Augmentable Gamma Belief Networks

Mingyuan Zhou *   MINGYUAN.ZHOU@MCCOMBS.UTEXAS.EDU
Department of Information, Risk, and Operations Management
McCombs School of Business
The University of Texas at Austin
Austin, TX 78712, USA

Yulai Cong   YULAI_CONG@163.COM
Bo Chen *   BCHEN@MAIL.XIDIAN.EDU.CN
National Laboratory of Radar Signal Processing
Collaborative Innovation Center of Information Sensing and Understanding
Xidian University
Xi'an, Shaanxi 710071, China

Editor: Francois Caron

Abstract

To infer multilayer deep representations of high-dimensional discrete and nonnegative real vectors, we propose an augmentable gamma belief network (GBN) that factorizes each of its hidden layers into the product of a sparse connection weight matrix and the nonnegative real hidden units of the next layer. The GBN's hidden layers are jointly trained with an upward-downward Gibbs sampler that solves each layer with the same subroutine. The gamma-negative binomial process combined with a layer-wise training strategy allows inferring the width of each layer given a fixed budget on the width of the first layer. Example results illustrate interesting relationships between the width of the first layer and the inferred network structure, and demonstrate that the GBN can add more layers to improve its performance in both unsupervisedly extracting features and predicting heldout data. For exploratory data analysis, we extract trees and subnetworks from the learned deep network to visualize how the very specific factors discovered at the first hidden layer and the increasingly more general factors discovered at deeper hidden layers are related to each other, and we generate synthetic data by propagating random variables through the deep network from the top hidden layer back to the bottom data layer.

Keywords: Bayesian nonparametrics, deep learning, multilayer representation, Poisson factor analysis, topic modeling, unsupervised learning

1. Introduction

There has been significant recent interest in deep learning. Despite its tremendous success in supervised learning, inferring a multilayer data representation in an unsupervised manner remains a challenging problem (Bengio and LeCun, 2007; Ranzato et al., 2007; Bengio et al., 2015). To generate data with a deep network, it is often unclear how to set the structure of the network, including the depth (number of layers) of the network and the width (number of hidden units) of each layer. In addition, for some commonly used deep

---

*. Correspondence should be addressed to M. Zhou or B. Chen.

generative models, including the sigmoid belief network (SBN), deep belief network (DBN), and deep Boltzmann machine (DBM), the hidden units are often restricted to be binary. More specifically, the SBN, which connects the binary units of adjacent layers via the sigmoid functions, infers a deep representation of multivariate binary vectors (Neal, 1992; Saul et al., 1996); the DBN (Hinton et al., 2006) is a SBN whose top hidden layer is replaced by the restricted Boltzmann machine (RBM) (Hinton, 2002) that is undirected; and the DBM is an undirected deep network that connects the binary units of adjacent layers using the RBMs (Salakhutdinov and Hinton, 2009). All these three deep networks are designed to model binary observations, without principled ways to set the network structure. Although one may modify the bottom layer to model Gaussian and multinomial observations, the hidden units of these networks are still typically restricted to be binary (Salakhutdinov and Hinton, 2009; Larochelle and Lauly, 2012; Salakhutdinov et al., 2013). To generalize these models, one may consider the exponential family harmoniums (Welling et al., 2004; Xing et al., 2005) to construct more general networks with non-binary hidden units, but often at the expense of noticeably increased complexity in training and data fitting. To model real-valued data without restricting the hidden units to be binary, one may consider the general framework of nonlinear Gaussian belief networks (Frey and Hinton, 1999) that constructs continuous hidden units by nonlinearly transforming Gaussian distributed latent variables, including as special cases both the continuous SBN of Frey (1997a,b) and the rectified Gaussian nets of Hinton and Ghahramani (1997). More recent scalable generalizations under that framework include variational auto-encoders (Kingma and Welling, 2014) and deep latent Gaussian models (Rezende et al., 2014).

Moving beyond conventional deep generative models using binary or nonlinearly transformed Gaussian hidden units and setting the network structure in a heuristic manner, we construct deep networks using gamma distributed nonnegative real hidden units, and combine the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) with a greedy-layer wise training strategy to automatically infer the network structure. The proposed model is called the augmentable gamma belief network, referred to hereafter for brevity as the GBN, which factorizes the observed or latent count vectors under the Poisson likelihood into the product of a factor loading matrix and the gamma distributed hidden units (factor scores) of layer one; and further factorizes the shape parameters of the gamma hidden units of each layer into the product of a connection weight matrix and the gamma hidden units of the next layer. The GBN together with Poisson factor analysis can unsupervisedly infer a multilayer representation from multivariate count vectors, with a simple but powerful mechanism to capture the correlations between the visible/hidden features across all layers and handle highly overdispersed counts. With the Bernoulli-Poisson link function (Zhou, 2015), the GBN is further applied to high-dimensional sparse binary vectors by truncating latent counts, and with a Poisson randomized gamma distribution, the GBN is further applied to high-dimensional sparse nonnegative real data by randomizing the gamma shape parameters with latent counts.

For tractable inference of a deep generative model, one often applies either a sampling based procedure (Neal, 1992; Frey, 1997a) or variational inference (Saul et al., 1996; Frey, 1997b; Ranganath et al., 2014b; Kingma and Welling, 2014). However, conjugate priors on the model parameters that connect adjacent layers are often unknown, making it difficult to develop fully Bayesian inference that infers the posterior distributions of these parameters.

It was not until recently that a Gibbs sampling algorithm, imposing priors on the network connection weights and sampling from their conditional posteriors, was developed for the SBN by Gan et al. (2015b), using the Polya-Gamma data augmentation technique developed for logistic models (Polson et al., 2012). In this paper, we will develop data augmentation technique unique for the augmentable GBN, allowing us to develop a fully Bayesian upward-downward Gibbs sampling algorithm to infer the posterior distributions of not only the hidden units, but also the connection weights between adjacent layers.

Distinct from previous deep networks that often require tuning both the width (number of hidden units) of each layer and the network depth (number of layers), the GBN employs nonnegative real hidden units and automatically infers the widths of subsequent layers given a fixed budget on the width of its first layer. Note that the budget could be infinite and hence the whole network can grow without bound as more data are being observed. Similar to other belief networks that can often be improved by adding more hidden layers (Hinton et al., 2006; Sutskever and Hinton, 2008; Bengio et al., 2015), for the proposed model, when the budget on the first layer is finite and hence the ultimate capacity of the network could be limited, our experimental results also show that a GBN equipped with a narrower first layer could increase its depth to match or even outperform a shallower one with a substantially wider first layer.

The gamma distribution density function has the highly desired strong non-linearity for deep learning, but the existence of neither a conjugate prior nor a closed-form maximum likelihood estimate (Choi and Wette, 1969) for its shape parameter makes a deep network with gamma hidden units appear unattractive. Despite seemingly difficult, we discover that, by generalizing the data augmentation and marginalization techniques for discrete data modeled with the Poisson, gamma, and negative binomial distributions (Zhou and Carin, 2015), one may propagate latent counts one layer at a time from the bottom data layer to the top hidden layer, with which one may derive an efficient upward-downward Gibbs sampler that, one layer at a time in each iteration, upward samples Dirichlet distributed connection weight vectors and then downward samples gamma distributed hidden units, with the latent parameters of each layer solved with the same subroutine.

With extensive experiments in text and image analysis, we demonstrate that the deep GBN with two or more hidden layers clearly outperforms the shallow GBN with a single hidden layer in both unsupervisedly extracting latent features for classification and predicting heldout data. Moreover, we demonstrate the excellent ability of the GBN in exploratory data analysis: by extracting trees and subnetworks from the learned deep network, we can follow the paths of each tree to visualize various aspects of the data, from very general to very specific, and understand how they are related to each other.

In addition to constructing a new deep network that well fits high-dimensional sparse binary, count, and nonnegative real data, developing an efficient upward-downward Gibbs sampler, and applying the learned deep network for exploratory data analysis, other contributions of the paper include: 1) proposing novel link functions, 2) combining the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) with a layer-wise training strategy to automatically infer the network structure; 3) revealing the relationship between the upper bound imposed on the width of the first layer and the inferred widths of subsequent layers; 4) revealing the relationship between the depth of the network and the model's ability to model overdispersed counts; and 5) generating multivariate high-

dimensional discrete or nonnegative real vectors, whose distributions are governed by the GBN, by propagating the gamma hidden units of the top hidden layer back to the bottom data layer. We note this paper significantly extends our recent conference publication (Zhou et al., 2015a) that proposes the Poisson GBN.

2. Augmentable Gamma Belief Networks

Denoting $\boldsymbol{\theta}_j^{(t)} \in \mathbb{R}_+^{K_t}$ as the K_t hidden units of sample j at layer t, where $\mathbb{R}_+ = \{x : x \geq 0\}$, the generative model of the augmentable gamma belief network (GBN) with T hidden layers, from top to bottom, is expressed as

$$\boldsymbol{\theta}_j^{(T)} \sim \text{Gam}\left(\boldsymbol{r}, 1/c_j^{(T+1)}\right),$$

$$\vdots$$

$$\boldsymbol{\theta}_j^{(t)} \sim \text{Gam}\left(\boldsymbol{\Phi}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}, 1/c_j^{(t+1)}\right),$$

$$\vdots$$

$$\boldsymbol{\theta}_j^{(1)} \sim \text{Gam}\left(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_j^{(2)}, p_j^{(2)}/\left(1-p_j^{(2)}\right)\right), \tag{1}$$

where $x \sim \text{Gam}(a, 1/c)$ represents a gamma distribution with mean a/c and variance a/c^2. For $t = 1, 2, \ldots, T-1$, the GBN factorizes the shape parameters of the gamma distributed hidden units $\boldsymbol{\theta}_j^{(t)} \in \mathbb{R}_+^{K_t}$ of layer t into the product of the connection weight matrix $\boldsymbol{\Phi}^{(t+1)} \in \mathbb{R}_+^{K_t \times K_{t+1}}$ and the hidden units $\boldsymbol{\theta}_j^{(t+1)} \in \mathbb{R}_+^{K_{t+1}}$ of layer $t + 1$; the top layer's hidden units $\boldsymbol{\theta}_j^{(T)}$ share the same vector $\boldsymbol{r} = (r_1, \ldots, r_{K_T})'$ as their gamma shape parameters; and the $p_j^{(2)}$ are probability parameters and $\{1/c^{(t)}\}_{3,T+1}$ are gamma scale parameters, with $c_j^{(2)} := \left(1-p_j^{(2)}\right)/p_j^{(2)}$. We will discuss later how to measure the connection strengths between the nodes of adjacent layers and the overall popularity of a factor at a particular hidden layer.

For scale identifiability and ease of inference and interpretation, each column of $\boldsymbol{\Phi}^{(t)} \in \mathbb{R}_+^{K_{t-1} \times K_t}$ is restricted to have a unit L_1 norm and hence $0 \leq \boldsymbol{\Phi}^{(t)}(k', k) \leq 1$. To complete the hierarchical model, for $t \in \{1, \ldots, T-1\}$, we let

$$\boldsymbol{\phi}_k^{(t)} \sim \text{Dir}\left(\eta^{(t)}, \ldots, \eta^{(t)}\right), \quad r_k \sim \text{Gam}\left(\gamma_0/K_T, 1/c_0\right) \tag{2}$$

where $\boldsymbol{\phi}_k^{(t)} \in \mathbb{R}_+^{K_{t-1}}$ is the kth column of $\boldsymbol{\Phi}^{(t)}$; we impose $c_0 \sim \text{Gam}(e_0, 1/f_0)$ and $\gamma_0 \sim \text{Gam}(a_0, 1/b_0)$; and for $t \in \{3, \ldots, T+1\}$, we let

$$p_j^{(2)} \sim \text{Beta}(a_0, b_0), \quad c_j^{(t)} \sim \text{Gam}(e_0, 1/f_0). \tag{3}$$

We expect the correlations between the K_t rows (latent features) of $(\boldsymbol{\theta}_1^{(t)}, \ldots, \boldsymbol{\theta}_J^{(t)})$ to be captured by the columns of $\boldsymbol{\Phi}^{(t+1)}$. Even if $\boldsymbol{\Phi}^{(t)}$ for $t \geq 2$ are all identity matrices, indicating no correlations between the latent features to be captured, our analysis in Section 3.2 will show that a deep structure with $T \geq 2$ could still benefit data fitting by better modeling the variability of the latent features $\boldsymbol{\theta}_j^{(1)}$. Before further examining the network structure, below we first introduce a set of distributions that will be used to either model different types of data or augment the model for simple inference.

2.1 Distributions for Count, Binary, and Nonnegative Real Data

Below we first describe some useful count distributions that will be used later.

2.1.1 USEFUL COUNT DISTRIBUTIONS AND THEIR RELATIONSHIPS

Let the Chinese restaurant table (CRT) distribution $l \sim \mathrm{CRT}(n, r)$ represent the random number of tables seated by n customers in a Chinese restaurant process (Blackwell and Mac-Queen, 1973; Antoniak, 1974; Aldous, 1985; Pitman, 2006) with concentration parameter r. Its probability mass function (PMF) can be expressed as

$$P(l \mid n, r) = \frac{\Gamma(r)r^l}{\Gamma(n+r)}|s(n,l)|,$$

where $l \in \mathbb{Z}$, $\mathbb{Z} := \{0, 1, \ldots, n\}$, and $|s(n, l)|$ are unsigned Stirling numbers of the first kind. A CRT distributed sample can be generated by taking the summation of n independent Bernoulli random variables as

$$l = \sum_{i=1}^{n} b_i, \ b_i \sim \mathrm{Bernoulli}\left[r/(r+i-1)\right].$$

Let $u \sim \mathrm{Log}(p)$ denote the logarithmic distribution (Fisher et al., 1943; Anscombe, 1950; Johnson et al., 1997) with PMF

$$P(u \mid p) = \frac{1}{-\ln(1-p)} \frac{p^u}{u},$$

where $u \in \{1, 2, \ldots\}$, and let $n \sim \mathrm{NB}(r, p)$ denote the negative binomial (NB) distribution (Greenwood and Yule, 1920; Bliss and Fisher, 1953) with PMF

$$P(n \mid r, p) = \frac{\Gamma(n+r)}{n!\Gamma(r)} p^n (1-p)^r,$$

where $n \in \mathbb{Z}$. The NB distribution $n \sim \mathrm{NB}(r, p)$ can be generated as a gamma mixed Poisson distribution as

$$n \sim \mathrm{Pois}(\lambda), \ \lambda \sim \mathrm{Gam}\left[r, p/(1-p)\right],$$

where $p/(1-p)$ is the gamma scale parameter.

As shown in (Zhou and Carin, 2015), the joint distribution of n and l given r and p in

$$l \sim \mathrm{CRT}(n, r), \ n \sim \mathrm{NB}(r, p),$$

where $l \in \{0, \ldots, n\}$ and $n \in \mathbb{Z}$, is the same as that in

$$n = \textstyle\sum_{t=1}^{l} u_t, \ u_t \sim \mathrm{Log}(p), \ l \sim \mathrm{Pois}[-r \ln(1-p)], \tag{4}$$

which is called the Poisson-logarithmic bivariate distribution, with PMF

$$P(n, l \mid r, p) = \frac{|s(n,l)|r^l}{n!} p^n (1-p)^r.$$

We will exploit these relationships to derive efficient inference for the proposed models.

2.1.2 Bernoulli-Poisson Link and Truncated Poisson Distribution

As in Zhou (2015), the Bernoulli-Poisson (BerPo) link thresholds a random count at one to obtain a binary variable as

$$b = \mathbf{1}(m \geq 1), \ m \sim \text{Pois}(\lambda), \tag{5}$$

where $b = 1$ if $m \geq 1$ and $b = 0$ if $m = 0$. If m is marginalized out from (5), then given λ, one obtains a Bernoulli random variable as $b \sim \text{Ber}\left(1 - e^{-\lambda}\right)$. The conditional posterior of the latent count m can be expressed as

$$(m \,|\, b, \lambda) \sim b \cdot \text{Pois}_+(\lambda),$$

where $x \sim \text{Pois}_+(\lambda)$ follows a truncated Poisson distribution, with $P(x = k) = (1 - e^{-\lambda})^{-1}\lambda^k e^{-\lambda}/k!$ for $k \in \{1, 2, \ldots\}$. Thus if $b = 0$, then $m = 0$ almost surely (a.s.), and if $b = 1$, then $m \sim \text{Pois}_+(\lambda)$, which can be simulated with a rejection sampler that has a minimal acceptance rate of 63.2% at $\lambda = 1$ (Zhou, 2015). Given the latent count m and a gamma prior on λ, one can then update λ using the gamma-Poisson conjugacy. The BerPo link shares some similarities with the probit link that thresholds a normal random variable at zero, and the logistic link that lets $b \sim \text{Ber}[e^x/(1 + e^x)]$. We advocate the BerPo link as an alternative to the probit and logistic links since if $b = 0$, then $m = 0$ a.s., which could lead to significant computational savings if the binary vectors are sparse. In addition, the conjugacy between the gamma and Poisson distributions makes it convenient to construct hierarchical Bayesian models amenable to posterior simulation.

2.1.3 Poisson Randomized Gamma and Truncated Bessel Distributions

To model nonnegative data that include both zeros and positive observations, we introduce the Poisson randomized gamma (PRG) distribution as

$$x \sim \text{PRG}(\lambda, c),$$

whose distribution has a point mass at $x = 0$ and is continuous for $x > 0$. The PRG distribution is generated as a Poisson mixed gamma distribution as

$$x \sim \text{Gam}(n, 1/c), \ n \sim \text{Pois}(\lambda),$$

in which we define $\text{Gam}(0, 1/c) = 0$ a.s. and hence $x = 0$ if and only $n = 0$. Thus the PMF of $x \sim \text{PRG}(\lambda, c)$ can be expressed as

$$
\begin{aligned}
f_X(x \,|\, \lambda, c) &= \sum_{n=0}^{\infty} \text{Gam}(x; n, 1/c)\text{Pois}(n; \lambda) \\
&= \left(e^{-\lambda}\right)^{\mathbf{1}(x=0)} \left[e^{-\lambda - cx}\sqrt{\frac{\lambda c}{x}}\, I_{-1}\left(2\sqrt{\lambda cx}\right)\right]^{\mathbf{1}(x>0)},
\end{aligned}
\tag{6}
$$

where

$$I_{-1}(\alpha) = \left(\frac{\alpha}{2}\right)^{-1} \sum_{n=1}^{\infty} \frac{\left(\frac{\alpha^2}{4}\right)^n}{n!\Gamma(n)}, \quad \alpha > 0$$

is the modified Bessel function of the first kind $I_\nu(\alpha)$ with ν fixed at -1. Using the laws of total expectation and total variance, or using the PMF directly, one may show that

$$\mathbb{E}[x \mid \lambda, c] = \lambda/c, \quad \text{var}[x \mid \lambda, c] = 2\lambda/c^2.$$

Thus the variance-to-mean ratio of the PRG distribution is $2/c$, as controlled by c.

