

Gamma Belief Networks (Deep Latent Dirichlet Allocation)

Mingyuan Zhou

(Joint work with **Yulai Cong** and **Bo Chen** at Xidian University)

IROM Department, McCombs School of Business
The University of Texas at Austin

Duke-Tsinghua Machine Learning Summer School
Duke Kunshan University, Kunshan, China
August 1, 2016

Deep learning

- ▶ Significant recent interest in deep learning due to its excellent performance in large-scale real applications, such as image classification and speech recognition.
- ▶ State-of-the-art results in supervised learning when the labeled data are abundant.
- ▶ Significant potential in unsupervised learning with deep models
- ▶ Deep generative models for nonlinear distributed representations:
 - ▶ SBN, sigmoid belief network
 - ▶ DBN, deep belief network (a SBN whose last layer is replaced with a restricted Boltzmann machine that is undirected)
 - ▶ DBM, deep Boltzmann machine (a hierarchy of restricted Boltzmann machines)

Restrictions of previous deep generative models

- ▶ The hidden units are often restricted to be binary.
- ▶ Difficult to train a deep network in an unsupervised manner.
- ▶ A greedy layer-wise training strategy is often used due to the difficulty of jointly training all hidden layers.
- ▶ Lack of principled ways to determine the network structure, including the depth (number of layers) of the network and width (number of units) of each of its hidden layers.
- ▶ Commonly used deep learning models are not naturally designed for count data.

Our objectives

- ▶ Design a multilayer deep generative model that is well suited for extracting nonlinear distributed representations for high-dimensional sparse count, binary, and nonnegative real vectors.
- ▶ Construct the deep network using nonnegative real hidden units rather than using binary ones.
- ▶ Using nonparametric Bayesian priors to automatically infer the network structure from the data.

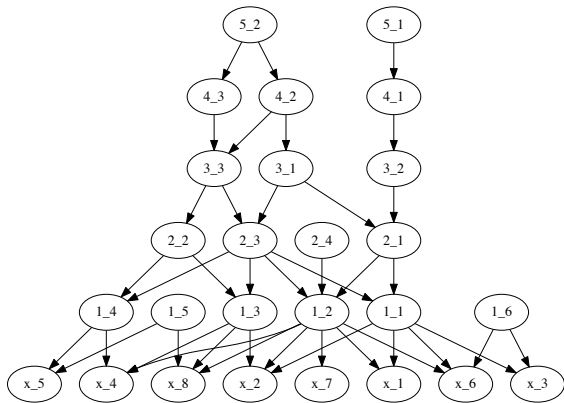


Figure: An example directed network of five hidden layers, with $K_0 = 8$ visible units, $[K_1, K_2, K_3, K_4, K_5] = [6, 4, 3, 3, 2]$, and sparse connections between the hidden units of adjacent layers.

$$\begin{aligned}
 P\left(\mathbf{x}_j^{(1)}, \{\theta_j^{(t)}\}_t \mid \{\Phi^{(t)}\}_t\right) &= P\left(\mathbf{x}_j^{(1)} \mid \Phi^{(1)}, \theta_j^{(1)}\right) \\
 &\times \left[\prod_{t=1}^{T-1} P\left(\theta_j^{(t)} \mid \Phi^{(t+1)}, \theta_j^{(t+1)}\right) \right] P\left(\theta_j^{(T)}\right).
 \end{aligned}$$

