_{dk}\pi^{(1)}_{ik}\right)^{C^F_{ikd}}}{(C^F_{ikd})!}\cdot\frac{\Gamma(K+\sum_{k',d'}C^F_{ik'd'})}{\Gamma(1+\sum_{d'}C^F_{ikd'})} \quad (16)$$

Sampling $\boldsymbol{C}^{(l)}_{iki'}$

$$P(C^{(l)}_{iki'}|\cdots)$$
$$\propto \frac{\left(M^{(l)}\pi^{(l)}_{ik}w_{ii'}\pi^{(l+1)}_{i'k}\right)^{C^{(l)}_{iki'}}}{C^{(l)}_{iki'}!}\frac{\Gamma(K+\sum_{i'',k'}C^{(l)}_{i''k'i'})}{\Gamma(1+\sum_{i''}C^{(l)}_{i''ki'})} \quad (17)$$

Sampling $\boldsymbol{\pi}^{(l)}_i$

$$\boldsymbol{\pi}^{(1)}_i \sim \text{Dirichlet}(1+\sum_d C^{(F)}_{i\cdot d}) \quad (18)$$

$$\boldsymbol{\pi}^{(l)}_i \sim \text{Dirichlet}(1+\sum_{i'}C^{(l-1)}_{i'\cdot i}+\sum_{i'}C^{(l)}_{i\cdot i'}), l=2,\cdots,L-1 \quad (19)$$

$$\boldsymbol{\pi}^{(L)}_i \sim \text{Dirichlet}(1+\sum_{i'}C^{(L-1)}_{i'\cdot i}+X_i) \quad (20)$$

Sampling $\boldsymbol{w}$

$$\boldsymbol{w}_{i\cdot} \sim \text{Dirichlet}\left(1+\sum_{i',l,k}\boldsymbol{C}^{(l)}_{iki'}\right) \quad (21)$$

Sampling $\{X_{ik}\}_{i,k}$:

$$P(X_{ik}|M,\boldsymbol{\pi},\boldsymbol{\Lambda},\boldsymbol{Z})$$
$$\propto \frac{\left[M^{(L)}\pi^{(L)}_{ik}e^{-\sum_{j\neq i,k_2}X_{jk_2}(\Lambda_{kk_2}+\Lambda_{k_2k})}\right]^{X_{ik}}}{X_{ik}!}$$
$$\cdot (X_{ik})^{\sum_{j_1,k_2}Z_{ij_1,kk_2}+\sum_{j_2,k_1}Z_{j_2i,k_1k}}. \quad (22)$$

Sampling $\{Z_{ij,k_1k_2}\}_{i,j,k_1,k_2}$

$$Z_{ij,\cdot\cdot} \sim \text{Poisson}_+(\sum_{k_1,k_2}X_{ik_1}X_{jk_2}\Lambda_{k_1k_2}) \quad (23)$$

where $Z_{ij,\cdot\cdot}=1,2,3,\ldots$

$$(\{Z_{ij,k_1k_2}\}_{k_1,k_2})$$
$$\sim\text{Multinomial}\left(Z_{ij,\cdot\cdot};\left\{\frac{X_{ik_1}X_{jk_2}\Lambda_{k_1k_2}}{\sum_{k_1,k_2}X_{ik_1}X_{jk_2}\Lambda_{k_1k_2}}\right\}_{k_1,k_2}\right) \quad (24)$$

Sampling M

$$M \sim \text{Gam}\left(k_M+\sum_{i,k}X_{ik},\frac{1}{\theta_M+N}\right) \quad (25)$$

Sampling $\{\Lambda_{k_1k_2}\}_{k_1,k_2}$

$$\Lambda_{k_1k_2} \sim \text{Gam}\left(\sum_{i,j}Z_{ij,k_1k_2}+k_\Lambda,\frac{1}{\theta_\Lambda+\sum_{i,j}X_{ik_1}X_{jk_2}}\right) \quad (26)$$