The conditional posterior of n given x, λ, and c can be expressed as

$$
\begin{aligned}
f_N(n \mid x, \lambda, c) &= \frac{\text{Gam}(x; n, 1/c)\text{Pois}(n; \lambda)}{\text{PRG}(x; \lambda, c)} \\
&= \mathbf{1}(x = 0)\delta_0 + \mathbf{1}(x > 0) \sum_{n=1}^{\infty} \frac{1}{I_{-1}\left(2\sqrt{\lambda c x}\right)} \frac{(\lambda c x)^{n-\frac{1}{2}}}{n!\Gamma(n)} \delta_n \\
&= \mathbf{1}(x = 0)\delta_0 + \mathbf{1}(x > 0) \sum_{n=1}^{\infty} \text{Bessel}_{-1}(n; 2\sqrt{c x \lambda})\delta_n,
\end{aligned}
\tag{7}
$$

where we define $n \sim \text{Bessel}_{-1}(\alpha)$ as the truncated Bessel distribution, with PMF

$$\text{Bessel}_{-1}(n; \alpha) = \frac{\left(\frac{\alpha}{2}\right)^{2n-1}}{I_{-1}(\alpha)n!\Gamma(n)}, \quad n \in \{1, 2, \ldots\}.$$

Thus $n = 0$ if and only if $x = 0$, and n is a positive integer drawn from a truncated Bessel distribution if $x > 0$. In Appendix A, we plot the probability distribution functions of the proposed PRG and truncated Bessel distributions and show how they differ from the randomized gamma and Bessel distributions (Yuan and Kalbfleisch, 2000), respectively.

2.2 Link Functions for Three Different Types of Observations

If the observations are multivariate count vectors $\boldsymbol{x}_j^{(1)} \in \mathbb{Z}^V$, where $V := K_0$, then we link the integer-valued visible units to the nonnegative real hidden units at layer one using Poisson factor analysis (PFA) as

$$\boldsymbol{x}_j^{(1)} \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_j^{(1)}\right).\tag{8}$$

Under this construction, the correlations between the K_0 rows (features) of $(\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_J^{(1)})$ are captured by the columns of $\boldsymbol{\Phi}^{(1)}$. Detailed descriptions on how PFA is related to a wide variety of discrete latent variable models, including nonnegative matrix factorization (Lee and Seung, 2001), latent Dirichlet allocation (Blei et al., 2003), the gamma-Poisson model (Canny, 2004), discrete Principal component analysis (Buntine and Jakulin, 2006), and the focused topic model (Williamson et al., 2010), can be found in Zhou et al. (2012) and Zhou and Carin (2015).

We call PFA using the GBN in (1) as the prior on its factor scores as the Poisson gamma belief network (PGBN), as proposed in Zhou et al. (2015a). The PGBN can be naturally applied to factorize the term-document frequency count matrix of a text corpus, not only extracting semantically meaningful topics at multiple layers, but also capturing the relationships between the topics of different layers using the deep network, as discussed below in both Sections 2.3 and 4.
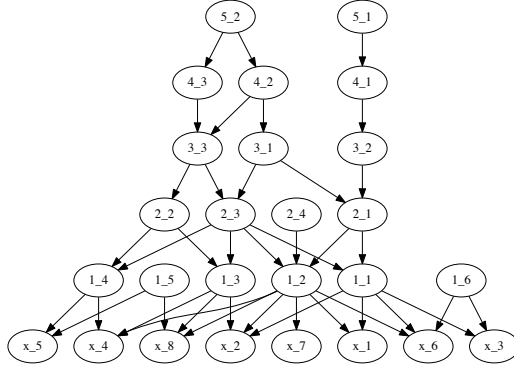
Figure 1: An example directed network of five hidden layers, with $K_0 = 8$ visible units, $[K_1, K_2, K_3, K_4, K_5] = [6, 4, 3, 3, 2]$, and sparse connections between the units of adjacent layers.

If the observations are high-dimensional sparse binary vectors $\boldsymbol{b}_j^{(1)} \in \{0, 1\}^V$, then we factorize them using Bernoulli-Poisson factor analysis (Ber-PFA) as

$$\boldsymbol{b}_j^{(1)} = \mathbf{1}\big(\boldsymbol{x}_j^{(1)} \geq 0\big), \ \boldsymbol{x}_j^{(1)} \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_j^{(1)}\right). \tag{9}$$

We call Ber-PFA with the augmentable GBN as the prior on its factor scores $\boldsymbol{\theta}_j^{(1)}$ as the Bernoulli-Poisson gamma belief network (BerPo-GBN).

If the observations are high-dimensional sparse nonnegative real-valued vectors $\boldsymbol{y}_j^{(1)} \in \mathbb{R}_+^V$, then we factorize them using Poisson randomized gamma (PRG) factor analysis as

$$\boldsymbol{y}_j^{(1)} \sim \text{Gam}(\boldsymbol{x}_j^{(1)}, 1/a_j), \ \boldsymbol{x}_j^{(1)} \sim \text{Pois}\left(\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_j^{(1)}\right). \tag{10}$$

We call PRG factor analysis with the augmentable GBN as the prior on its factor scores $\boldsymbol{\theta}_j^{(1)}$ as the PRG gamma belief network (PRG-GBN).

We show in Figure 1 an example directed belief network of five hidden layers, with $K_0 = 8$ visible units, with 6, 4, 3, 3, and 2 hidden units for layers one, two, three, four, and five, respectively, and with sparse connections between the units of adjacent layers.

2.3 Exploratory Data Analysis

To interpret the network structure of the GBN, we notice that

$$\mathbb{E}\big[\boldsymbol{x}_j^{(1)} \,\big|\, \boldsymbol{\theta}_j^{(t)}, \{\boldsymbol{\Phi}^{(\ell)}, c_j^{(\ell)}\}_{1,t}\big] = \left[\prod_{\ell=1}^{t} \boldsymbol{\Phi}^{(\ell)}\right] \frac{\boldsymbol{\theta}_j^{(t)}}{\prod_{\ell=2}^{t} c_j^{(\ell)}}, \tag{11}$$

$$\mathbb{E}\big[\boldsymbol{\theta}_j^{(t)} \,\big|\, \{\boldsymbol{\Phi}^{(\ell)}, c_j^{(\ell)}\}_{t+1,T}, \boldsymbol{r}\big] = \left[\prod_{\ell=t+1}^{T} \boldsymbol{\Phi}^{(\ell)}\right] \frac{\boldsymbol{r}}{\prod_{\ell=t+1}^{T+1} c_j^{(\ell)}}. \tag{12}$$
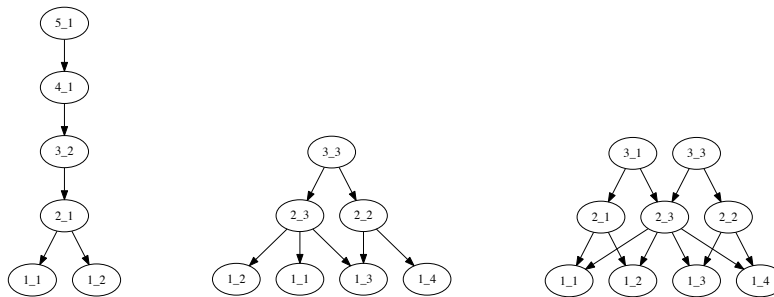
Figure 2: Extracted from the network shown in Figure 1, the left plot is a tree rooted at node 5_1, the middle plot is a tree rooted at node 3_3, and the right plot is a subnetwork consisting of both the tree rooted at node 3_1 and the tree rooted at node 3_3.

Thus for visualization, it is straightforward to project the K_t topics/hidden units/factor loadings/nodes of layer $t \in \{1, \ldots, T\}$ to the bottom data layer as the columns of the $V \times K_t$ matrix

$$\prod_{\ell=1}^{t} \mathbf{\Phi}^{(\ell)}, \tag{13}$$

and rank their popularities using the K_t dimensional nonnegative weight vector

$$\boldsymbol{r}^{(t)} := \left[\prod_{\ell=t+1}^{T} \mathbf{\Phi}^{(\ell)} \right] \boldsymbol{r}. \tag{14}$$

To measure the connection strength between node k of layer t and node k' of layer $t-1$, we use the value of

$$\mathbf{\Phi}^{(t)}(k', k),$$

which is also expressed as $\phi_k^{(t)}(k')$ or $\phi_{k'k}^{(t)}$.

Our intuition is that examining the nodes of the hidden layers, via their projections to the bottom data layer, from the top to bottom layers will gradually reveal less general and more specific aspects of the data. To verify this intuition and further understand the relationships between the general and specific aspects of the data, we consider extracting a tree for each node of layer t, where $t \geq 2$, to help visualize the inferred multilayer deep structure. To be more specific, to construct a tree rooted at a node of layer t, we grow the tree downward by linking the root node (if at layer t) or each leaf node of the tree (if at a layer below layer t) to all the nodes at the layer below that are connected to the root/leaf node with non-negligible weights. Note that a tree in our definition permits a node to have more than one parent, which means that different branches of the tree can overlap with each other. In addition, we also consider extracting subnetworks, each of which consists of multiple related trees from the full deep network. For example, shown in the left of Figure 2 is the tree extracted from the network in Figure 1 using node 5_1 as the root, shown in the middle is the tree using node 3_3 as the root, and shown in the right is a subnetwork consisting of two related trees that are rooted at nodes 3_1 and 3_3, respectively.

2.3.1 Visualizing Nodes of Different Layers

Before presenting the technical details, we first provide some example results obtained with the PGBN on extracting multilayer representations from the 11,269 training documents of the 20newsgroups data set (http://qwone.com/~jason/20Newsgroups/). Given a fixed budget of $K_{1\max} = 800$ on the width of the first layer, with $\eta^{(t)} = 0.1$ for all t, a five-layer deep network inferred by the PGBN has a network structure as $[K_1, K_2, K_3, K_4, K_5] = [386, 63, 58, 54, 51]$, meaning that there are 386, 63, 58, 54, and 51 nodes at layers one to five, respectively.

For visualization, we first relabel the nodes at each layer based on their weights $\{r_k^{(t)}\}_{1,K_t}$, calculated as in (14), with a more popular (larger weight) node assigned with a smaller label. We visualize node k of layer t by displaying its top 12 words ranked according to their probabilities in $\left(\prod_{\ell=1}^{t-1} \mathbf{\Phi}^{(\ell)}\right)\phi_k^{(t)}$, the kth column of the projected representation calculated as in (13). We set the font size of node k of layer t proportional to $\left(r_k^{(t)}/r_1^{(t)}\right)^{\frac{1}{10}}$ in each subplot, and color the outside border of a text box as red, green, orange, blue, or black for a node of layer five, four, three, two, or one, respectively. For better interpretation, we also exclude from the vocabulary the top 30 words of node 1 of layer one: "don just like people think know time good make way does writes edu ve want say really article use right did things point going better thing need sure used little," and the top 20 words of node 2 of layer one: "edu writes article com apr cs ca just know don like think news cc david university john org wrote world." These 50 words are not in the standard list of stopwords but can be considered as stopwords specific to the 20newsgroups corpus discovered by the PGBN.

For the $[386, 63, 58, 54, 51]$ PGBN learned on the 20newsgroups corpus, we plot 54 example topics of layer one in Figure 3, the top 30 topics of layer three in Figure 4, and the top 30 topics of layer five in Figure 5. Figure 3 clearly shows that the topics of layer one, except for topics 1-3 that mainly consist of common functional words of the corpus, are all very specific. For example, topics 71 and 81 shown in the first row are about "candida yeast symptoms" and "sex," respectively, topics 53, 73, 83, and 84 shown in the second row are about "printer," "msg," "police radar detector," and "Canadian health care system," respectively, and topics 46 and 76 shown in third row are about "ice hockey" and "second amendment," respectively. By contrast, the topics of layers three and five, shown in Figures 4 and 5, respectively, are much less specific and can in general be matched to one or two news groups out of the 20 news groups, including comp.{graphics, os.ms-windows.misc, sys.ibm.pc.hardware, sys.mac.hardware, windows.x}, rec.{autos, motorcycles}, rec.sport.{baseball, hockey}, sci.{crypt, electronics, med, space}, misc.forsale, talk.politics.{misc, guns, mideast}, and {talk.religion.misc, alt.atheism, soc.religion.christian}.

2.3.2 Visualizing Trees Rooted at The Top-Layer Hidden Units

While it is interesting to examine the topics of different layers to understand the general and specific aspects of the corpus used to train the PGBN, it would be more informative to further illustrate how the topics of different layers are related to each other. Thus we consider constructing trees to visualize the PGBN. We first pick a node as the root of a tree and grow the tree downward by drawing a line from node k at layer t, the root or a leaf node of the tree, to node k' at layer $t-1$ for all k' in the set $\{k' : \mathbf{\Phi}^{(t)}(k', k) > \tau_t/K_{t-1}\}$,

Figure 3: Topics (layer one of the PGBN trained on the 20newsgroups corpus)

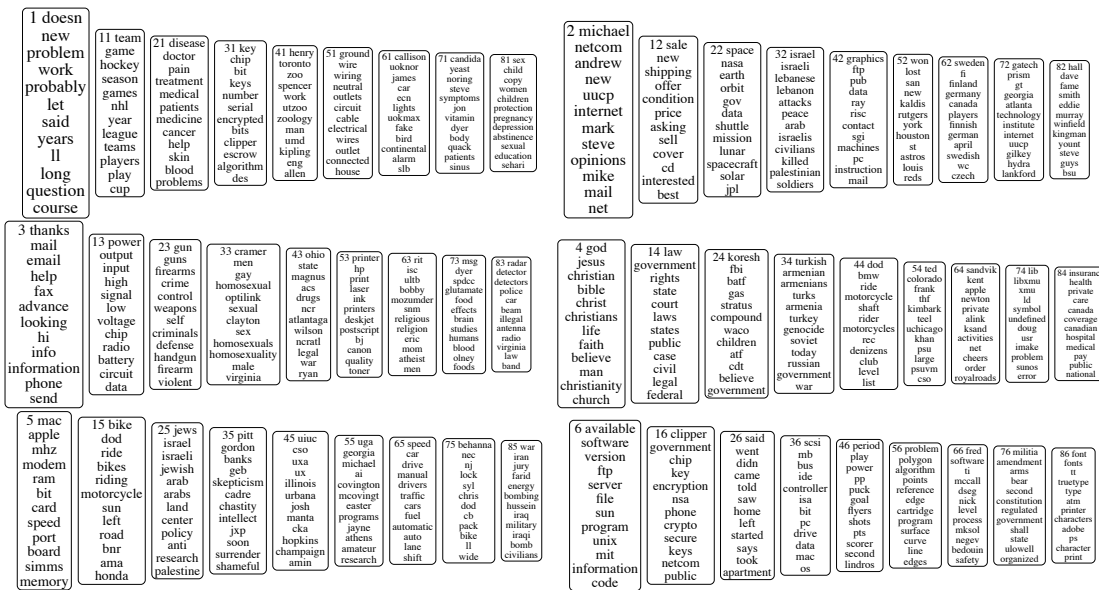| #	words
1	doesn new problem work probably let said years ll long question course
11	team game hockey season games nhl year league teams players play cup
21	disease doctor pain treatment medical patients medicine cancer help skin blood problems
31	key chip bit keys number serial encrypted bits clipper escrow algorithm des
41	henry toronto zoo spencer work utzoo zoology man umd kipling eng allen
51	ground wire wiring neutral outlets circuit cable electrical wires outlet connected house
61	callison uoknor james car ecn lights uokmax fake bird continental alarm slb
71	candida yeast noring steve symptoms jon vitamin dyer body quack patients sinus
81	sex child copy women children protection pregnancy depression abstinence sexual education sehari
2	michael netcom andrew new uucp internet mark steve opinions mike mail net
12	sale new shipping offer condition price asking sell cover cd interested best
22	space nasa earth orbit gov data shuttle mission lunar spacecraft solar jpl
32	israel israeli lebanese lebanon attacks peace arab israelis civilians killed palestinian soldiers
42	graphics ftp pub data ray risc contact sgi machines pc instruction mail
52	won lost san new kaldis rutgers york houston st astros louis reds
62	sweden fi finland germany canada players finnish german we czech
72	gatech prism gt georgia atlanta technology institute internet uucp gilkey hydra lankford
82	hall dave fame smith eddie murray winfield kingman yount steve guys bsu
3	thanks mail email help fax advance looking hi info information phone send
13	power output input high signal low voltage chip radio battery circuit data
23	gun guns firearms crime control weapons self criminals defense handgun firearm violent
33	cramer men gay homosexual optilink sexual clayton sex homosexuals homosexuality male virginia
43	ohio state magnus acs drugs ncr atlantaga wilson ncraif legal war ryan
53	printer hp print laser ink printers deskjet postscript bj canon quality toner
63	rit isc ultb bobby mozumder ssm religious eric mom atheist men
73	msg dyer spdcc glutamate food effects brain studies humans blood olney foods
83	radar detector detectors police car beam illegal antenna radio virginia law band
4	god jesus christian bible christ christians life faith believe man christianity church
14	law government rights state court laws states public case civil legal federal
24	koresh fbi batf gas stratus compound waco children atf cdt believe government
34	turkish armenian armenians turks armenia turkey genocide soviet today russian government war
44	dod bmw ride motorcycle shaft rider motorcycles rec denizens club level list
54	ted colorado frank thf kimbark teel uchicago khan psu large psuvm cso
64	sandvik kent apple newton private alink ksand net cheers order royalroads
74	lib libxmu xmu ld symbol undefined doug usr imake problem sunos error
84	insurance health private care canada coverage canadian hospital medical pay public national
5	mac apple mhz modem ram bit card speed port board simms memory
15	bike dod ride bikes riding motorcycle sun left road bnr ama honda
25	jews israel israeli jewish arab arabs land center policy anti research palestine
35	pitt gordon banks geb skepticism cadre chastity intellect jxp soon surrender shameful
45	uiuc cso uxa ux illinois urbana josh manta cka hopkins champaign amin
55	uga georgia michael ai covington mcovingt easter programs jayne athens amateur research
65	speed car drive manual drivers traffic cars fuel automatic auto lane shift
75	behanna nec nj lock syl chris dod cb pack bike ll wide
85	war iran jury farid energy bombing hussein iraq military iraqi bomb civilians
6	available software version ftp server file sun program unix mit information code
16	clipper government chip key encryption nsa phone crypto secure keys netcom public
26	said went didn came told saw home left started says took apartment
36	scsi mb bus ide controller isa bit pc drive data mac os
46	period play power pp puck goal flyers shots pts scorer second lindros
56	problem polygon algorithm points reference edge cartridge program surface curve line edges
66	fred software ti mccall dseg nick level process mksol negev bedouin safety
76	militia amendment arms bear second constitution regulated government shall state ulowell organized
86	font fonts tt truetype type atm printer characters adobe ps character print

Figure 3: Example topics of layer one of the PGBN trained on the 20newsgroups corpus.