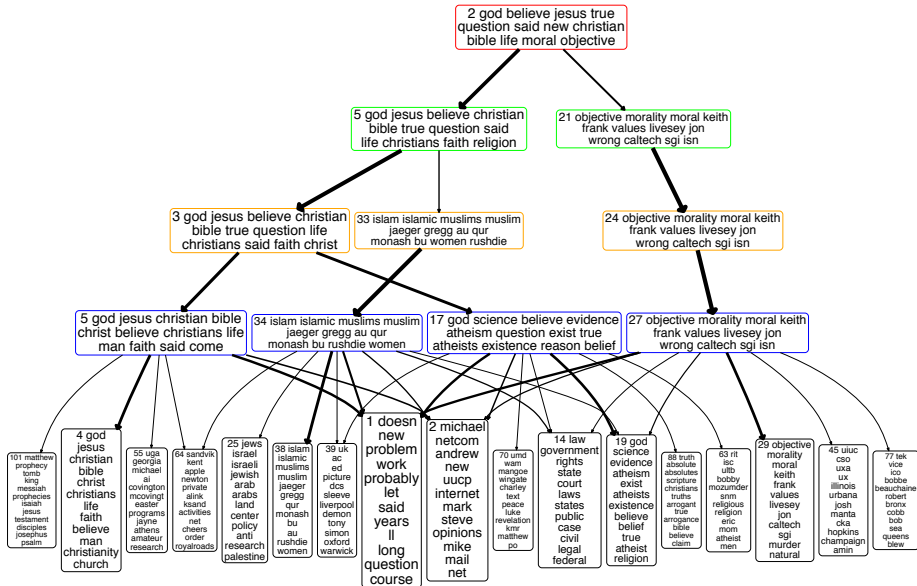


Figure: A tree on "religion."

The Poisson gamma belief network (PGBN)

- ▶ Assume the observations are multivariate count vectors $\mathbf{x}_j^{(1)} \in \mathbb{Z}^{K_0}$, where $\mathbb{Z} = \{0, 1, \dots\}$.
- ▶ We construct the PGBN [Zhou, Cong & Chen, 2015] to infer a multilayer deep representation for $\{\mathbf{x}_j^{(1)}\}_j$.
- ▶ With $\Phi^{(t)} \in \mathbb{R}_+^{K_{t-1} \times K_t}$, the generative model of the PGBN with T hidden layers, from the top to bottom, is expressed as

$$\theta_j^{(T)} \sim \text{Gam} \left(\mathbf{r}, 1/c_j^{(T+1)} \right),$$

...

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

...

$$\mathbf{x}_j^{(1)} \sim \text{Pois} \left(\Phi^{(1)} \theta_j^{(1)} \right), \quad \theta_j^{(1)} \sim \text{Gam} \left(\Phi^{(2)} \theta_j^{(2)}, p_j^{(2)} / (1 - p_j^{(2)}) \right).$$

- ▶ The PGBN factorizes the observed count vectors under the Poisson likelihood into the product of a factor loading matrix and the gamma distributed hidden units of layer one.
- ▶ The PGBN factorizes the hidden units of each hidden layer into the product a connection weight matrix and the hidden units of the next layer under the gamma likelihood.
- ▶ The PGBN with a single hidden layer (i.e., $T = 1$) reduces to Poisson factor analysis as

$$\mathbf{x}_j^{(1)} \sim \text{Pois} \left(\mathbf{\Phi}^{(1)} \boldsymbol{\theta}_j^{(1)} \right), \quad \boldsymbol{\theta}_j^{(1)} \sim \text{Gam} \left(\mathbf{r}, \mathbf{p}_j^{(2)} / (1 - \mathbf{p}_j^{(2)}) \right).$$

The gamma-negative binomial process [Zhou & Carin, 2015] can be used to support potentially $K_1 = \infty$ number of factors.

Deep Latent Dirichlet Allocation (DLDA)

- ▶ With $q_j^{(1)} = 1$, $q_j^{(t+1)} := \ln \left(1 + q_j^{(t)} / c_j^{(t+1)} \right)$, and $p_j^{(t)} := 1 - e^{-q_j^{(t)}}$, and with all $\theta_{jk}^{(t)}$ marginalized out, one may re-express the PGBN hierarchical model as deep latent Dirichlet allocation (DLDA) as

$$m_{kj}^{(T)(T+1)} \sim \text{SumLog}(x_{kj}^{(T+1)}, p_j^{(T+1)}), \quad x_{kj}^{(T+1)} \sim \text{Pois}(r_k q_j^{(T+1)}),$$