Algorithm 1 Sampling for U-DBN

Require: relational data $\{R_{ij}\}_{i,j=1}^N$, nodes' feature information $F\in(\mathbb{R}^+\cup 0)^{N\times D}$, iteration time T
Ensure: $\{\boldsymbol{\pi}^{(l)}_i\}_{i,l},\{\boldsymbol{C}^{(l)}\}_l,\boldsymbol{W},\{\boldsymbol{X}_i\}_i,\{\Lambda_{k_1k_2}\}_{k_1,k_2},\boldsymbol{T},M$
for $t=1,\ldots,T$ **do**
  Update T_{kd} according to Eq. (15), $\forall k,d$
  Update C^F_{ikd} according to Eq. (16), $\forall i,k,d$;
  for $l=1,\ldots,L$ **do**
    Update $C^{(l)}_{iki'}$ according to Eq. (17), $\forall k,(i,i')|R_{ii'}=1$;
    Update $\boldsymbol{\pi}^{(l)}_i$ according to Eq. (18), $i=1,\ldots,N$;
  end for
  Update $\boldsymbol{w}_i$ according to Eq. (21), $i=1,\ldots,N$;
  Update X_{ik} according to Eq. (22), $\forall i,k$;
  Update Z_{ij,k_1k_2} according to Eq. (23,24), $\forall k_1,k_2,(i,j)|R_{ij}=1$;
  Update $M^{(l)}$ according to Eq. (25), $\forall l$;
  Update $\Lambda_{k_1k_2}$ according to Eq. (26), $\forall k_1,k_2$.
end for

B. Miscellaneous

Direct relations between counts Integrating over the latent distribution $\boldsymbol{\pi}^{(l)}$ can obtain direct result as:

$$P(C^{(l+1)}_1,\ldots,C^{(l+1)}_K|C^{(l)}_1,\ldots,C^{(l)}_K)\propto\frac{(\sum_k C^{(l+1)}_k)!}{\prod_k\left[C^{(l+1)}_k!\right]}\cdot\frac{\prod_k\left[(C^{(l+1)}_k+\right.}{\Gamma\left[\sum_k(C^{(l+1)}_k+C\right.} \quad (27)$$

Case 1 Assume we have $\boldsymbol{\pi}^{(1)}=[0.3,0.2,0.2,0.2,0.1]^\top$, where the last element of 0.1 refers to the ratio of unknown components. Letting $M=1000$, we may obtain $\boldsymbol{C}^{(1)}=[300,199,201,200,100]^\top$, where the last element of 100 refers to the counts for the unknown components. Using stick-breaking construction, we may obtain $\widetilde{\boldsymbol{\pi}}^{(1)}=[0.3,0.2,0.2,0.2,0.05,0.03,0.02]^\top$ and $\widetilde{\boldsymbol{C}}^{(1)}=$

$[300, 199, 201, 200, 69, 31]^\top$. The Dirichlet distribution can help us re-distribute the weight ratio as $\widetilde{\boldsymbol{\pi}}^{(2)} = [0.28, 0.19, 0.23, 0.17, 0.06, 0.02, 0.02]^\top$.

Case 2 Assume we have $\boldsymbol{\pi}^{(1)} = [0.3, 0.2, 0.01, 0.39, 0.1]^\top$, where the last element of 0.1 refers to the ratio of unknown components. Letting $M = 1000$, we may obtain $\boldsymbol{C}^{(1)} = [300, 199, 0, 401, 100]^\top$, where the last element of 100 refers to the counts for the unknown components. Using stick-breaking construction, we may obtain $\widetilde{\boldsymbol{\pi}}^{(1)} = [0.3, 0.2, 0.01, 0.39, 0.05, 0.03, 0.02]^\top$ and $\widetilde{\boldsymbol{C}}^{(1)} = [300, 199, 0, 401, 69, 31]^\top$. We may remove the third elements (i.e. with 0 count value) and allocate the corresponding weight to the unknown components' ratio. The Dirichlet distribution can help us re-distribute the weight ratio as $\widetilde{\boldsymbol{\pi}}^{(2)} = [0.28, 0.19, 0.40, 0.06, 0.02, 0.03]^\top$.