Figure 4: Top 30 topics of layer three of the PGBN

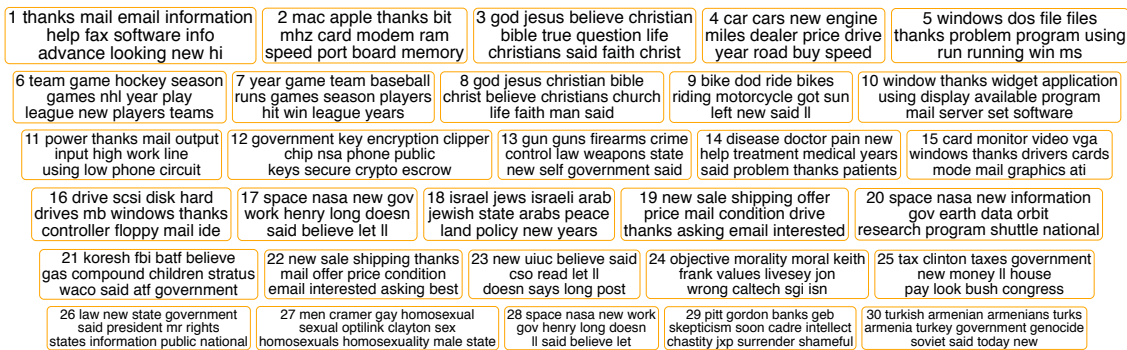| #	words
1	thanks mail email information help fax software info advance looking new hi
2	mac apple thanks bit mhz card modem ram speed port board memory
3	god jesus believe christian bible true question life christians said faith christ
4	car cars new engine miles dealer price drive year road buy speed
5	windows dos file files thanks problem program using run running win ms
6	team game hockey season games nhl year play league new players teams
7	year game team baseball runs games season players hit win league years
8	god jesus christian bible christ believe christians church life faith man said
9	bike dod ride bikes riding motorcycle got sun left new said ll
10	window thanks widget application using display available program mail server set software
11	power thanks mail output input high work line using low phone circuit
12	government key encryption clipper chip nsa phone public keys secure crypto escrow
13	gun guns firearms crime control law weapons state new self government said
14	disease doctor pain new help treatment medical years said problem thanks patients
15	card monitor video vga windows thanks drivers cards mode mail graphics ati
16	drive scsi disk hard drives mb windows thanks controller floppy mail ide
17	space nasa new gov work henry long doesn said believe let ll
18	israel jews israeli arab jewish state arabs peace land policy new years
19	new sale shipping offer price mail condition drive thanks asking email interested
20	space nasa new information gov earth data orbit research program shuttle national
21	koresh fbi batf believe gas compound children stratus waco said atf government
22	new sale shipping thanks mail offer price condition email interested asking best
23	new israel jews said cso read let ll doesn says long post
24	objective morality moral keith frank values livesey jon wrong caltech sgi isn
25	tax clinton taxes government new money ll house pay look bush congress
26	law new state government said president mr rights states information public national
27	men cramer gay homosexual sexual optilink clayton sex homosexuals homosexuality male state
28	space nasa new work gov henry long doesn ll said believe let
29	pitt gordon banks geb skepticism soon cadre intellect chastity jxp surrender shameful
30	turkish armenian armenians turks armenia turkey government genocide soviet said today new

Figure 4: The top 30 topics of layer three of the PGBN trained on the 20newsgroups corpus.

Figure 5: Top 30 topics of layer five of the PGBN

#	words
1	windows thanks file dos window using mail program problem help files work
2	god believe jesus true question said new christian bible life moral objective
3	car bike new cars dod engine got thanks price road ll miles
4	thanks mail email help information fax software info new advance looking hi
5	card thanks new mac apple mail drive bit work video mb problem
6	space nasa new gov work years long earth said orbit ll year
7	team game hockey season games nhl year play new league players teams
8	god jesus christian bible believe christ christians life new church said man
9	key chip government clipper encryption keys nsa phone public escrow secure new
10	card thanks mac windows apple mail drive bit problem work mb new
11	car bike new cars dod engine got road said ll miles ride
12	thanks new mail power email work phone high help sale price using
13	year game team baseball games runs season players hit win league years
14	disease new doctor pain help treatment said years thanks problem medical work
15	israel jews israeli armenian turkish new state government said armenians years law
16	gun koresh fbi believe batf said government children guns new gas compound
17	thanks drive card mail work mac new bit apple problem power mb
18	gun guns firearms crime law control new state weapons government said believe
19	tax clinton government new taxes law ll state said money believe years
20	god jesus believe new said christian bible life read day says doesn
21	drive disk scsi hard drives mb thanks windows controller mail floppy ide
22	thanks mail information email new help fax software space info available send
23	israel jews israeli arab jewish state new arabs peace land years true
24	pitt gordon banks geb skepticism soon cadre intellect chastity jxp surrender shameful
25	new sale shipping offer price mail thanks condition drive asking email interested
26	team game year season games hockey players play league new win baseball
27	men cramer gay new homosexual sexual optilink said believe state government law
28	god jesus christian bible believe christ christians life new church said faith
29	year game team baseball games runs season players hit new years win
30	new mail thanks key internet information email work number ll read believe

Figure 5: The top 30 topics of layer five of the PGBN trained on the 20newsgroups corpus.

Figure 6: A $[18, 5, 4, 1, 1]$ tree that includes all the lower-layer nodes (directly or indirectly) linked with non-negligible weights to the top ranked node of the top layer, taken from the full $[386, 63, 58, 54, 51]$ network inferred by the PGBN on the 11,269 training documents of the 20newsgroups corpus, with $\eta^{(t)} = 0.1$ for all t. A line from node k at layer t to node k' at layer $t-1$ indicates that $\Phi^{(t)}(k', k) > 3/K_{t-1}$, with the width of the line proportional to $\sqrt{\Phi^{(t)}(k', k)}$. For each node, the rank (in terms of popularity) at the corresponding layer and the top 12 words of the corresponding topic are displayed inside the text box, where the text font size monotonically decreases as the popularity of the node decreases, and the outside border of the text box is colored as red, green, orange, blue, or black if the node is at layer five, four, three, two, or one, respectively.

where we set the width of the line connecting node k of layer t to node k' of layer $t-1$ be proportional to $\sqrt{\Phi^{(t)}(k', k)}$ and use τ_t to adjust the complexity of a tree. In general, increasing τ_t would discard more weak connections and hence make the tree simpler and easier to visualize.

We set $\tau_t = 3$ for all t to visualize both a five-layer tree rooted at the top ranked node of the top hidden layer, as shown in Figure 6, and a five-layer tree rooted at the second ranked node of the top hidden layer, as shown in Figure 7. For the tree in Figure 6, while it is somewhat vague to determine the actual meanings of both node 1 of layer five and node 1 of layer four based on their top words, examining the more specific topics of layers three and two within the tree clearly indicate that this tree is primarily about "windows," "window system," "graphics," "information," and "software," which are relatively specific concepts that are all closely related to each other. The similarities and differences between the five nodes of layer two can be further understood by examining the nodes of layer one that are connected to them. For example, while nodes 26 and 16 of layer two share their connections to multiple nodes of layer one, node 27 of layer one on "image" is strongly connected to node 26 of layer two but not to node 16 of layer two, and node 17 of layer one on "video" is strongly connected to node 16 of layer two but not to node 26 of layer two.

Figure 7: Analogous plot to Figure 6 for a tree on "religion," rooted at node 2 of the top-layer.

Following the branches of each tree shown in both figures, it is clear that the topics become more and more specific when moving along the tree from the top to bottom. Taking the tree on "religion" shown in Figure 7 for example, the root node splits into two nodes when moving from layers five to four: while the left node is still mainly about "religion," the right node is on "objective morality." When moving from layers four to three, node 5 of layer four splits into a node about "Christian" and another node about "Islamic." When moving from layers three to two, node 3 of layer three splits into a node about "God, Jesus, & Christian," and another node about "science, atheism, & question of the existence of God." When moving from layers two to one, all four nodes of layer two split into multiple topics, and they are all strongly connected to both topics 1 and 2 of layer one, whose top words are those that appear frequently in the 20newsgroups corpus.

2.3.3 Visualizing Subnetworks Consisting of Related Trees

Examining the top-layer topics shown in Figure 5, one may find that some of the nodes seem to be closely related to each other. For example, topics 3 and 11 share eleven words out of the top twelve ones; topics 15 and 23 both have "Israel" and "Jews" as their top two words; topics 16 and 18 are both related to "gun;" and topics 7, 13, and 26 all share "team(s)," "game(s)," "player(s)," "season," and "league."

To understand the relationships and distinctions between these related nodes, we construct subnetworks that include the trees rooted at them, as shown in Figures 17-20 in Appendix C. It is clear from Figure 17 that the top-layer topic 3 differs from topic 11 in that it is not only strongly connected to topic 2 of layer four on "car & bike," but also has a non-negligible connection to topic 27 of layer four on "sales." It is clear from Figure 18

that topic 15 differs from topic 23 in that it is not only about "Israel & Arabs," but also about "Israel, Armenia, & Turkey." It is clear from Figure 19 in that topic 16 differs from topic 18 in that it is mainly about Waco siege happened in 1993 involving David Koresh, the Federal Bureau of Investigation (FBI), and the Bureau of Alcohol, Tobacco, Firearms and Explosives (BATF). It is clear from Figure 20 that topics 7 and 13 are mainly about "ice hockey" and "baseball," respectively, and topic 26 is a mixture of both.

2.3.4 Capturing Correlations Between Nodes

For the augmentable GBN, as in (18), given the weight vector $\boldsymbol{\theta}_j^{(1)}$, we have

$$\mathbb{E}\big[\boldsymbol{x}_j^{(1)} \,\big|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_j^{(1)}\big] = \boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_j^{(1)}. \tag{15}$$

A distinction between a shallow augmentable GBN with $T = 1$ hidden layer and a deep augmentable GBN with $T \geq 2$ hidden layers is that the prior for $\boldsymbol{\theta}_j^{(1)}$ changes from $\boldsymbol{\theta}_j^{(1)} \sim \mathrm{Gam}(\boldsymbol{r}, 1/c_j^{(2)})$ for $T = 1$ to $\boldsymbol{\theta}_j^{(1)} \sim \mathrm{Gam}(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_j^{(2)}, 1/c_j^{(2)})$ for $T \geq 2$. For the GBN with $T = 1$, given the shared weight vector $\boldsymbol{r}$, we have

$$\mathbb{E}\big[\boldsymbol{x}_j^{(1)} \,\big|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{r}\big] = \boldsymbol{\Phi}^{(1)}\boldsymbol{r}/c_j^{(2)}; \tag{16}$$

for the GBN with $T = 2$, given the shared weight vector $\boldsymbol{r}$, we have

$$\mathbb{E}\big[\boldsymbol{x}_j^{(1)} \,\big|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(2)}, \boldsymbol{r}\big] = \boldsymbol{\Phi}^{(1)}\boldsymbol{\Phi}^{(2)}\boldsymbol{r} \Big/ \left(c_j^{(2)} c_j^{(3)} \right); \tag{17}$$

and for the GBN with $T \geq 2$, given the weight vector $\boldsymbol{\theta}_j^{(2)}$, we have

$$\mathbb{E}\big[\boldsymbol{x}_j^{(1)} \,\big|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\Phi}^{(2)}, \boldsymbol{\theta}_j^{(2)}\big] = \boldsymbol{\Phi}^{(1)}\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_j^{(2)}/c_j^{(2)}. \tag{18}$$

Thus in the prior, the co-occurrence patterns of the columns of $\boldsymbol{\Phi}^{(1)}$ are modeled by only a single vector $\boldsymbol{r}$ when $T = 1$, but are captured in the columns of $\boldsymbol{\Phi}^{(2)}$ when $T \geq 2$. Similarly, in the prior, if $T \geq t + 1$, the co-occurrence patterns of the K_t columns of the projected topics $\prod_{\ell=1}^{t} \boldsymbol{\Phi}^{(\ell)}$ will be captured in the columns of the $K_t \times K_{t+1}$ matrix $\boldsymbol{\Phi}^{(t+1)}$.

To be more specific, we show in Figure 21 in Appendix C three example trees rooted at three different nodes of layer three, where we lower the threshold to $\tau_t = 1$ to reveal more weak links between the nodes of adjacent layers. The top subplot reveals that, in addition to strongly co-occurring with the top two topics of layer one, topic 21 of layer one on "medicine" tends to co-occur not only with topics 7, 21, and 26, which are all common topics that frequently appear, but also with some much less common topics that are related to very specific diseases or symptoms, such as topic 67 on "msg" and "Chinese restaurant syndrome," topic 73 on "candida yeast symptoms," and topic 180 on "acidophilous" and "astemizole (hismanal)."

The middle subplot reveals that topic 31 of layer two on "encryption & cryptography" tends to co-occur with topic 13 of layer two on "government & encryption," and it also indicates that topic 31 of layer one is more purely about "encryption" and more isolated from "government" in comparison to the other topics of layer one.

The bottom subplot reveals that in layer one, topic 14 on "law & government," topic 32 on "Israel & Lebanon," topic 34 on "Turkey, Armenia, Soviet Union, & Russian," topic 132 on "Greece, Turkey, & Cyprus," topic 98 on "Bosnia, Serbs, & Muslims," topic 143 on "Armenia, Azeris, Cyprus, Turkey, & Karabakh," and several other very specific topics related to Turkey and/or Armenia all tend to co-occur with each other.

We note that capturing the co-occurrence patterns between the topics not only helps exploratory data analysis, but also helps extract better features for classification in an unsupervised manner and improves prediction for held-out data, as will be demonstrated in detail in Section 4.

2.4 Related Models

The structure of the augmentable GBN resembles the sigmoid belief network and recently proposed deep exponential family model (Ranganath et al., 2014b). Such kind of gamma distribution based network and its inference procedure were vaguely hinted in Corollary 2 of Zhou and Carin (2015), and had been exploited by Acharya et al. (2015) to develop a gamma Markov chain to model the temporal evolution of the factor scores of a dynamic count matrix, but have not yet been investigated for extracting multilayer data representations. The proposed augmentable GBN may also be considered as an exponential family harmonium (Welling et al., 2004; Xing et al., 2005).

2.4.1 SIGMOID AND DEEP BELIEF NETWORKS

Under the hierarchical model in (1), given the connection weight matrices, the joint distribution of the observed/latent counts and gamma hidden units of the GBN can be expressed, similar to those of the sigmoid and deep belief networks (Bengio et al., 2015), as

$$
P\left(\boldsymbol{x}_j^{(1)}, \{\boldsymbol{\theta}_j^{(t)}\}_t \,\Big|\, \{\boldsymbol{\Phi}^{(t)}\}_t\right) = P\left(\boldsymbol{x}_j^{(1)} \,\Big|\, \boldsymbol{\Phi}^{(1)}, \boldsymbol{\theta}_j^{(1)}\right) \left[\prod_{t=1}^{T-1} P\left(\boldsymbol{\theta}_j^{(t)} \,\Big|\, \boldsymbol{\Phi}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}\right)\right] P\left(\boldsymbol{\theta}_j^{(T)}\right).
$$

With $\boldsymbol{\phi}_{v:}$ representing the vth row $\boldsymbol{\Phi}$, for the gamma hidden units $\theta_{vj}^{(t)}$ we have

$$
P\left(\theta_{vj}^{(t)} \,\Big|\, \boldsymbol{\phi}_{v:}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}, c_{j+1}^{(t+1)}\right) = \frac{\left(c_{j+1}^{(t+1)}\right)^{\boldsymbol{\phi}_{v:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}}}{\Gamma\left(\boldsymbol{\phi}_{v:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}\right)} \left(\theta_{vj}^{(t)}\right)^{\boldsymbol{\phi}_{v:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}-1} e^{-c_{j+1}^{(t+1)}\theta_{vj}^{(t)}}, \quad (19)
$$

which are highly nonlinear functions that are strongly desired in deep learning. By contrast, with the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ and bias terms $b_v^{(t+1)}$, a sigmoid/deep belief network would connect the binary hidden units $\theta_{vj}^{(t)} \in \{0, 1\}$ of layer t (for deep belief networks, $t < T-1$) to the product of the connection weights and binary hidden units of the next layer with

$$
P\left(\theta_{vj}^{(t)} = 1 \,\Big|\, \boldsymbol{\phi}_{v:}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}, b_v^{(t+1)}\right) = \sigma\left(b_v^{(t+1)} + \boldsymbol{\phi}_{v:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}\right). \quad (20)
$$

Comparing (19) with (20) clearly shows the distinctions between the gamma distributed nonnegative hidden units and the sigmoid link function based binary hidden units. The

limitation of binary units in capturing the approximately linear data structure over small ranges is a key motivation for Frey and Hinton (1999) to investigate nonlinear Gaussian belief networks with real-valued units. As a new alternative to binary units, it would be interesting to further investigate whether the gamma distributed nonnegative real units can in theory carry richer information and model more complex nonlinearities given the same network structure. Note that the rectified linear units have emerged as powerful alternatives of sigmoid units to introduce nonlinearity (Nair and Hinton, 2010). It would be interesting to investigate whether the gamma units can be used to introduce nonlinearity into the positive region of the rectified linear units.

2.4.2 Deep Poisson Factor Analysis

With $T = 1$, the PGBN specified by (1)-(3) and (8) reduces to Poisson factor analysis (PFA) using the (truncated) gamma-negative binomial process (Zhou and Carin, 2015), with a truncation level of K_1. As discussed in (Zhou et al., 2012; Zhou and Carin, 2015), with priors imposed on neither $\boldsymbol{\phi}_k^{(1)}$ nor $\boldsymbol{\theta}_j^{(1)}$, PFA is related to nonnegative matrix factorization (Lee and Seung, 2001), and with the Dirichlet priors imposed on both $\boldsymbol{\phi}_k^{(1)}$ and $\boldsymbol{\theta}_j^{(1)}$, PFA is related to latent Dirichlet allocation (Blei et al., 2003).

Related to the PGBN and the dynamic model in (Acharya et al., 2015), the deep exponential family model of Ranganath et al. (2014b) also considers a gamma chain under Poisson observations, but it is the gamma scale parameters that are chained and factorized, which allows learning the network parameters using black box variational inference (Ranganath et al., 2014a). In the proposed PGBN, we chain the gamma random variables via the gamma shape parameters. Both strategies worth through investigation. We prefer chaining the shape parameters in this paper, which leads to efficient upward-downward Gibbs sampling via data augmentation and makes it clear how the latent counts are propagated across layers, as discussed in detail in the following sections. The sigmoid belief network has also been recently incorporated into PFA for deep factorization of count data (Gan et al., 2015a), however, that deep structure captures only the correlations between binary factor usage patterns but not the full connection weights. In addition, neither Ranganath et al. (2014b) nor Gan et al. (2015a) provide a principled way to learn the network structure, whereas the proposed GBN uses the gamma-negative binomial process together with a greedy layer-wise training strategy to automatically infer the widths of the hidden layers, which will be described in Section 3.3.

2.4.3 Correlated and Tree-Structured Topic Models

The PGBN with $T = 2$ can also be related to correlated topic models (Blei and Lafferty, 2006; Paisley et al., 2012; Chen et al., 2013; Ranganath and Blei, 2015; Linderman et al., 2015), which typically use the logistic normal distributions to replace the topic-proportion Dirichlet distributions used in latent Dirichlet allocation (Blei et al., 2003), capturing the co-occurrence patterns between the topics in the latent Gaussian space using a covariance matrix. By contrast, the PGBN factorizes the topic usage weights (not proportions) under the gamma likelihood, capturing the co-occurrence patterns between the topics of the first layer (i.e., the columns of $\boldsymbol{\Phi}^{(1)}$) in the columns of $\boldsymbol{\Phi}^{(2)}$, the latent weight matrix connecting the hidden units of layers two and one. For the PGBN, the computation does not involve

matrix inversion, which is often necessary for correlated topic models without specially structured covariance matrices, and scales linearly with the number of topics, hence it is suitable to be used to capture the correlations between hundreds of or thousands of topics.

As in Figures 6, 7, and 17-21, trees and subnetworks can be extracted from the inferred deep network to visualize the data. Tree-structured topic models have also been proposed before, such as those in Blei et al. (2010), Adams et al. (2010), and Paisley et al. (2015), but they usually artificially impose the tree structures to be learned, whereas the PGBN learns a directed network, from which trees and subnetworks can be extracted for visualization, without the need to specify the number of nodes per layer, restrict the number of branches per node, and forbid a node to have multiple parents.