...

$$m_{vj}^{(t-1)(t)} \sim \text{SumLog}(x_{vj}^{(t)}, p_j^{(t)}), \quad x_{vj}^{(t)} = \sum_{k=1}^{K_t} x_{v kj}^{(t)}, \quad (x_{v kj}^{(t)})_v \sim \text{Mult} \left(m_{kj}^{(t)(t+1)}, \phi_k^{(t)} \right),$$

...

$$x_{vj}^{(1)} = \sum_{k=1}^{K_1} x_{v kj}^{(1)}, \quad (x_{v kj}^{(1)})_v \sim \text{Mult} \left(m_{kj}^{(1)(2)}, \phi_k^{(1)} \right).$$

Deep Latent Dirichlet Allocation (DLDA)

- ▶ With $\tilde{p} := q^{(T+1)} / (c_0 + q^{(T+1)})$ and the gamma process weights r_k marginalized out, one may re-express the PGBN hierarchical model as

$$\begin{aligned}
 X^{(T+1)} &= \sum_{k=1}^{K_T} x_{k \cdot}^{(T+1)} \delta_{\phi_k^{(T)}}, \quad x_{k \cdot}^{(T+1)} \sim \text{Log}(\tilde{p}), \quad K_T \sim \text{Pois}[-\gamma_0 \ln(1 - \tilde{p})], \\
 m_{vj}^{(T)(T+1)} &\sim \text{SumLog}(x_{vj}^{(T+1)}, p_j^{(T+1)}), \quad (x_{vj}^{(T+1)})_{1,J} \sim \text{Mult} \left[x_{v \cdot}^{(T+1)}, (q_j^{(T+1)})_{1,J} / q^{(T+1)} \right], \\
 &\dots \\
 m_{vj}^{(t-1)(t)} &\sim \text{SumLog}(x_{vj}^{(t)}, p_j^{(t)}), \quad x_{vj}^{(t)} = \sum_{k=1}^{K_t} x_{vkj}^{(t)}, \quad (x_{vkj}^{(t)})_v \sim \text{Mult} \left(m_{kj}^{(t)(t+1)}, \phi_k^{(t)} \right), \\
 &\dots \\
 x_{vj}^{(1)} &= \sum_{k=1}^{K_1} x_{vkj}^{(1)}, \quad (x_{vkj}^{(1)})_v \sim \text{Mult} \left(m_{kj}^{(1)(2)}, \phi_k^{(1)} \right).
 \end{aligned}$$

- ▶ The count matrix $\{m_{vj}^{(T)(T+1)}\}_{j=1,J; v=1,K_T}$ is drawn from a gamma-negative binomial process [Zhou, Padilla & Scott, 2016]

Model likelihood for PGBN

- ▶ The joint distribution of the observed counts and gamma hidden units given the network in the PGBN:

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

$$P\left(\theta_{vj}^{(t)} \mid \phi_{v:}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}, c_{j+1}^{(t+1)}\right) = \frac{\left(c_{j+1}^{(t+1)}\right)^{\phi_{v:}^{(t+1)} \boldsymbol{\theta}_j^{(t+1)}}}{\Gamma\left(\phi_{v:}^{(t+1)} \boldsymbol{\theta}_j^{(t+1)}\right)} \left(\theta_{vj}^{(t)}\right)^{\phi_{v:}^{(t+1)} \boldsymbol{\theta}_j^{(t+1)} - 1} e^{-c_{j+1}^{(t+1)} \theta_{vj}^{(t)}}$$

- ▶ The joint distribution of the binary visible and hidden units given the network for the sigmoid belief network (SBN):

$$P\left(\theta_{vj}^{(t)} = 1 \mid \phi_{v:}^{(t+1)}, \boldsymbol{\theta}_j^{(t+1)}, b_v^{(t+1)}\right) = \sigma\left(b_v^{(t+1)} + \phi_{v:}^{(t+1)} \boldsymbol{\theta}_j^{(t+1)}\right).$$

where $\sigma(x) = 1/(1 + e^{-x})$ denotes the sigmoid function.