3. Model Properties and Inference

Inference for the GBN shown in (1) appears challenging, because not only the conjugate prior is unknown for the shape parameter of a gamma distribution, but also the gradients are difficult to evaluate for the parameters of the (log) gamma probability density function, which, as in (19), includes the parameters inside the (log) gamma function. To address these challenges, we consider data augmentation (van Dyk and Meng, 2001) that introduces auxiliary variables to make it simple to compute the conditional posteriors of model parameters via the joint distribution of the auxiliary and existing random variables. We will first show that each gamma hidden unit can be linked to a Poisson distributed latent count variable, leading to a negative binomial likelihood for the parameters of the gamma hidden unit if it is margined out from the Poisson distribution; we then introduce an auxiliary count variable, which is sampled from the CRT distribution parametrized by the negative binomial latent count and shape parameter, to make the joint likelihood of the auxiliary CRT count and latent negative binomial count given the parameters of the gamma hidden unit amenable to posterior simulation. More specifically, under the proposed augmentation scheme, the gamma shape parameters will be linked to auxiliary counts under the Poisson likelihoods, making it straightforward for posterior simulation, as described below in detail.

3.1 The Upward Propagation of Latent Counts

We break the inference of the GBN of T hidden layers into T related subproblems, each of which is solved with the same subroutine. Thus for implementation, it is straightforward for the GBN to adjust its depth T. Let us denote $\boldsymbol{x}_j^{(t)} \in \mathbb{Z}^{K_{t-1}}$ as the observed or latent count vector of layer $t \in \{1, \ldots, T\}$, and $x_{vj}^{(t)}$ as its vth element, where $v \in \{1, \ldots, K_{t-1}\}$.

Lemma 1 (Augment-and-Conquer The Gamma Belief Network) *With $p_j^{(1)} := 1 - e^{-1}$ and*

$$p_j^{(t+1)} := -\ln(1 - p_j^{(t)}) \Big/ \left[c_j^{(t+1)} - \ln(1 - p_j^{(t)}) \right] \tag{21}$$

for $t = 1, \ldots, T$, one may connect the observed or latent counts $\boldsymbol{x}_j^{(t)} \in \mathbb{Z}^{K_{t-1}}$ to the product $\boldsymbol{\Phi}^{(t)}\boldsymbol{\theta}_j^{(t)}$ at layer t under the Poisson likelihood as

$$\boldsymbol{x}_j^{(t)} \sim \text{Pois}\left[-\boldsymbol{\Phi}^{(t)}\boldsymbol{\theta}_j^{(t)} \ln\left(1 - p_j^{(t)}\right) \right]. \tag{22}$$

Proof By definition (22) is true for layer $t = 1$. Suppose that (22) is also true for layer $t > 1$, then we can augment each count $x_{vj}^{(t)}$, where $v \in \{1, \ldots, K_{t-1}\}$, into the summation of K_t latent counts, which are smaller than or equal to $x_{vj}^{(t)}$ as

$$x_{vj}^{(t)} = \sum_{k=1}^{K_t} x_{vjk}^{(t)}, \quad x_{vjk}^{(t)} \sim \text{Pois}\left[-\phi_{vk}^{(t)}\theta_{kj}^{(t)}\ln\left(1 - p_j^{(t)}\right)\right]. \tag{23}$$

Let the $\cdot$ symbol represent summing over the corresponding index and let

$$m_{kj}^{(t)(t+1)} := x_{\cdot jk}^{(t)} := \sum_{v=1}^{K_{t-1}} x_{vjk}^{(t)}$$

represent the number of times that factor $k \in \{1, \ldots, K_t\}$ of layer t appears in observation j and $\boldsymbol{m}_j^{(t)(t+1)} := \left(x_{\cdot j1}^{(t)}, \ldots, x_{\cdot jK_t}^{(t)}\right)'$. Since $\sum_{v=1}^{K_{t-1}} \phi_{vk}^{(t)} = 1$, we can marginalize out $\boldsymbol{\Phi}^{(t)}$ as in (Zhou et al., 2012), leading to

$$\boldsymbol{m}_j^{(t)(t+1)} \sim \text{Pois}\left[-\boldsymbol{\theta}_j^{(t)}\ln\left(1 - p_j^{(t)}\right)\right].$$

Further marginalizing out the gamma distributed $\boldsymbol{\theta}_j^{(t)}$ from the Poisson likelihood leads to

$$\boldsymbol{m}_j^{(t)(t+1)} \sim \text{NB}\left(\boldsymbol{\Phi}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}, p_j^{(t+1)}\right). \tag{24}$$

Element k of $\boldsymbol{m}_j^{(t)(t+1)}$ can be augmented under its compound Poisson representation as

$$m_{kj}^{(t)(t+1)} = \sum_{\ell=1}^{x_{kj}^{(t+1)}} u_\ell, \quad u_\ell \sim \text{Log}(p_j^{(t+1)}), \quad x_{kj}^{(t+1)} \sim \text{Pois}\left[-\phi_{k:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}\ln\left(1 - p_j^{(t+1)}\right)\right].$$

Thus if (22) is true for layer t, then it is also true for layer $t + 1$. $\blacksquare$

Corollary 2 (Propagate the latent counts upward) *Using Lemma 4.1 of (Zhou et al., 2012) on (23) and Theorem 1 of (Zhou and Carin, 2015) on (24), we can propagate the latent counts $x_{vj}^{(t)}$ of layer t upward to layer $t + 1$ as*

$$\left\{\left(x_{vj1}^{(t)}, \ldots, x_{vjK_t}^{(t)}\right) \middle| x_{vj}^{(t)}, \boldsymbol{\phi}_{v:}^{(t)}, \boldsymbol{\theta}_j^{(t)}\right\} \sim \text{Mult}\left(x_{vj}^{(t)}, \frac{\phi_{v1}^{(t)}\theta_{1j}^{(t)}}{\sum_{k=1}^{K_t}\phi_{vk}^{(t)}\theta_{kj}^{(t)}}, \ldots, \frac{\phi_{vK_t}^{(t)}\theta_{K_tj}^{(t)}}{\sum_{k=1}^{K_t}\phi_{vk}^{(t)}\theta_{kj}^{(t)}}\right), \tag{25}$$

$$\left(x_{kj}^{(t+1)} \middle| m_{kj}^{(t)(t+1)}, \boldsymbol{\phi}_{k:}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}\right) \sim \text{CRT}\left(m_{kj}^{(t)(t+1)}, \boldsymbol{\phi}_{k:}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)}\right). \tag{26}$$

We provide a set of graphical representations in Figure 8 to describe the GBN model and illustrate the augment-and-conquer inference scheme. We provide the upward-downward Gibbs sampler in Appendix B.
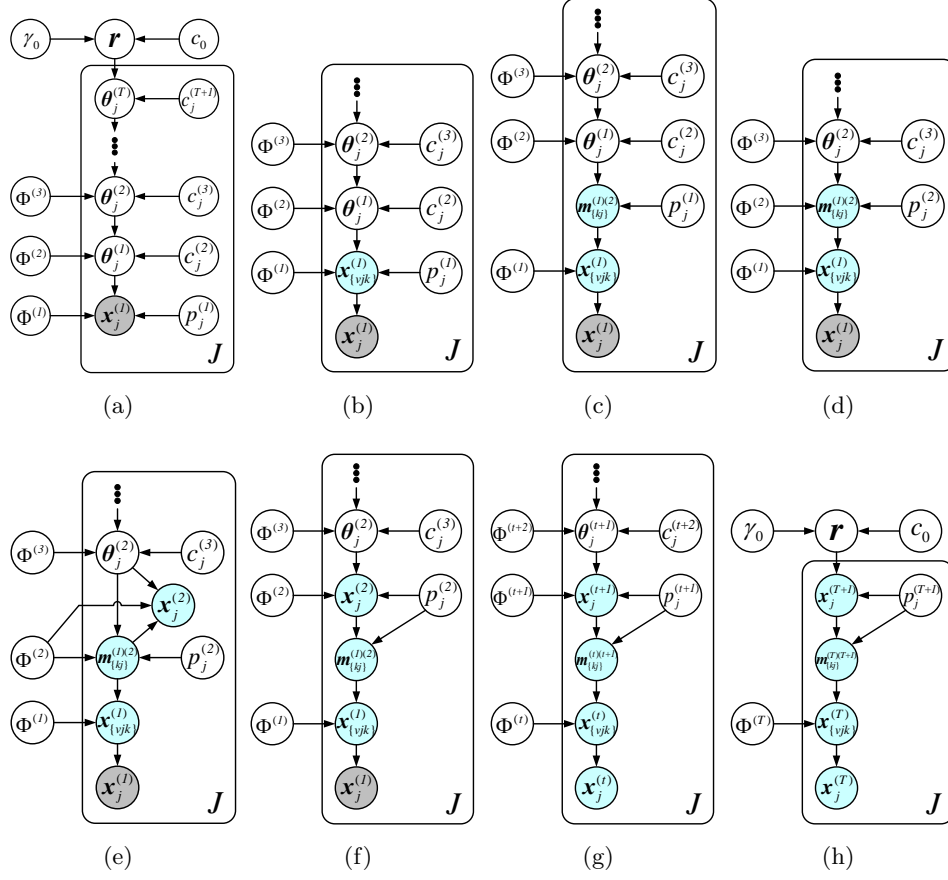
Figure 8: Graphical representations of the model and data augmentation and marginalization based inference scheme. (a) graphical representation of the GBN hierarchical model. (b) an augmented representation of Poisson factor model of layer $t = 1$, corresponding to (23) with $t = 1$. (c) an alternative representation using the relationships between the Poisson and multinomial distributions, obtained by applying Lemma 4.1 of (Zhou et al., 2012) on (23) for $t = 1$. (d) a negative binomial distribution based representation that marginalizes out the gamma from the Poisson distributions, corresponding to (24) for $t = 1$. (e) an equivalent representation that introduces CRT distributed auxiliary variables, corresponding to (26) with $t = 1$. (f) an equivalent representation using Theorem 1 of (Zhou and Carin, 2015) on (24) and (26) for $t = 1$. (g) An representation obtained by repeating the same augmentation-marginalization steps described in (b)-(f) one layer at a time from layers 1 to t. (h) An representation of the top hidden layer.

Note that $x_{\cdot j}^{(t)} = m_{\cdot j}^{(t)(t+1)}$, and as the number of tables occupied by the customers is in the same order as the logarithm of the customer number in a Chinese restaurant process, $x_{kj}^{(t+1)}$ is in the same order as $\ln\left(m_{kj}^{(t)(t+1)}\right)$. Thus the total count of layer $t+1$ as $\sum_j x_{\cdot j}^{(t+1)}$ would often be much smaller than that of layer t as $\sum_j x_{\cdot j}^{(t)}$ (though in general not as small as a count that is in the same order of the logarithm of $\sum_j x_{\cdot j}^{(t)}$), and hence one may use the total count $\sum_j x_{\cdot j}^{(T)}$ as a simple criterion to decide whether it is necessary to add more

layers to the GBN. In addition, if the latent count $x_{k' \cdot k}^{(t)} := \sum_j x_{k'jk}^{(t)}$ becomes close or equal to zero, then the posterior mean of $\mathbf{\Phi}^{(t)}(k', k)$ could become so small that node k' of layer $t - 1$ can be considered to be disconnected from node k of layer t.

3.2 Modeling Data Variability With Distributed Representation

In comparison to a single-layer model with $T = 1$, which assumes that the hidden units of layer one are independent in the prior, the multilayer model with $T \geq 2$ captures the correlations between them. Note that for the extreme case that $\mathbf{\Phi}^{(t)} = \mathbf{I}_{K_t}$ for $t \geq 2$ are all identity matrices, which indicates that there are no correlations between the features of $\boldsymbol{\theta}_j^{(t-1)}$ left to be captured, the deep structure could still provide benefits as it helps model latent counts $\boldsymbol{m}_j^{(1)(2)}$ that may be highly overdispersed. For example, let us assume $\mathbf{\Phi}^{(t)} = \mathbf{I}_{K_2}$ for all $t \geq 2$, then from (1) and (24) we have

$$m_{kj}^{(1)(2)} \sim \mathrm{NB}(\theta_{kj}^{(2)}, p_j^{(2)}), \quad \ldots, \quad \theta_{kj}^{(t)} \sim \mathrm{Gam}(\theta_{kj}^{(t+1)}, 1/c_j^{(t+1)}), \quad \ldots, \quad \theta_{kj}^{(T)} \sim \mathrm{Gam}(r_k, 1/c_j^{(T+1)}).$$

Using the laws of total expectation and total variance, we have

$$\mathbb{E}\big[\theta_{kj}^{(2)} \mid r_k\big] = \frac{r_k}{\prod_{t=3}^{T+1} c_j^{(t)}}, \qquad \mathrm{var}\big[\theta_{kj}^{(2)} \mid r_k\big] = r_k \sum_{t=3}^{T+1} \left[\prod_{\ell=3}^{t} \left(c_j^{(\ell)}\right)^{-2}\right] \left[\prod_{\ell=t+1}^{T+1} \left(c_j^{(\ell)}\right)^{-1}\right].$$

Further applying the same laws, we have

$$\mathbb{E}\big[m_{kj}^{(1)(2)} \mid r_k\big] = \frac{r_k p_j^{(2)}}{\left(1 - p_j^{(2)}\right) \prod_{t=3}^{T+1} c_j^{(t)}},$$

$$\mathrm{var}\big[m_{kj}^{(1)(2)} \mid r_k\big] = \frac{r_k p_j^{(2)}}{\left(1 - p_j^{(2)}\right)^2 \prod_{t=3}^{T+1} c_j^{(t)}} \left\{1 + p_j^{(2)} \sum_{t=3}^{T+1} \left[\prod_{\ell=3}^{t} \left(c_j^{(\ell)}\right)^{-1}\right]\right\}.$$

Thus the variance-to-mean ratio (VMR) of the count $m_{kj}^{(1)(2)}$ given r_k can be expressed as

$$\mathrm{VMR}\big[m_{kj}^{(1)(2)} \mid r_k\big] = \frac{1}{\left(1 - p_j^{(2)}\right)} \left\{1 + p_j^{(2)} \sum_{t=3}^{T+1} \left[\prod_{\ell=3}^{t} \left(c_j^{(\ell)}\right)^{-1}\right]\right\}. \tag{27}$$

In comparison to PFA with $m_{kj}^{(1)(2)} \sim \mathrm{NB}(r_k, p_j^{(2)})$ given r_k, with a VMR of $1/(1 - p_j^{(2)})$, the GBN with T hidden layers, which mixes the shape of $m_{kj}^{(1)(2)} \sim \mathrm{NB}(\theta_{kj}^{(2)}, p_j^{(2)})$ with a chain of gamma random variables, increases $\mathrm{VMR}\big[m_{kj}^{(1)(2)} \mid r_k\big]$ by a factor of

$$1 + p_j^{(2)} \sum_{t=3}^{T+1} \left[\prod_{\ell=3}^{t} \left(c_j^{(\ell)}\right)^{-1}\right],$$

which is equal to

$$1 + (T - 1)p_j^{(2)}$$

if we further assume $c_j^{(t)} = 1$ for all $t \geq 3$. Therefore, by increasing the depth of the network to distribute the variability into more layers, the multilayer structure could increase its capacity to model data variability.

3.3 Learning The Network Structure With Layer-Wise Training

As jointly training all layers together is often difficult, existing deep networks are typically trained using a greedy layer-wise unsupervised training algorithm, such as the one proposed in (Hinton et al., 2006) to train the deep belief networks. The effectiveness of this training strategy is further analyzed in (Bengio et al., 2007). By contrast, the augmentable GBN has a simple Gibbs sampler to jointly train all its hidden layers, as described in Appendix B, and hence does not necessarily require greedy layer-wise training, but the same as these commonly used deep learning algorithms, it still needs to specify the number of layers and the width of each layer.

In this paper, we adopt the idea of layer-wise training for the GBN, not because of the lack of an effective joint-training algorithm that trains all layers together in each iteration, but for the purpose of learning the width of each hidden layer in a greedy layer-wise manner, given a fixed budget on the width of the first layer. The basic idea is to first train a GBN with a single hidden layer, $i.e.$, $T = 1$, for which we know how to use the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) to infer the posterior distribution of the number of active factors; we fix the width of the first layer K_1 with the number of active factors inferred at iteration B_1, prune all inactive factors of the first layer, and continue Gibbs sampling for another C_1 iterations. Now we describe the proposed recursive procedure to build a GBN with $T \geq 2$ layers. With a GBN of $T - 1$ hidden layers that has already been inferred, for which the hidden units of the top layer are distributed as $\boldsymbol{\theta}_j^{(T-1)} \sim \text{Gam}(\boldsymbol{r}, 1/c_j^{(T)})$, where $\boldsymbol{r} = (r_1, \ldots, r_{K_{T-1}})'$, we add another layer by letting $\boldsymbol{\theta}_j^{(T-1)} \sim \text{Gam}(\boldsymbol{\Phi}^{(T)}\boldsymbol{\theta}_j^{(T)}, 1/c_j^{(T)})$, $\boldsymbol{\theta}_j^{(T)} \sim \text{Gam}(\boldsymbol{r}, 1/c_j^{(T+1)})$, where $\boldsymbol{\Phi}^{(T)} \in \mathbb{R}_+^{K_{T-1} \times K_{T\max}}$ and $\boldsymbol{r}$ is redefined as $\boldsymbol{r} = (r_1, \ldots, r_{K_{T\max}})'$. The key idea is with latent counts $m_{kj}^{(T)(T+1)}$ upward propagated from the bottom data layer, one may marginalize out $\theta_{kj}^{(T)}$, leading to $m_{kj}^{(T)(T+1)} \sim \text{NB}(r_k, p_j^{(T+1)})$, $r_k \sim \text{Gam}(\gamma_0/K_{T\max}, 1/c_0)$, and hence can again rely on the shrinkage mechanism of a truncated gamma-negative binomial process to prune inactive factors (connection weight vectors, columns of $\boldsymbol{\Phi}^{(T)}$) of layer T, making K_T, the inferred layer width for the newly added layer, smaller than $K_{T\max}$ if $K_{T\max}$ is set to be sufficiently large. The newly added layer and all the layers below would be jointly trained, but with the structure below the newly added layer kept unchanged. Note that when $T = 1$, the GBN infers the number of active factors if $K_{1\max}$ is set large enough, otherwise, it still assigns the factors with different weights r_k, but may not be able to prune any of them. The details of the proposed layer-wise training strategies are summarized in Algorithm 1 for multivariate count data, and in Algorithm 2 for multivariate binary and nonnegative real data.

4. Experimental Results

In this section, we present experimental results for count, binary, and nonnegative real data.

4.1 Deep Topic Modeling

We first analyze multivariate count data with the Poisson gamma belief network (PGBN). We apply the PGBNs for topic modeling of text corpora, each document of which is represented as a term-frequency count vector. Note that the PGBN with a single hidden layer

is identical to the (truncated) gamma-negative binomial process PFA of Zhou and Carin (2015), which is a nonparametric Bayesian algorithm that performs similarly to the hierarchical Dirichlet process latent Dirichlet allocation of Teh et al. (2006) for text analysis, and is considered as a strong baseline. Thus we will focus on making comparison to the PGBN with a single layer, with its layer width set to be large to approximate the performance of the gamma-negative binomial process PFA. We evaluate the PGBNs' performance by examining both how well they unsupervisedly extract low-dimensional features for document classification, and how well they predict heldout word tokens. Matlab code will be available in http://mingyuanzhou.github.io/.