Model likelihood for DLDA

- ▶ The likelihood of $\Phi^{(t)}$ in deep latent Dirichlet allocation (DLDA) can be expressed as

$$P\left(\{\mathbf{x}_j^{(t)}\}_t \mid \{\Phi^{(t)}\}_t, \mathbf{r}\right) \propto \prod_{t=1}^T \prod_{k=1}^{K_t} \text{Multinomial} \left[\left(x_{vkj}^{(t)} \right)_v ; m_{kj}^{(t)(t+1)}, \phi_k^{(t)} \right] \\ \times \prod_{k=1}^{K_T} \text{Pois}(x_{kj}^{(T+1)}; r_k q_j^{(T+1)}) .$$

- ▶ The expected Fisher information matrix for the global parameters $\phi_k^{(t)}$ and \mathbf{r} is block diagonal.
- ▶ Stochastic gradient MCMC for DLDA is simple under this likelihood.
- ▶ The likelihood becomes the same as that of latent Dirichlet allocation (LDA) [Blei, Ng & Jordan, 2003] if $T=1$.

Interpreting the structure of the PGBN

- ▶ Using the law of total expectation, we have

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

$$\mathbb{E}[\boldsymbol{\theta}_j^{(t)} \mid \{\boldsymbol{\Phi}^{(\ell)}, c_j^{(\ell)}\}_{t+1,T}, \mathbf{r}] = \left[\prod_{\ell=t+1}^T \boldsymbol{\Phi}^{(\ell)} \right] \frac{\mathbf{r}}{\prod_{\ell=t+1}^{T+1} c_j^{(\ell)}}.$$

- ▶ Consider $\prod_{\ell=1}^t \boldsymbol{\Phi}^{(\ell)}$ as the K_t topics/factors/nodes of layer $t \in \{1, \dots, T\}$ and use $\mathbf{r}^{(t)} := \left[\prod_{\ell=t+1}^T \boldsymbol{\Phi}^{(\ell)} \right] \mathbf{r}$ to rank them.
- ▶ Consider $\phi_{k'k}^{(t)} = \boldsymbol{\Phi}^{(t)}(k', k)$ as the weight that connects node k of layer t and node k' of layer $t - 1$.
- ▶ Our intuition is that examining the topics/factors from the top to bottom layers will gradually reveal less general and more specific aspects of the data.

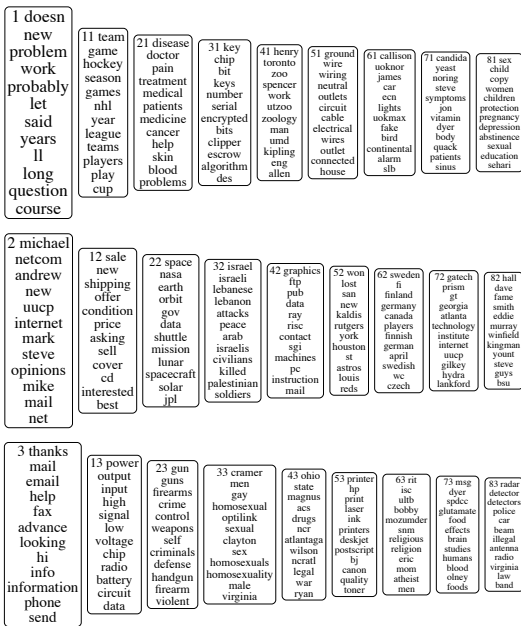


Figure: Example topics of layer one of the PGBN learned on the 20newsgroups corpus.



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



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

Visualize the inferred deep network

- ▶ Visualize the whole network is challenging if the network is large.
- ▶ But we can easily visualize a tree or a subnetwork that consists of multiple trees.
- ▶ Construct a tree starting from a top-layer node: grow the tree downward by linking each leaf node of the tree to all the hidden units at the layer below that are connected to the leaf node with nonnegligible weights.