We use Algorithm 1 to learn, in a layer-wise manner, from the training data the connection weight matrices $\mathbf{\Phi}^{(1)}, \ldots, \mathbf{\Phi}^{(T_{\max})}$ and the top-layer hidden units' gamma shape parameters $\boldsymbol{r}$: to add layer T to a previously trained network with $T-1$ layers, we use B_T iterations to jointly train $\mathbf{\Phi}^{(T)}$ and $\boldsymbol{r}$ together with $\{\mathbf{\Phi}^{(t)}\}_{1,T-1}$, prune the inactive factors of layer T, and continue the joint training with another C_T iterations. We set the hyper-parameters as $a_0 = b_0 = 0.01$ and $e_0 = f_0 = 1$. Given the trained network, we apply the upward-downward Gibbs sampler to collect 500 MCMC samples after 500 burnins to estimate the posterior mean of the feature usage proportion vector $\boldsymbol{\theta}_j^{(1)}/\theta_{\cdot j}^{(1)}$ at the first hidden layer, for every document in both the training and testing sets.

4.1.1 FEATURE LEARNING FOR BINARY CLASSIFICATION

We consider the 20newsgroups data set that consists of 18,774 documents from 20 different news groups, with a vocabulary of size $K_0 = 61,188$. It is partitioned into a training set of 11,269 documents and a testing set of 7,505 ones. We first consider two binary classification tasks that distinguish between the *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware*, and between the *sci.electronics* and *sci.med* news groups. For each binary classification task, we remove a standard list of stop words and only consider the terms that appear at least five times, and report the classification accuracies based on 12 independent random trials. With the upper bound of the first layer's width set as $K_{1\max} \in \{25, 50, 100, 200, 400, 600, 800\}$, and $B_t = C_t = 1000$ and $\eta^{(t)} = 0.01$ for all t, we use Algorithm 1 to train a network with $T \in \{1, 2, \ldots, 8\}$ layers. Denote $\bar{\boldsymbol{\theta}}_j$ as the estimated K_1 dimensional feature vector for document j, where $K_1 \leq K_{1\max}$ is the inferred number of active factors of the first layer that is bounded by the pre-specified truncation level $K_{1\max}$. We use the L_2 regularized logistic regression provided by the LIBLINEAR package (Fan et al., 2008) to train a linear classifier on $\bar{\boldsymbol{\theta}}_j$ in the training set and use it to classify $\bar{\boldsymbol{\theta}}_j$ in the test set, where the regularization parameter is five-folder cross-validated on the training set from $(2^{-10}, 2^{-9}, \ldots, 2^{15})$.

As shown in Figure 9, modifying the PGBN from a single-layer shallow network to a multilayer deep one clearly improves the qualities of the unsupervisedly extracted feature vectors. In a random trial, with $K_{1\max} = 800$, we infer a network structure of $[K_1, \ldots, K_8] = [512, 154, 75, 54, 47, 37, 34, 29]$ for the first binary classification task, and $[K_1, \ldots, K_8] = [491, 143, 74, 49, 36, 32, 28, 26]$ for the second one. Figures 9(c)-(d) also show that increasing the network depth in general improves the performance, but the first-layer width clearly plays a critical role in controlling the ultimate network capacity. This insight is further illustrated below.
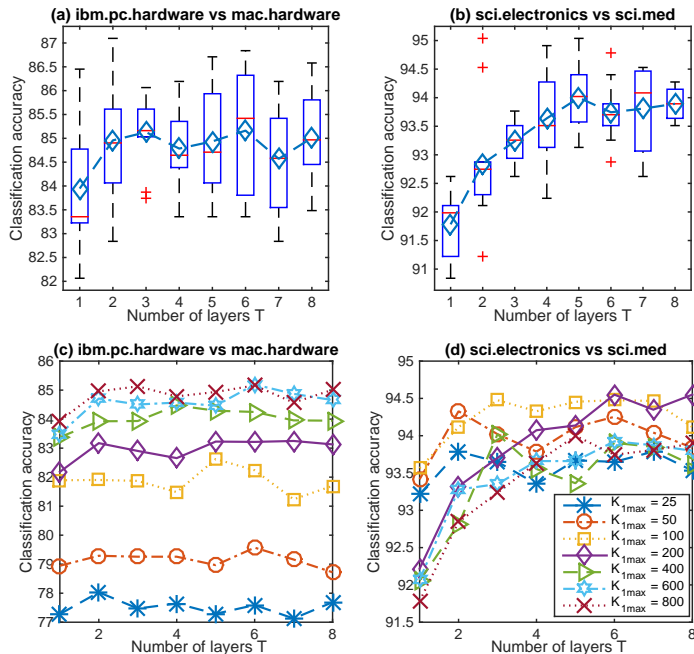
Figure 9: Classification accuracy (%) as a function of the network depth T for two 20newsgroups binary classification tasks, with $\eta^{(t)} = 0.01$ for all layers. (a)-(b): the boxplots of the accuracies of 12 independent runs with $K_{1\max} = 800$. (c)-(d): the average accuracies of these 12 runs for various $K_{1\max}$ and T. Note that $K_{1\max} = 800$ is large enough to cover all active first-layer topics (inferred to be around 500 for both binary classification tasks), whereas all the first-layer topics would be used if $K_{1\max} = 25$, 50, 100, or 200.

4.1.2 Feature Learning for Multi-Class Classification

We test the PGBNs for multi-class classification on 20newsgroups. After removing a standard list of stopwords and the terms that appear less than five times, we obtain a vocabulary with $V = 33,420$. We set $C_t = 500$ and $\eta^{(t)} = 0.05$ for all t; we set $B_t = 1000$ for all t if $K_{1\max} \leq 400$, and set $B_1 = 1000$ and $B_t = 500$ for $t \geq 2$ if $K_{1\max} > 400$. We use all 11,269 training documents to infer a set of networks with $T_{\max} \in \{1, \ldots, 5\}$ and $K_{1\max} \in \{50, 100, 200, 400, 600, 800\}$, and mimic the same testing procedure used for binary classification to extract low-dimensional feature vectors, with which each testing document is classified to one of the 20 news groups using the L_2 regularized logistic regression.

Figure 10 shows a clear trend of improvement in classification accuracy by increasing the network depth with a limited first-layer width, or by increasing the upper bound of the width of the first layer with the depth fixed. For example, a single-layer PGBN with $K_{1\max} = 100$ could add one or more layers to slightly outperform a single-layer PGBN with $K_{1\max} = 200$, and a single-layer PGBN with $K_{1\max} = 200$ could add layers to clearly outperform a single-layer PGBN with $K_{1\max}$ as large as 800.

The proposed Gibbs sampler also exhibits several desirable computational properties. Each iteration of jointly training multiple layers usually only costs moderately more than that of training a single layer, e.g., with $K_{1\max} = 400$, a training iteration on a single core of an Intel Xeon 2.7 GHz CPU takes about 5.6, 6.7, 7.1 seconds for the PGBN with 1, 3,
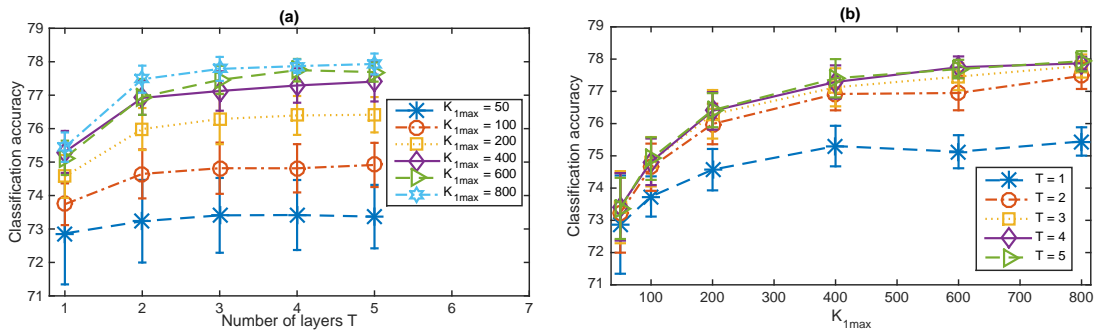
Figure 10: Classification accuracy (%) of the PGBNs with Algorithm 1 for 20newsgroups multi-class classification (a) as a function of the depth T with various $K_{1\max}$ and (b) as a function of $K_{1\max}$ with various depths, with $\eta^{(t)} = 0.05$ for all layers. The widths of the hidden layers are automatically inferred. In a random trial, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 99, 99, 94, 87]$, $[200, 161, 130, 94, 63]$, $[396, 109, 99, 82, 68]$, $[528, 129, 109, 98, 91]$, and $[608, 100, 99, 96, 89]$, respectively.
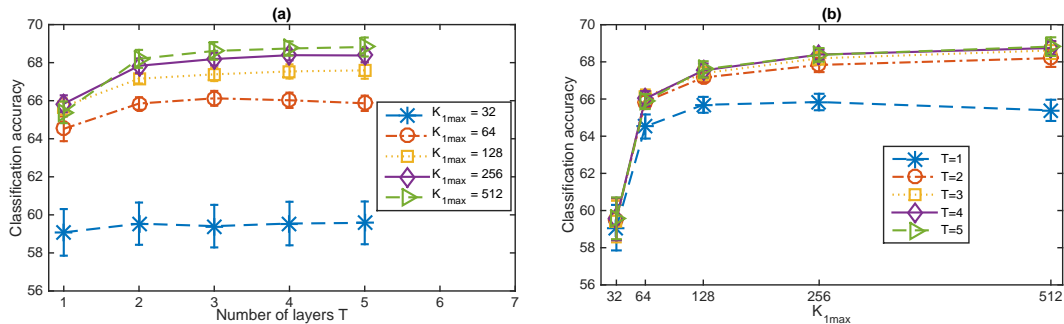
Figure 11: Analogous plots to Figure 10 with the vocabulary size restricted to be 2000, including the most frequent 2000 terms after removing a standard list of stopwords. The widths of the hidden layers are automatically inferred. In a random trial, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 32, 64, 128, 256$, and 512 are $[32, 32, 32, 32, 32]$, $[64, 64, 64, 59, 59]$, $[128, 125, 118, 106, 87]$, $[256, 224, 124, 83, 65]$, and $[512, 187, 89, 78, 62]$, respectively.

and 5 layers, respectively. Since the per iteration cost increases approximately as a linear function of the inferred K_1 and as a linear function of the size of the data set, given a fixed computational budget, one may choose a moderate $K_{1\max}$ to allow adding a sufficiently large number of hidden layers. In addition, the samplings of $x_{vkj}^{(t)}$, $\phi_k^{(t)}$, and $\theta_{kj}^{(t)}$ in each layer can all be made embarrassingly parallel with blocked Gibbs sampling, and hence can potentially significantly benefit from implementing the algorithm using graphics processing units (GPUs) or other parallel computing architectures.

Examining the inferred network structure also reveals interesting details. For example, in a random trial with Algorithm 1, with $\eta^{(t)} = 0.05$ for all t, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 99, 99, 94, 87]$, $[200, 161, 130, 94, 63]$, $[396, 109, 99, 82, 68]$, $[528, 129, 109, 98, 91]$, and $[608,$

100, 99, 96, 89], respectively. This indicates that for a network with an insufficient budget on its first-layer width, as the network depth increases, its inferred layer widths decay more slowly than a network with a sufficient or surplus budget on its first-layer width; and a network with a surplus budget on its first-layer width may only need relatively small widths for its higher hidden layers.

In order to make comparison to related algorithms, we also consider restricting the vocabulary to the 2000 most frequent terms of the vocabulary after moving a standard list of stopwords. We repeat the same experiments with the same settings except that we set $K_{1\max} \in \{32, 64, 128, 256, 512\}$, $B_1 = 1000$, $C_1 = 500$, and $B_t = C_t = 500$ for all $t \geq 2$. We show the results in Figure 11. Again, we observe a clear trend of improvement by increasing the network depth with a limited first-layer width, or by increasing the upper bound of the width of the first layer with the depth fixed. In a random trial with Algorithm 1, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 32, 64, 128, 256$, and 512 are $[32, 32, 32, 32, 32]$, $[64, 64, 64, 59, 59]$, $[128, 125, 118, 106, 87]$, $[256, 224, 124, 83, 65]$, and $[512, 187, 89, 78, 62]$, respectively.

For comparison, we first consider the same L_2 regularized logistic regression multiclass classifier, trained either on the raw word counts or normalized term-frequencies of the 20newsgroups training documents using five-folder cross-validation. As summarized in Table 1 of Appendix C, when using the raw term-frequency word counts as covariates, the same classifier achieves 69.8% (68.2%) accuracy on the 20newsgroups test documents if using the top 2000 terms that exclude (include) a standard list of stopwords, achieves 75.8% if using all the $61,188$ terms in the vocabulary, and achieves 78.0% if using the $33,420$ terms remained after removing a standard list of stopwords and the terms that appear less than five times; and when using the normalized term-frequencies as covariates, the corresponding accuracies are 70.8% (67.9%) if using the top 2000 terms excluding (including) stopwords, 77.6% with all the $61,188$ terms, and 79.4% with the $33,420$ selected terms.

As summarized in Table 2 of Appendix C, for multi-class classification on the same data set, with a vocabulary size of 2000 that consists of the 2000 most frequent terms after removing stopwords and stemming, the DocNADE (Larochelle and Lauly, 2012) and the over-replicated softmax (Srivastava et al., 2013) provide the accuracies of 67.0% and 66.8%, respectively, for a feature dimension of $K = 128$, and provide the accuracies of 68.4% and 69.1%, respectively, for a feature dimension of $K = 512$.

As shown in Figure 11 and summarized in Table 3 of Appendix C, with the same vocabulary size of 2000 (but different terms due to different preprocessing), the proposed PGBN provides 65.9% (67.5%) with $T = 1$ ($T = 5$) for $K_{1\max} = 128$, and 65.9% (69.2%) with $T = 1$ ($T = 5$) for $K_{1\max} = 512$, which may be further improved if we also consider the stemming step, as done in these two algorithms, for word preprocessing, or if we set the values of $\eta^{(t)}$ to be smaller than 0.05 to encourage a more complex network structure. We also summarize in Table 3 the classification accuracies shown in Figure 10 for the PGBNs with $V = 33,420$. Note that the accuracies in Tables 2 and 3 are provided to show that the PGBNs are in the same ballpark as both the DocNADE (Larochelle and Lauly, 2012) and over-replicated softmax (Srivastava et al., 2013). Note these results are not intended to provide a head-to-head comparison, which is possible if the same data preprocessing and classifier were used and the error bars were shown in Srivastava et al. (2013), or we could obtain the code to replicate the experiments using the same preprocessed data and classifier.

Note that 79.4% achieved using the 33, 420 selected features is the best accuracy reported in the paper, which is unsurprising since all the unsupervisedly extracted latent feature vectors have much lower dimensions and are not optimized for classification (Zhu et al., 2012; Zhou et al., 2015b). In comparison to using appropriately preprocessed high-dimensional features, our experiments show that while text classification performance often clearly deteriorates if one trains a multi-class classifier on the lower-dimensional features extracted using "shallow" unsupervised latent feature models, one could obtain much improved results using appropriate "deep" generalizations. For further improvement, one may consider adding an extra supervised component into the model, which is shown to boost the classification performance for latent Dirichlet allocation (Blei and Mcauliffe, 2008; Zhu et al., 2012), a shallow latent feature model related to the PGBN with a single hidden layer.

4.1.3 Perplexities for Heldout Words

In addition to examining the performance of the PGBN for unsupervised feature learning, we also consider a more direct approach that we randomly choose 30% of the word tokens in each document as training, and use the remaining ones to calculate per-heldout-word perplexity. We consider both all the 18,774 documents of the 20newsgroups corpus, limiting the vocabulary to the 2000 most frequent terms after removing a standard list of stopwords, and the NIPS12 (http://www.cs.nyu.edu/∼roweis/data.html) corpus whose stopwords have already been removed, limiting the vocabulary to the 2000 most frequent terms. We set $\eta^{(t)} = 0.05$ and $C_t = 500$ for all t, set $B_1 = 1000$ and $B_t = 500$ for $t \geq 2$, and consider five random trials. Among the $B_t + C_t$ Gibbs sampling iterations used to train layer t, we collect one sample per five iterations during the last 500 iterations, for each of which we draw the topics $\{\phi_k^{(1)}\}_k$ and topics weights $\theta_j^{(1)}$, to compute the per-heldout-word perplexity using Equation (34) of Zhou and Carin (2015). This evaluation method is similar to those used in Newman et al. (2009), Wallach et al. (2009), and Paisley et al. (2012).

As shown in both Figures 12 and 13, we observe a clear trend of improvement by increasing both $K_{1\,\mathrm{max}}$ and T. We have also examined the topics and network structure learned on the NIPS12 corpus. Similar to the exploratory data analysis performed on the 20newsgroups corpus, as described in detail in Section 2.3, the inferred deep networks also allow us to extract trees and subnetworks to visualize various aspects of the NIPS12 corpus from general to specific and reveal how they are related to each other. We omit these details for brevity and instead provide a brief description: with $K_{1\,\mathrm{max}} = 200$ and $T = 5$, the PGBN infers a network with $[K_1, \ldots, K_5] = [200, 164, 106, 60, 42]$ in one of the five random trials. The ranks, according to the weights $r_k^{(t)}$ calculated in (14), and the top five words of three example topics for layer $T = 5$ are "6 network units input learning training," "15 data model learning set image," and "34 network learning model input neural;" while these of five example topics of layer $T = 1$ are "19 likelihood em mixture parameters data," "37 bayesian posterior prior log evidence," "62 variables belief networks conditional inference," "126 boltzmann binary machine energy hinton," and "127 speech speaker acoustic vowel phonetic." It is clear that the topics of the bottom hidden layers are very specific whereas these of the top hidden layer are quite general.
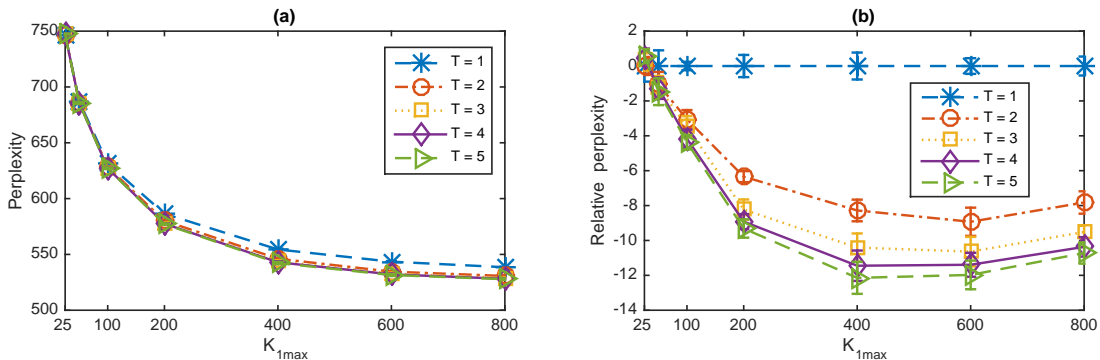
Figure 12: (a) per-heldout-word perplexity (the lower the better) for the NIPS12 corpus (using the 2000 most frequent terms) as a function of the upper bound of the first layer width $K_{1\max}$ and network depth T, with 30% of the word tokens in each document used for training and $\eta^{(t)} = 0.05$ for all t. (b) for visualization, each curve in (a) is reproduced by subtracting its values from the average perplexity of the single-layer network. In a random trial, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 25, 50, 100, 200, 400, 600$, and 800 are $[25, 25, 25, 25, 25]$, $[50, 50, 50, 49, 42]$, $[100, 99, 93, 78, 54]$, $[200, 164, 106, 60, 42]$, $[400, 130, 83, 52, 39]$, $[596, 71, 68, 58, 37]$, and $[755, 57, 53, 46, 42]$, respectively.
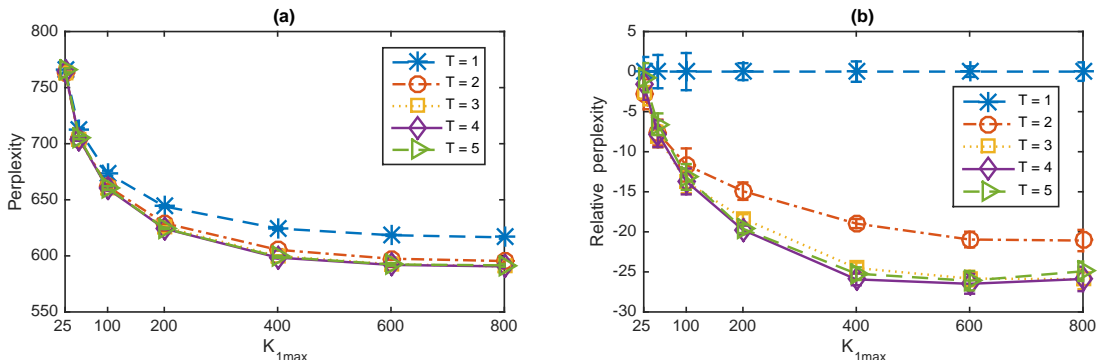
Figure 13: Analogous plots to Figure 12 for the 20newsgroups corpus (using the 2000 most frequent terms after removing a standard list of stopwords). In a random trial, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 25, 50, 100, 200, 400, 600$, and 800 are $[25, 25, 25, 25, 25]$, $[50, 50, 50, 50, 50]$, $[100, 99, 99, 97, 97]$, $[200, 194, 177, 152, 123]$, $[398, 199, 140, 116, 105]$, $[557, 156, 133, 118, 103]$, and $[701, 119, 116, 112, 103]$, respectively.

4.1.4 Generating Synthetic Documents

We have also tried drawing $\boldsymbol{\theta}_{j'}^{(T)} \sim \mathrm{Gam}\big(\boldsymbol{r}, 1/c_{j'}^{(T+1)}\big)$ and downward passing it through a T-layer network trained on a text corpus to generate synthetic bag-of-words documents, which are found to be quite interpretable and reflect various general aspects of the corpus used to train the network. We consider the PGBN with $[K_1, \ldots, K_5] = [608, 100, 99, 96, 89]$, which is trained on the training set of the 20newsgroups corpus with $K_{1\max} = 800$ and $\eta^{(t)} = 0.05$ for all t. We set $c_{j'}^{(t)}$ as the median of the inferred $\{c_j^{(t)}\}_j$ of the training doc-
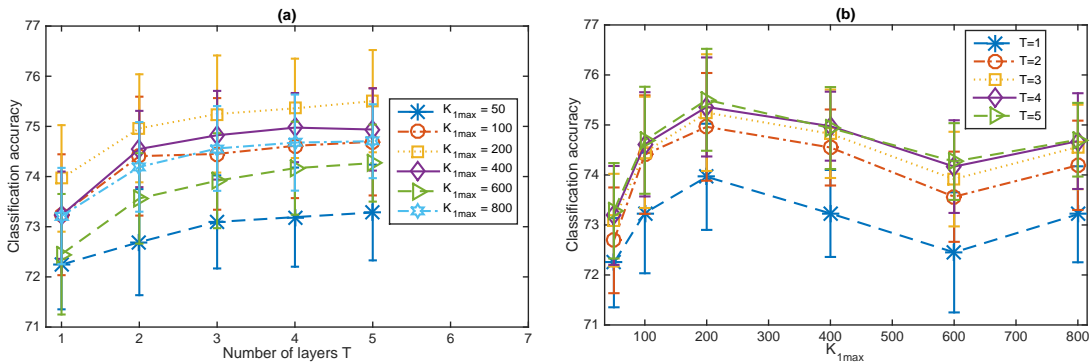
Figure 14: Analogous plots to Figure 10 for the BerPo-GBNs on the binarized 20news-groups term-document count matrix. The widths of the hidden layers are automatically inferred. In a random trial with Algorithm 2, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 97, 95, 90, 82]$, $[178, 145, 122, 97, 72]$, $[184, 139, 119, 101, 75]$, $[172, 165, 158, 138, 110]$, and $[156, 151, 147, 134, 117]$, respectively.

uments for all t. Given $\{\boldsymbol{\Phi}^{(t)}\}_{1,T}$ and $\boldsymbol{r}$, we first generate $\boldsymbol{\theta}_{j'}^{(T)} \sim \text{Gam}\left(\boldsymbol{r}, 1/c_{j'}^{(T+1)}\right)$ and then downward pass it through the network by drawing nonnegative real random variables, one layer after another, from the gamma distributions as in (1). With the simulated $\boldsymbol{\theta}_{j'}^{(1)}$, we calculate the Poisson rates for all the V words using $\boldsymbol{\Phi}^{(1)}\boldsymbol{\theta}_{j'}^{(1)}$ and display the top 100 words ranked by their Poisson rates. As shown in the text file available at http://mingyuanzhou.github.io/Results/GBN-BOW.txt, the synthetic documents generated in this manner are all easy to interpret and reflect various general aspects of the 20newsgroups corpus on which the PGBN is trained.

4.2 Multilayer Representation for Binary Data

We apply the BerPo-GBN to extract multilayer representations for high-dimensional sparse binary vectors. The BerPo link is proposed in Zhou (2015) to construct edge partition models for network analysis, whose computation is mainly spent on pairs of linked nodes and hence is scalable to big sparse relational networks. That link function and its inference procedure have also been adopted by Hu et al. (2015) to analyze big sparse binary tensors.

We consider the same problem of feature learning for multi-class classification studied in detail in Section 4.1.2. We consider the same setting except that the original term-document word count matrix is now binarized into a term-document indicator matrix, the (v, j) element of which is set as one if and only if $n_{vj} \geq 1$ and set as zero otherwise. We test the BerPo-GBNs on the 20newsgroups corpus, with $\eta^{(t)} = 0.05$ for all layers. As shown in Figure 14, given the same upper-bound on the width of the first layer, increasing the depth of the network clearly improves the performance. Whereas given the same number of hidden layers, the performance initially improves and then fluctuates as the upper-bound of the first layer increases. Such kind of fluctuations when $K_{1\max}$ reaches over 200 are expected, since the width of the first layer is inferred to be less than 190 and hence the budget as small as $K_{1\max} = 200$ is already large enough to cover all active factors.
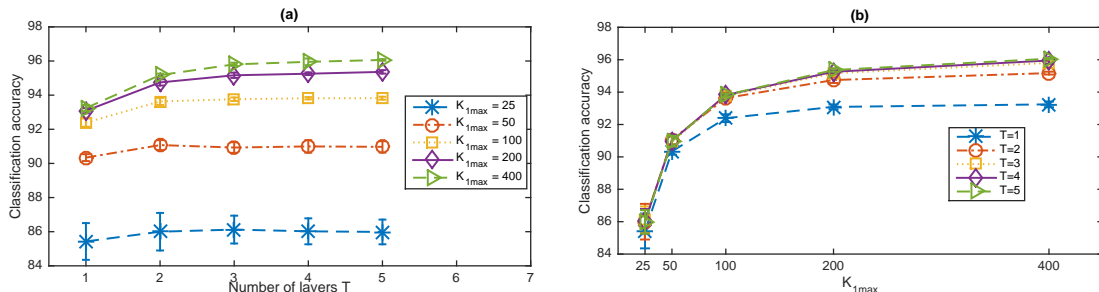
Figure 15: Analogous plots to Figure 10 for the PRG-GBNs on the MNIST data set. In a random trial with Algorithm 2, the inferred network widths $[K_1, \ldots, K_5]$ for $K_{1\,\text{max}}$=50, 100, 200, and 400 are $[50, 50, 50, 50, 50]$, $[100, 100, 100, 100, 100]$, $[200, 200, 200, 200, 200]$, and $[400, 400, 399, 385, 321]$, respectively.

4.3 Multilayer Representation for Nonnegative Real Data

We use the PRG-GBN to unsupervisedly extract features from nonnegative real data. We consider the MNIST data set (http://yann.lecun.com/exdb/mnist/), which consists of 60000 training handwritten digits and 10000 testing ones. We divide the gray-scale pixel values of each 28×28 image by 255 and represent each image as a 784 dimensional nonnegative real vector. We set $\eta^{(1)} = 0.05$ and use all training digits to infer the PRG-GBNs with $T_{max} \in \{1, \cdots, 5\}$ and $K_{1max} \in \{50, 100, 200, 400\}$. We consider the same problem of feature extraction for multi-class classification studied in detail in Section 4.1.2, and we follow the same experimental settings over there. As shown in Figure 15, both increasing the width of the first layer and the depth of the network could clearly improve the performance in terms of unsupervisedly extracting features that are better suited for multi-class classification.

Note that the PRG distribution might not be the best distribution to fit MNIST digits, but nevertheless, displaying the inferred features at various layers as images provides a straightforward way to visualize the latent structures inferred from the data and hence provides an excellent example to understand the properties and working mechanisms of the GBN. We display the projections to the first layer of the factors $\mathbf{\Phi}^{(t)}$ at all five hidden layers as images for $K_{1\,\text{max}} = 100$ and $K_{1\,\text{max}} = 400$ in Figures 22 and 23, respectively, which clearly show that the inferred latent factors become increasingly more general as the layer increases. In both Figures 22 and 23, the latent factors inferred at the first hidden layer represent filters that are only active at very particular regions of the images, those inferred at the second hidden layer represent larger parts of the hidden-written digits, and those inferred at the third and deeper layers resemble the whole digits.

To visualize the relationships between the factors of different layers, we show in Figure 24 in Appendix C a subset of nodes of each layer and the nodes of the layer below that are connected to them with non-negligible weights.

It is interesting to note that unlike Lee et al. (2009) and many other following works that rely on the convolutional and pooling operations, which are pioneered by LeCun et al. (1989), to extract hierarchical representation for images at different spatial scales, we show that the proposed algorithm, while not breaking the images into spatial patches, is already able to learn the factors that are active on very specific spatial regions of the image in

the bottom hidden layer, and learn these increasingly more general factors covering larger spatial regions of the images as the number of layer increases. However, due to the lack of the ability to discover spatially localized features that can be shared at multiple different spatial regions, our algorithm does not at all exploit the redundancies of the spatially localized features inside a single image and hence may require much more data to train. Therefore, it would be interesting to investigate whether one can introduce convolutional and pooling operations into the GBNs, which may substantially improve their performance on modeling natural images.

5. Conclusions

The augmentable gamma belief network (GBN) is proposed to extract a multilayer representation for high-dimensional count, binary, or nonnegative real vectors, with an efficient upward-downward Gibbs sampler to jointly train all its layers and a layer-wise training strategy to automatically infer the network structure. A GBN of T layers can be broken into T subproblems that are solved by repeating the same subroutine, with the computation mainly spent on training the first hidden layer. When used for deep topic modeling, the GBN extracts very specific topics at the first hidden layer and increasingly more general topics at deeper hidden layers. It provides an excellent way for exploratory data analysis through the visualization of the inferred deep network, whose hidden units of adjacent layers are sparsely connected. Its good performance is further demonstrated in unsupervisedly extracting features for document classification and predicting heldout word tokens. The extracted deep network can also be used to simulate very interpretable synthetic documents, which reflect various general aspects of the corpus that the network is trained on. When applied for image analysis, without using the convolutional and pooling operations, the GBN is already able to extract interpretable factors in the first hidden layer that are active in very specific spatial regions and interpretable factors in deeper hidden layers with increasingly more general spatial patterns covering larger spatial regions. For big data problems, in practice one may rarely have a sufficient budget to allow the first-layer width to grow without bound, thus it is natural to consider a deep network that can use a multilayer deep representation to better allocate its resource and increase its representation power with limited computational power. Our algorithm provides a natural solution to achieve a good compromise between the width of each layer and the depth of the network.

Acknowledgments
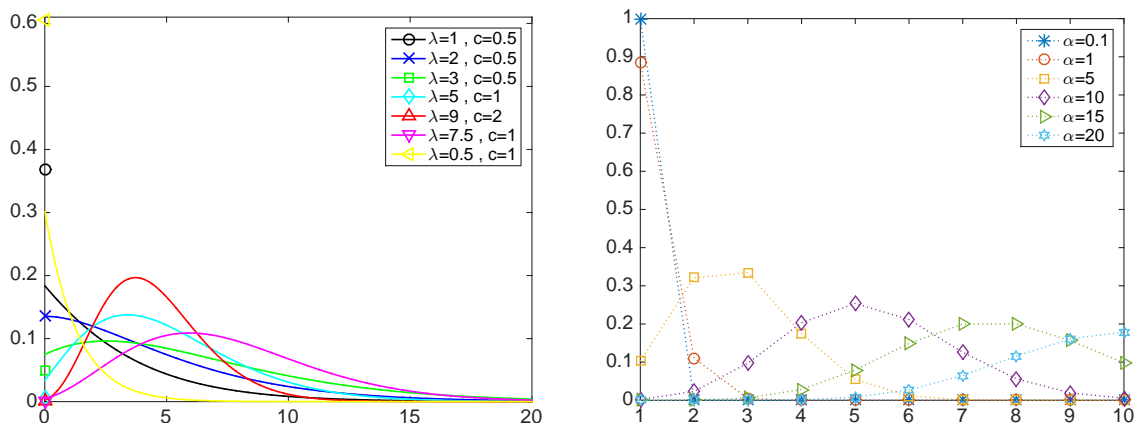
Figure 16: Left: probability distribution functions for the Poisson randomized gamma (PRG) distribution $x \sim \mathrm{PRG}(\lambda, c)$, where the sum of the probability mass at $x = 0$ and the area under the probability density function curve for $x > 0$ is equal to one; Right: probability mass functions for the truncated Bessel distribution $n \sim \mathrm{Bessel}_{-1}(\alpha)$, where $n \in \{1, 2, \ldots\}$.

Appendix A. Randomized Gamma and Bessel Distributions

Related to our work, Yuan and Kalbfleisch (2000) proposed the randomized gamma distribution to generate a random positive real number as

$$x \,|\, n, \nu \sim \mathrm{Gam}(n + \nu + 1, 1/c), \ n \sim \mathrm{Pois}(\lambda),$$

where $\nu > -1$ and $c > 0$. As in Yuan and Kalbfleisch (2000), the conditional posterior of n can be expressed as

$$(n \,|\, x, \nu, \alpha) \sim \mathrm{Bessel}_{\nu}(2\sqrt{cx\lambda})$$

where we denote $n \sim \mathrm{Bessel}_{\nu}(\alpha)$ as the Bessel distribution with parameters $\nu > -1$ and $\alpha > 0$, with PMF

$$\mathrm{Bessel}_{\nu}(n; \alpha) = \frac{\left(\frac{\alpha}{2}\right)^{2n+\nu}}{I_{\nu}(\alpha) n! \Gamma(n + \nu + 1)}, \ \ n \in \{0, 1, 2, \ldots\}.$$

Algorithms to draw Bessel random variables can be found in Devroye (2002).

The proposed PRG is different from the randomized gamma distribution of Yuan and Kalbfleisch (2000) in that it models both positive real numbers and exact zeros, and the proposed truncated Bessel distribution $n \sim \mathrm{Bessel}_{-1}(\alpha)$ is different from the Bessel distribution $n \sim \mathrm{Bessel}_{\nu}(\alpha)$, where $\nu > -1$, in that it is defined only on positive integers. For illustration, we show in Figure 16 the probability distribution functions of both the PRG and truncated Bessel distributions under a variety of parameter settings.

Appendix B. Upward-Downward Gibbs Sampling

Below we first discuss Gibbs sampling for count data and then generalize it for both binary and nonnegative real data.

B.1 Inference for the PGBN

With Lemma 1 and Corollary 2 and the width of the first layer being bounded by $K_{1\,\max}$, we first consider multivariate count observations and develop an upward-downward Gibbs sampler for the PGBN, each iteration of which proceeds as follows.

Sample $x_{vjk}^{(t)}$. We can sample $x_{vjk}^{(t)}$ for all layers using (25). But for the first hidden layer, we may treat each observed count $x_{vj}^{(1)}$ as a sequence of word tokens at the vth term (in a vocabulary of size $V := K_0$) in the jth document, and assign the $x_{\cdot j}^{(1)}$ words $\{v_{ji}\}_{i=1,x_{\cdot j}^{(1)}}$ one after another to the latent factors (topics), with both the topics $\mathbf{\Phi}^{(1)}$ and topic weights $\boldsymbol{\theta}_j^{(1)}$ marginalized out, as

$$P(z_{ji} = k \,|\, -) \propto \frac{\eta^{(1)} + x_{v_{ji}\cdot k}^{(1)^{-ji}}}{V\eta^{(1)} + x_{\cdot\cdot k}^{(1)^{-ji}}} \left(x_{\cdot jk}^{(1)^{-ji}} + \boldsymbol{\phi}_{k:}^{(2)}\boldsymbol{\theta}_j^{(2)} \right), \quad k \in \{1, \dots, K_{1\,\max}\}, \tag{28}$$

where z_{ji} is the topic index for v_{ji} and $x_{vjk}^{(1)} := \sum_i \delta(v_{ji} = v, z_{ji} = k)$ counts the number of times that term v appears in document j; we use x^{-ji} to denote the count x calculated without considering word i in document j. The collapsed Gibbs sampling update equation shown above is related to the one developed in (Griffiths and Steyvers, 2004) for latent Dirichlet allocation, and the one developed in (Zhou, 2014) for PFA using the beta-negative binomial process. When $T = 1$, we would replace the terms $\boldsymbol{\phi}_{k:}^{(2)}\boldsymbol{\theta}_j^{(2)}$ with r_k for PFA built on the gamma-negative binomial process (Zhou and Carin, 2015) (or with $\alpha\pi_k$ for hierarchical Dirichlet process latent Dirichlet allocation, see (Teh et al., 2006) and (Zhou, 2014) for details), and add an additional term to account for the possibility of creating an additional factor (Zhou, 2014). For simplicity, in this paper, we truncate the nonparametric Bayesian model with $K_{1\,\max}$ factors and let $r_k \sim \mathrm{Gam}(\gamma_0/K_{1\,\max}, 1/c_0)$ if $T = 1$. Note that although we use collapsed Gibbs sampling inference in this paper, if one desires embarrassingly parallel inference and possibly lower computation, then one may consider explicitly sampling $\{\boldsymbol{\phi}_k^{(1)}\}_k$ and $\{\boldsymbol{\theta}_j^{(1)}\}_j$ and sampling $x_{vjk}^{(1)}$ with (25).

Sample $\boldsymbol{\phi}_k^{(t)}$. Given these latent counts, we sample the factors/topics $\boldsymbol{\phi}_k^{(t)}$ as

$$(\boldsymbol{\phi}_k^{(t)} \,|\, -) \sim \mathrm{Dir}\left(\eta_1^{(t)} + x_{1\cdot k}^{(t)}, \dots, \eta_{K_{t-1}}^{(t)} + x_{K_{t-1}\cdot k}^{(t)} \right). \tag{29}$$

Sample $x_{vj}^{(t+1)}$. We sample $\boldsymbol{x}_j^{(t+1)}$ using (26), where we replace the term $\boldsymbol{\phi}_{v:}^{(T+1)}\boldsymbol{\theta}_j^{(T+1)}$ with r_v.

Sample $\boldsymbol{r}$. Both γ_0 and c_0 are sampled using related equations in (Zhou and Carin, 2015), omitted here for brevity. We sample $\boldsymbol{r}$ as

$$(r_v \,|\, -) \sim \mathrm{Gam}\left(\gamma_0/K_T + x_{v\cdot}^{(T+1)}, \left[c_0 - \sum_j \ln\left(1 - p_j^{(T+1)}\right) \right]^{-1} \right). \tag{30}$$

Sample $\boldsymbol{\theta}_j^{(t)}$. Using (22) and the gamma-Poisson conjugacy, we sample $\boldsymbol{\theta}_j$ as

$$(\boldsymbol{\theta}_j^{(T)} \,|\, -) \sim \mathrm{Gam}\left(\boldsymbol{r} + \boldsymbol{m}_j^{(T)(T+1)}, \left[c_j^{(T+1)} - \ln\left(1 - p_j^{(T)}\right)\right]^{-1}\right),$$

$$\vdots$$

$$(\boldsymbol{\theta}_j^{(t)} \,|\, -) \sim \mathrm{Gam}\left(\boldsymbol{\Phi}^{(t+1)}\boldsymbol{\theta}_j^{(t+1)} + \boldsymbol{m}_j^{(t)(t+1)}, \left[c_j^{(t+1)} - \ln\left(1 - p_j^{(t)}\right)\right]^{-1}\right),$$

$$\vdots$$

$$(\boldsymbol{\theta}_j^{(1)} \,|\, -) \sim \mathrm{Gam}\left(\boldsymbol{\Phi}^{(2)}\boldsymbol{\theta}_j^{(2)} + \boldsymbol{m}_j^{(1)(2)}, \left[c_j^{(2)} - \ln\left(1 - p_j^{(1)}\right)\right]^{-1}\right), \tag{31}$$

Sample $c_j^{(t)}$. With $\theta_{\cdot j}^{(t)} := \sum_{k=1}^{K_t} \theta_{kj}^{(t)}$ for $t \leq T$ and $\theta_{\cdot j}^{(T+1)} := r_\cdot$, we sample $p_j^{(2)}$ and $\{c_j^{(t)}\}_{t \geq 3}$ as

$$(p_j^{(2)} \,|\, -) \sim \mathrm{Beta}\left(a_0 + m_{\cdot j}^{(1)(2)}, b_0 + \theta_{\cdot j}^{(2)}\right), \quad (c_j^{(t)} \,|\, -) \sim \mathrm{Gam}\left(e_0 + \theta_{\cdot j}^{(t)}, \left[f_0 + \theta_{\cdot j}^{(t-1)}\right]^{-1}\right), \tag{32}$$

and calculate $c_j^{(2)}$ and $\{p_j^{(t)}\}_{t \geq 3}$ with (21).

B.2 Handling Binary and Nonnegative Real Observations

For binary observations that are linked to the latent counts at layer one as $b_{vj}^{(1)} = \mathbf{1}(x_{vj}^{(1)} \geq 1)$, we first sample the latent counts at layer one from the truncated Poisson distribution as

$$\left(x_{vj}^{(1)} \,|\, -\right) \sim b_{vj}^{(1)} \cdot \mathrm{Pois}_+\left(\sum_{k=1}^{K_1} \phi_{vk}^{(1)}\theta_{kj}^{(1)}\right) \tag{33}$$

and then sample $x_{vjk}^{(t)}$ for all layers using (25).

For nonnegative real observations $y_{vj}^{(1)}$ that are linked to the latent counts at layer one as

$$y_{vj}^{(1)} \sim \mathrm{Gam}(x_{vj}^{(1)}, 1/a_j),$$

we let $x_{vj}^{(1)} = 0$ if $y_{vj}^{(1)} = 0$ and sample $x_{vj}^{(1)}$ from the truncated Bessel distribution as

$$\left(x_{vj}^{(1)} \,|\, -\right) \sim \mathrm{Bessel}_{-1}\left(2\sqrt{a_j y_{vj}^{(1)} \sum_{k=1}^{K_1} \phi_{vk}^{(1)}\theta_{kj}^{(1)}}\right) \tag{34}$$

if $y_{vj}^{(1)} > 0$. We let $a_j \sim \mathrm{Gam}(e_0, 1/f_0)$ in the prior and sample a_j as

$$(a_j \,|\, -) \sim \mathrm{Gam}\left(e_0 + \sum_v x_{vj}^{(1)}, \frac{1}{f_0 + \sum_v y_{vj}^{(1)}}\right). \tag{35}$$

We then sample $x_{vjk}^{(t)}$ for all layers using (25).

Appendix C. Additional Tables and Figures

Algorithm 1 The PGBN upward-downward Gibbs sampler that uses a layer-wise training strategy to train a set of networks, each of which adds an additional hidden layer on top of the previously inferred network, retrains all its layers jointly, and prunes inactive factors from the last layer. **Inputs:** observed counts $\{x_{vj}\}_{v,j}$, upper bound of the width of the first layer $K_{1\,\max}$, upper bound of the number of layers $T_{\max}$, number of iterations $\{B_T, S_T\}_{1,T_{\max}}$, and hyper-parameters.
Outputs: A total of $T_{\max}$ jointly trained PGBNs with depths $T = 1$, $T = 2$, ..., and $T = T_{\max}$.

1: **for** $T = 1, 2, \ldots, T_{\max}$ **do** Jointly train all the T layers of the network
2:   Set K_{T-1}, the inferred width of layer $T-1$, as $K_{T\,\max}$, the upper bound of layer T's width.
3:   **for** $iter = 1 : B_T + C_T$ **do** Upward-downward Gibbs sampling
4:     Sample $\{z_{ji}\}_{j,i}$ using collapsed inference; Calculate $\{x_{vjk}^{(1)}\}_{v,k,j}$; Sample $\{x_{vj}^{(2)}\}_{v,j}$;
5:     **for** $t = 2, 3, \ldots, T$ **do**
6:       Sample $\{x_{vjk}^{(t)}\}_{v,j,k}$; Sample $\{\boldsymbol{\phi}_k^{(t)}\}_k$; Sample $\{x_{vj}^{(t+1)}\}_{v,j}$;
7:     **end for**
8:     Sample $p_j^{(2)}$ and Calculate $c_j^{(2)}$; Sample $\{c_j^{(t)}\}_{j,t}$ and Calculate $\{p_j^{(t)}\}_{j,t}$ for $t = 3, \ldots, T+1$;
9:     **for** $t = T, T-1, \ldots, 2$ **do**
10:       Sample $\boldsymbol{r}$ if $t = T$; Sample $\{\boldsymbol{\theta}_j^{(t)}\}_j$;
11:     **end for**
12:     **if** $iter = B_T$ **then**
13:       Prune layer T's inactive factors $\{\boldsymbol{\phi}_k^{(T)}\}_{k:x_{\cdot\cdot k}^{(T)}=0}$;
14:       let $K_T = \sum_k \delta(x_{\cdot\cdot k}^{(T)} > 0)$ and update $\boldsymbol{r}$;
15:     **end if**
16:   **end for**
17:   Output the posterior means (according to the last MCMC sample) of all remaining factors $\{\boldsymbol{\phi}_k^{(t)}\}_{k,t}$ as the inferred network of T layers, and $\{r_k\}_{k=1}^{K_T}$ as the gamma shape parameters of layer T's hidden units.
18: **end for**

Algorithm 2 The upward-downward Gibbs samplers for the Ber-GBN and PRG-GBN are constructed by using Lines 1-8 shown below to substitute Lines 4-11 of the PGBN Gibbs sampler shown in Algorithm 1.

1: Sample $\{x_{vj}^{(1)}\}_{v,j}$ using (33) for binary observations; Sample $\{x_{vj}^{(1)}\}_{v,j}$ using (34) and sample a_j using (35) for nonnegative real observations;
2: **for** $t = 1, 2, \ldots, T$ **do**
3:   Sample $\{x_{vjk}^{(t)}\}_{v,j,k}$; Sample $\{\boldsymbol{\phi}_k^{(t)}\}_k$; Sample $\{x_{vj}^{(t+1)}\}_{v,j}$;
4: **end for**
5: Sample $p_j^{(2)}$ and Calculate $c_j^{(2)}$; Sample $\{c_j^{(t)}\}_{j,t}$ and Calculate $\{p_j^{(t)}\}_{j,t}$ for $t = 3, \ldots, T+1$;
6: **for** $t = T, T-1, \ldots, 1$ **do**
7:   Sample $\boldsymbol{r}$ if $t = T$; Sample $\{\boldsymbol{\theta}_j^{(t)}\}_j$;
8: **end for**

$V = 61,188$ with stopwords with rare words raw word counts	$V = 61,188$ with stopwords with rare words term frequencies	$V = 33,420$ remove stopwords remove rare words raw word counts	$V = 33,420$ remove stopwords remove rare words term frequencies
75.8%	77.6%	78.0%	79.4%

$V = 2000$ with stopwords raw counts	$V = 2000$ with stopwords term frequencies	$V = 2000$ remove stopwords raw counts	$V = 2000$ remove stopwords term frequencies
68.2%	67.9%	69.8%	70.8%

Table 1: Multi-class classification accuracies of L_2 regularized logistic regression.

	$V = 2000$, $K = 128$ remove stopwords, stemming	$V = 2000$, $K = 512$ remove stopwords, stemming
DocNADE	67.0%	68.4%
Over-replicated softmax	66.8%	69.1%

Table 2: Multi-class classification accuracies of the DocNADE (Larochelle and Lauly, 2012) and over-replicated softmax (Srivastava et al., 2013).

	$V = 2000$, $K_{1\max} = 128$ remove stopwords	$V = 2000$, $K_{1\max} = 256$ remove stopwords	$V = 2000$, $K_{1\max} = 512$ remove stopwords
PGBN ($T = 1$)	$65.9\% \pm 0.4\%$	$66.3\% \pm 0.4\%$	$65.9\% \pm 0.4\%$
PGBN ($T = 2$)	$67.1\% \pm 0.5\%$	$67.9\% \pm 0.4\%$	$68.3\% \pm 0.3\%$
PGBN ($T = 3$)	$67.3\% \pm 0.3\%$	$68.6\% \pm 0.5\%$	$69.0\% \pm 0.4\%$
PGBN ($T = 5$)	$67.5\% \pm 0.4\%$	$68.8\% \pm 0.3\%$	$69.2\% \pm 0.4\%$

	$V = 33,420$, $K_{1\max} = 200$ remove stopwords remove rare words	$V = 33,420$, $K_{1\max} = 400$ remove stopwords remove rare words	$V = 33,420$, $K_{1\max} = 800$ remove stopwords remove rare words
PGBN ($T = 1$)	$74.6\% \pm 0.6\%$	$75.3\% \pm 0.6\%$	$75.4\% \pm 0.4\%$
PGBN ($T = 2$)	$76.0\% \pm 0.6\%$	$76.9\% \pm 0.5\%$	$77.5\% \pm 0.4\%$
PGBN ($T = 3$)	$76.3\% \pm 0.8\%$	$77.1\% \pm 0.6\%$	$77.8\% \pm 0.4\%$
PGBN ($T = 5$)	$76.4\% \pm 0.5\%$	$77.4\% \pm 0.6\%$	$77.9\% \pm 0.3\%$

Table 3: Multi-class classification accuracies of the PGBN trained with $\eta^t = 0.05$ for all t.
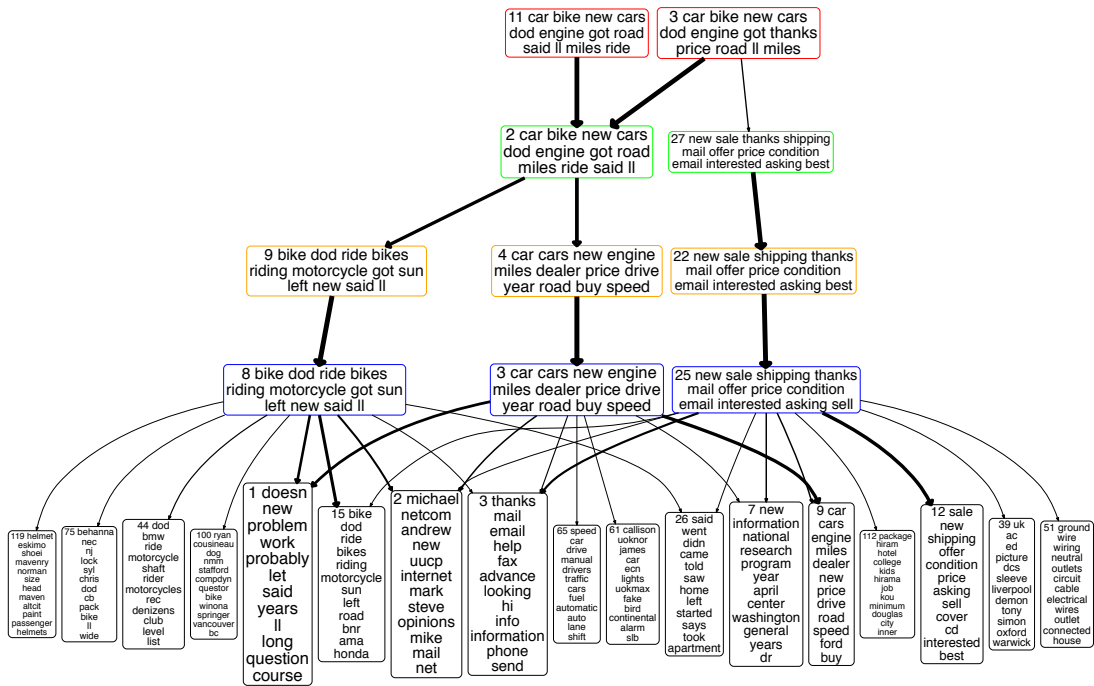
Figure 17: Analogous plots to Figure 6 for a subnetwork on "car & bike", consisting of two trees rooted at nodes 3 and 11, respectively, of layer one.
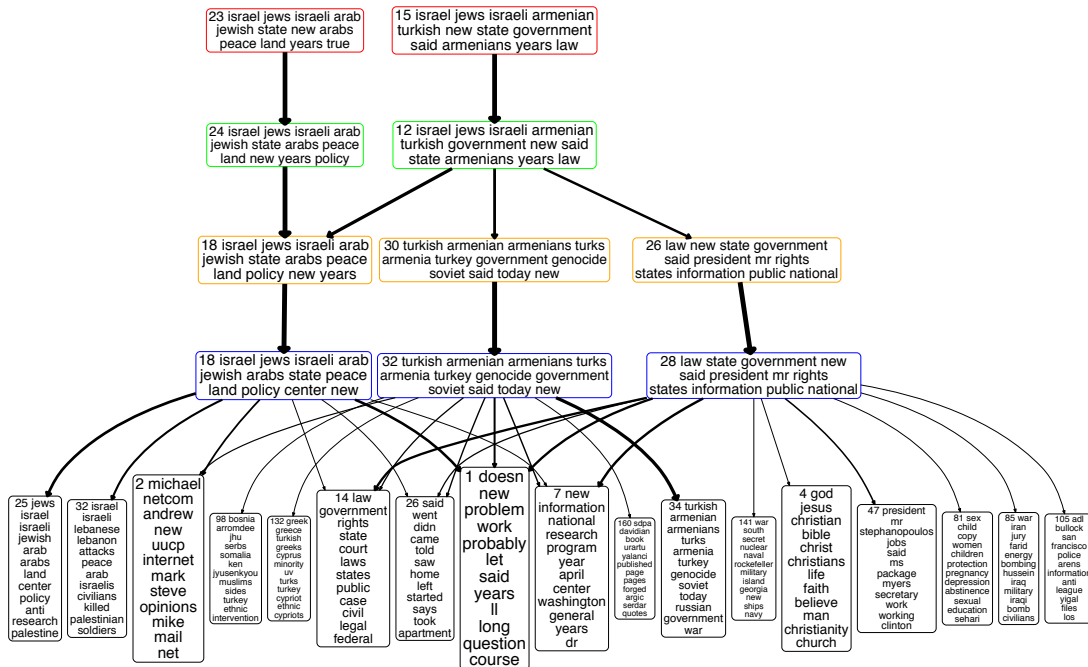
Figure 18: Analogous plot to Figure 6 for a subnetwork on "Middle East," consisting of two trees rooted at nodes 15 and 23, respectively, of layer one.
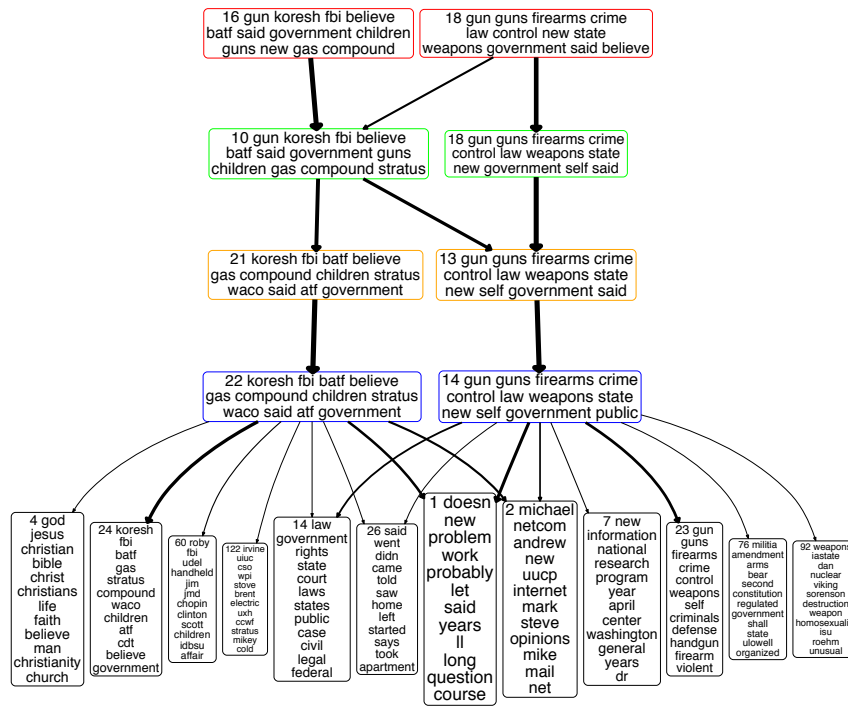
Figure 19: Analogous plot to Figure 6 for a subnetwork related to "gun," consisting of two trees rooted at nodes 16 and 18, respectively, of layer one.
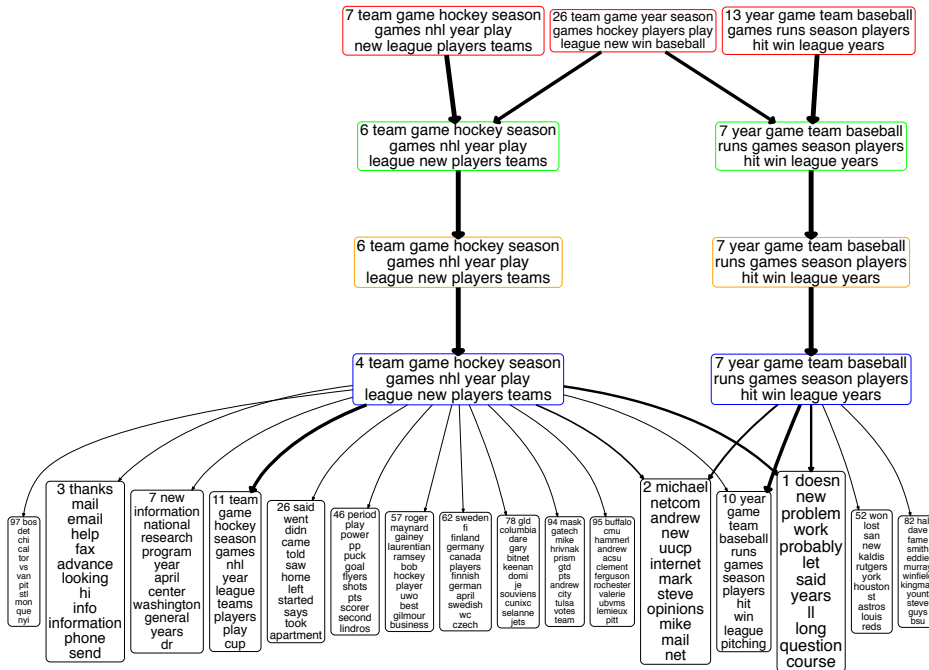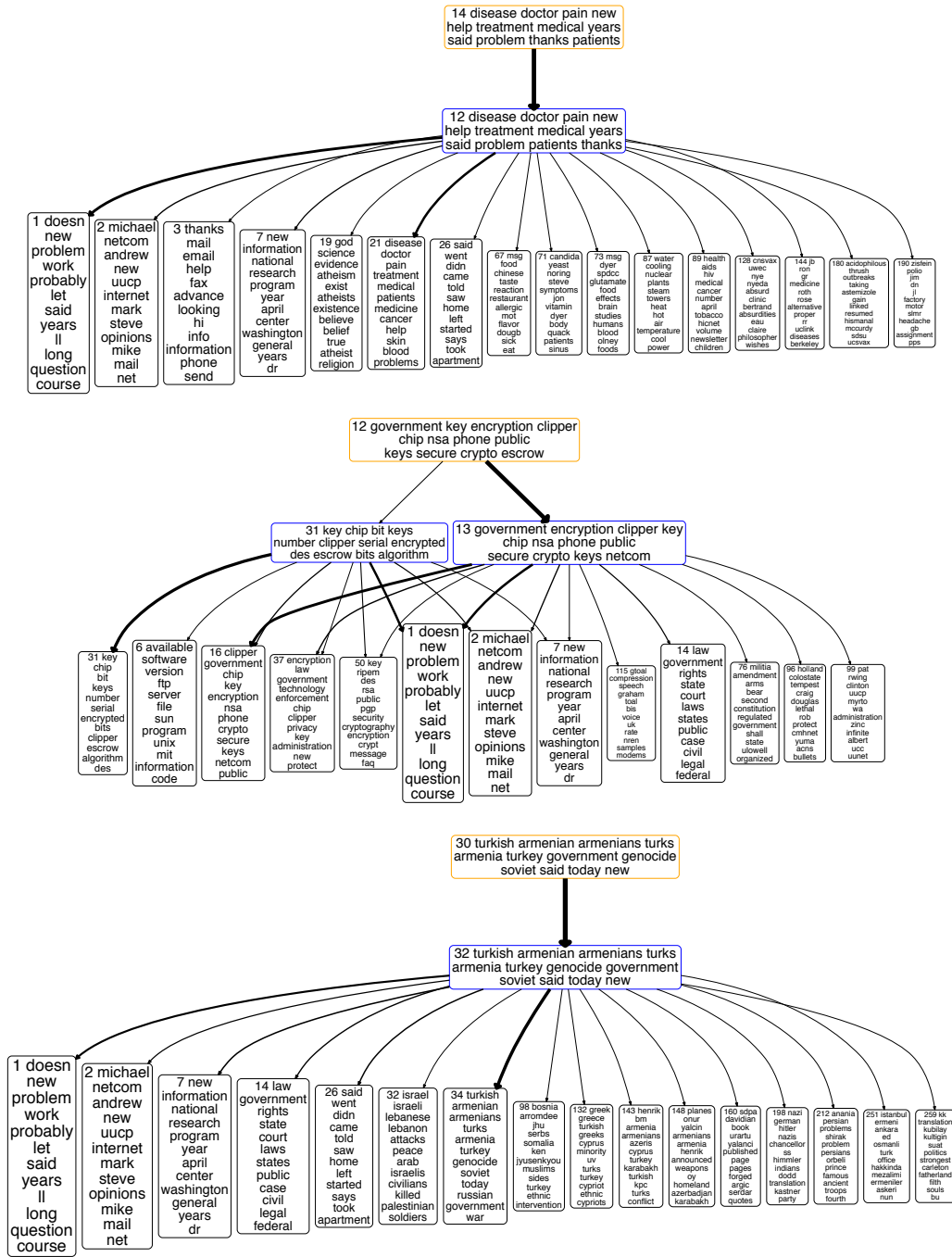

Figure 20: Analogous plot to Figure 6 for a subnetwork on "ice hockey" and "baseball," consisting of three trees rooted at nodes 7, 13, and 26, respectively, of layer one.

Figure 21: Analogous plots to Figure 6, with $\tau_t = 1$ to reveal more weak links. Top: the tree rooted at node 14 of layer three on "medicine." Middle: the tree rooted at node 12 of layer three on "encryption." Bottom: the tree rooted at node 30 of layer three on "Turkey & Armenia."
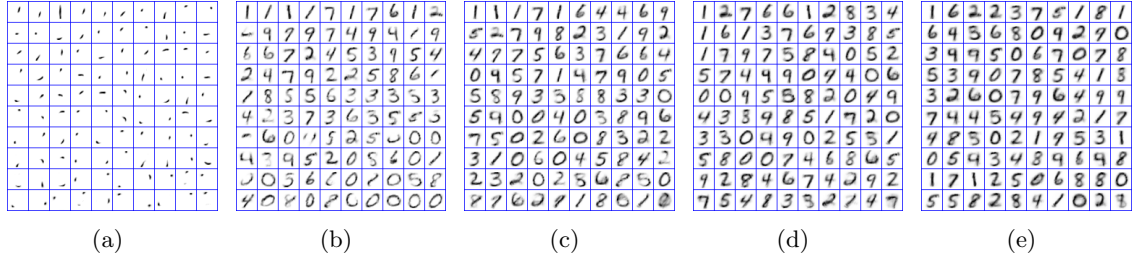
Figure 22: Visualization of the inferred $\{\mathbf{\Phi}^{(1)}, \cdots, \mathbf{\Phi}^{(T)}\}$ on the MNIST data set using the PRG-GBN with $K_{1max} = 100$ and $\eta^{(t)} = 0.05$ for all t. The latent factors of all layers are projected to the first layer: (a) $\mathbf{\Phi}^{(1)}$, (b) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}$, (c) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}$, (d) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}\mathbf{\Phi}^{(4)}$, and (e) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}\mathbf{\Phi}^{(4)}\mathbf{\Phi}^{(5)}$.
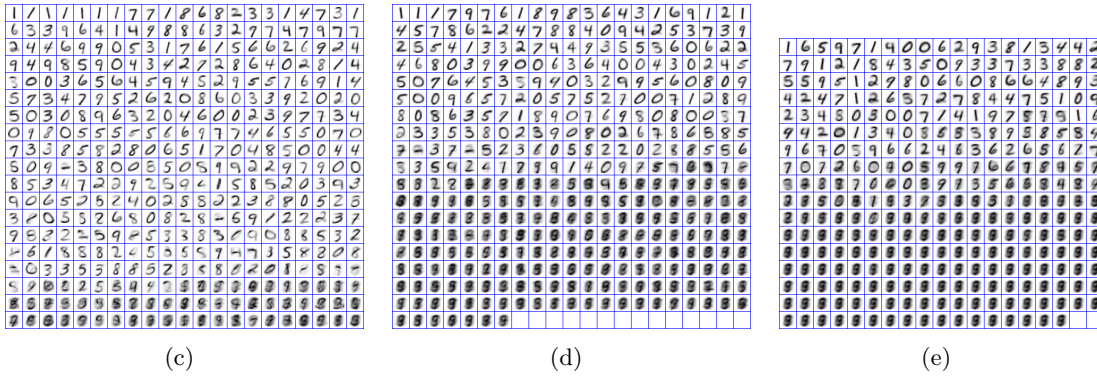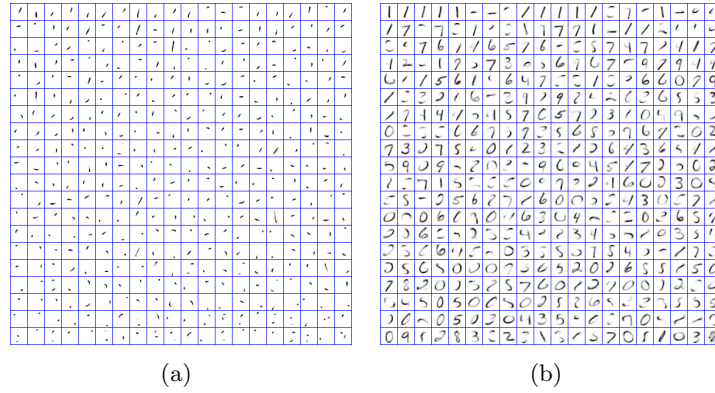
Figure 23: Visualization of the inferred $\{\mathbf{\Phi}^{(1)}, \cdots, \mathbf{\Phi}^{(T)}\}$ on the MNIST data set using the PRG-GBN with $K_{1max} = 400$ and $\eta^{(t)} = 0.05$ for all t. The latent factors of all layers are projected to the first layer: (a) $\mathbf{\Phi}^{(1)}$, (b) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}$, (c) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}$, (d) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}\mathbf{\Phi}^{(4)}$, and (e) $\mathbf{\Phi}^{(1)}\mathbf{\Phi}^{(2)}\mathbf{\Phi}^{(3)}\mathbf{\Phi}^{(4)}\mathbf{\Phi}^{(5)}$.

(a)

(b)

(c)

(d)
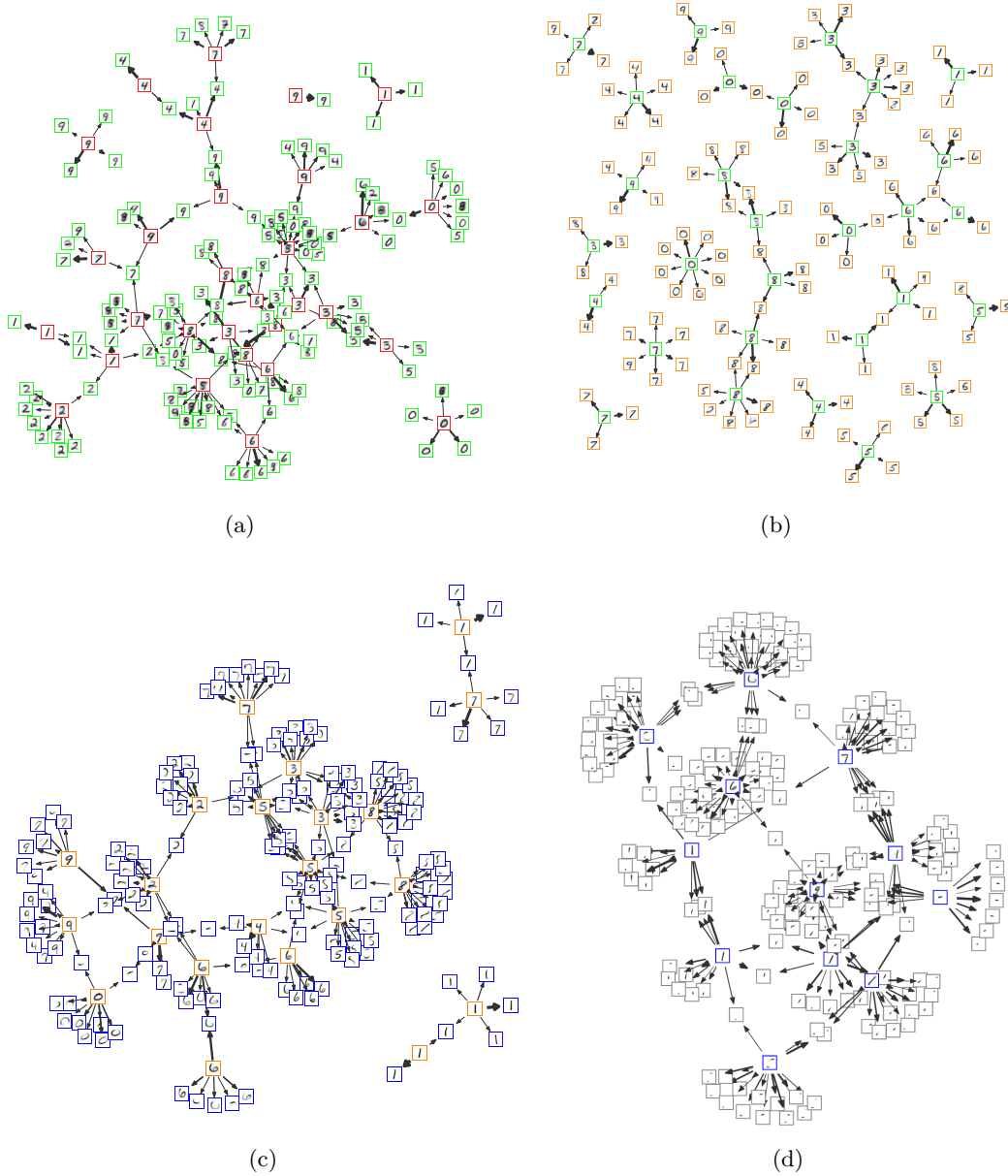
Figure 24: Visualization of the network structures inferred by the PRG-GBN on the MNIST data set with $K_{1max} = 400$. (a) Visualization of the factors $(\phi_1^{(5)}, \phi_{11}^{(5)}, \phi_{21}^{(5)}, \ldots, \phi_{111}^{(5)})$ of layer five and those of layer four that are strongly connected to them. (b) Visualization of the factors $(\phi_1^{(4)}, \phi_6^{(4)}, \phi_{11}^{(4)}, \ldots, \phi_{106}^{(4)})$ of layer four and those of layer three that are strongly connected to them. (c)the Visualization of the factors $(\phi_1^{(3)}, \phi_6^{(3)}, \phi_{11}^{(3)}, \ldots, \phi_{146}^{(3)})$ of layer three and those of layer two that are strongly connected to them. (d) Visualization of the factors $(\phi_1^{(2)}, \phi_6^{(2)}, \phi_{11}^{(2)}, \ldots, \phi_{146}^{(2)})$ of layer two and those of layer one that are strongly connected to them.

References

A. Acharya, J. Ghosh, and M. Zhou. Nonparametric Bayesian factor analysis for dynamic count matrices. In *AISTATS*, pages 1–9, 2015.

R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. In *NIPS*, 2010.

D. Aldous. Exchangeability and related topics. *École d'ete de probabilités de Saint-Flour XIII-1983*, pages 1–198, 1985.

F. J. Anscombe. Sampling theory of the negative binomial and logarithmic series distributions. *Biometrika*, 37(3-4):358–382, 1950.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.*, 2(6):1152–1174, 1974.

Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In Léon Bottou, Olivier Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007.

Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, 2007.

Y. Bengio, I. J. Goodfellow, and A. Courville. Deep Learning. Book in preparation for MIT Press, 2015.

D. Blackwell and J. MacQueen. Ferguson distributions via Pólya urn schemes. *Ann. Statist.*, 1(2):353–355, 1973.

D. Blei and J. Lafferty. Correlated topic models. *NIPS*, 2006.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3: 993–1022, 2003.

D. M. Blei and J. D. Mcauliffe. Supervised topic models. In *NIPS*, 2008.

D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of ACM*, 2010.

C. I. Bliss and R. A. Fisher. Fitting the negative binomial distribution to biological data. *Biometrics*, 9(2):176–200, 1953.

W. Buntine and A. Jakulin. Discrete component analysis. In *Subspace, Latent Structure and Feature Selection Techniques*. Springer-Verlag, 2006.

J. Canny. Gap: a factor model for discrete data. In *SIGIR*, 2004.

J. Chen, J. Zhu, Z. Wang, X. Zheng, and B. Zhang. Scalable inference for logistic-normal topic models. In *NIPS*, 2013.

S. C. Choi and R. Wette. Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, pages 683–690, 1969.

L. Devroye. Simulating Bessel random variables. *Statistics & probability letters*, 57(3): 249–257, 2002.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, pages 1871–1874, 2008.

R. A. Fisher, A. S. Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology*, 12(1):42–58, 1943.

B. J. Frey. Continuous sigmoidal belief networks trained using slice sampling. pages 452–458, 1997a.

B. J. Frey. Variational inference for continuous sigmoidal Bayesian networks. In *Sixth International Workshop on Artificial Intelligence and Statistics. Ft. Lauderdale FL*. Citeseer, 1997b.

B. J. Frey and G. E. Hinton. Variational learning in nonlinear Gaussian belief networks. *Neural Computation*, 11(1):193–213, 1999.

Z. Gan, C. Chen, R. Henao, D. Carlson, and L. Carin. Scalable deep Poisson factor analysis for topic modeling. In *ICML*, 2015a.

Z. Gan, R. Henao, D. E. Carlson, and L. Carin. Learning deep sigmoid belief networks with data augmentation. In *AISTATS*, 2015b.

M. Greenwood and G. U. Yule. An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. *J. R. Stat. Soc.*, 1920.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 2004.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

G. E. Hinton and Z. Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1177–1190, 1997.

G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

C. Hu, P. Rai, and L. Carin. Zero-truncated poisson tensor factorization for massive binary tensors. In *UAI*, 2015.

N. L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*, volume 165. Wiley New York, 1997.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

H. Larochelle and S. Lauly. A neural autoregressive topic model. In *NIPS*, 2012.

Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.

S. Linderman, M. Johnson, and R. P. Adams. Dependent multinomial models made easy: Stick-breaking with the Pólya-Gamma augmentation. In *NIPS*, pages 3438–3446, 2015.

V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pages 807–814, 2010.

R. M. Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.

D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 2009.

J. Paisley, C. Wang, and D. M. Blei. The discrete infinite logistic normal distribution. *Bayesian Analysis*, 2012.

J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

J. Pitman. *Combinatorial stochastic processes*. Lecture Notes in Mathematics. Springer-Verlag, 2006.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using polya-gamma latent variables. *arXiv preprint arXiv:1205.0310*, 2012.

R. Ranganath and D. Blei. Correlated random measures. *arXiv:1507.00720v1*, 2015.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *AISTATS*, 2014a.

R. Ranganath, L. Tang, L. Charlin, and D. M. Blei. Deep exponential families. *arXiv*, 2014b.

M Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014.

R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *AISTATS*, 2009.

R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba. Learning with hierarchical-deep models. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1958–1971, 2013.

L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4(1):61–76, 1996.

N. Srivastava, R. Salakhutdinov, and G. E. Hinton. Modeling documents with a deep Boltzmann machine. In *UAI*, 2013.

I. Sutskever and G. E. Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20(11):2629–2636, 2008.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 2006.

D.A. van Dyk and X. Meng. The art of data augmentation. *J. Comp. Graph. Stat.*, 2001.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *ICML*, 2009.

M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *NIPS*, pages 1481–1488, 2004.

S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP compound Dirichlet process and its application to focused topic modeling. In *ICML*, 2010.

E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *UAI*, 2005.

L Yuan and J. D. Kalbfleisch. On the Bessel distribution and related problems. *Annals of the Institute of Statistical Mathematics*, 52(3):438–447, 2000.

M. Zhou. Beta-negative binomial process and exchangeable random partitions for mixed-membership modeling. In *NIPS*, pages 3455–3463, 2014.

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

M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):307–320, 2015.

M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and Poisson factor analysis. In *AISTATS*, pages 1462–1471, 2012.

M. Zhou, Y. Cong, and B. Chen. The Poisson gamma belief network. In *NIPS*, 2015a.

M. Zhou, O. H. M. Padilla, and J. G. Scott. Priors for random count matrices derived from a family of negative binomial processes. *To appear in J. Amer. Statist. Assoc.*, 2015b.

J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models. *J. Mach. Learn. Res.*, 13(1):2237–2278, 2012.