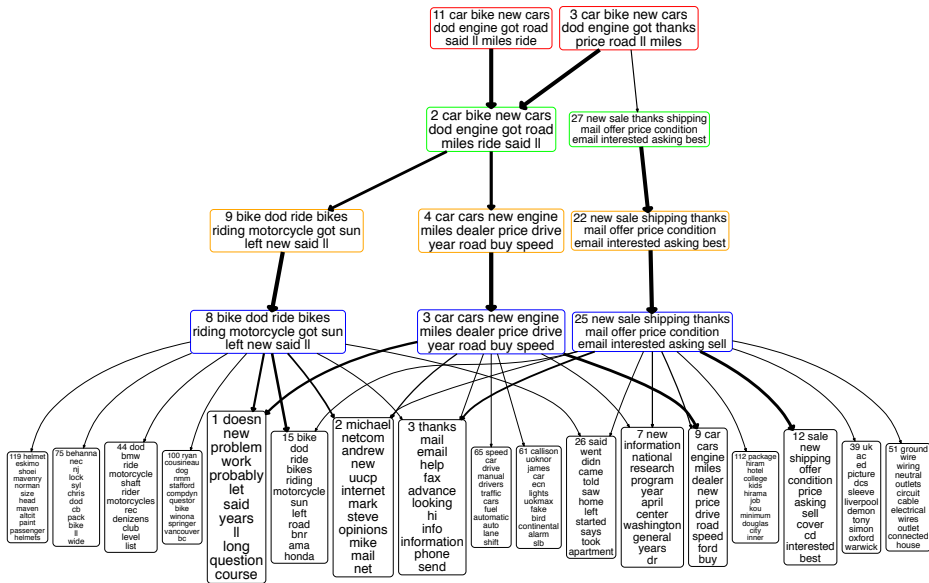


Figure: A subnetwork on "car & bike."

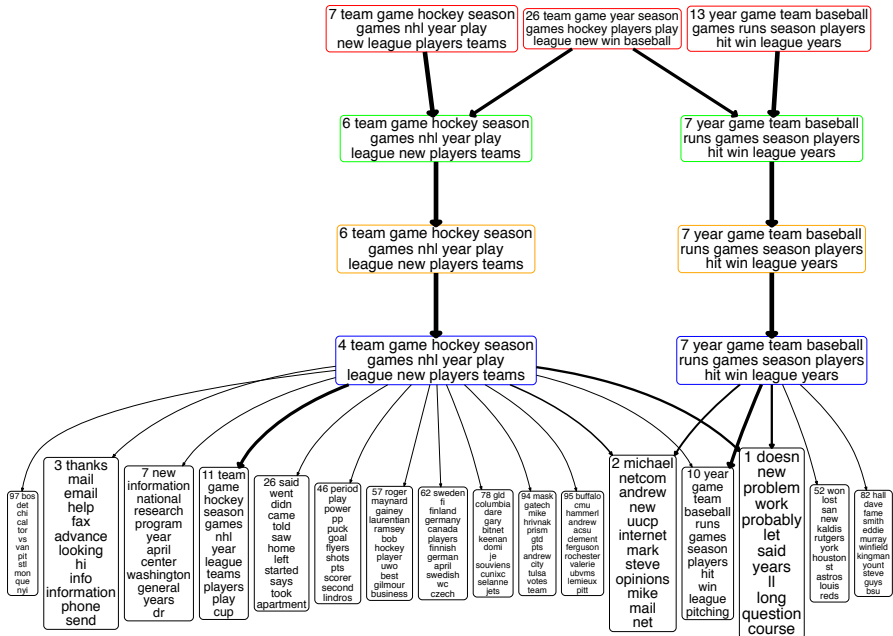


Figure: A subnetwork on "sports."

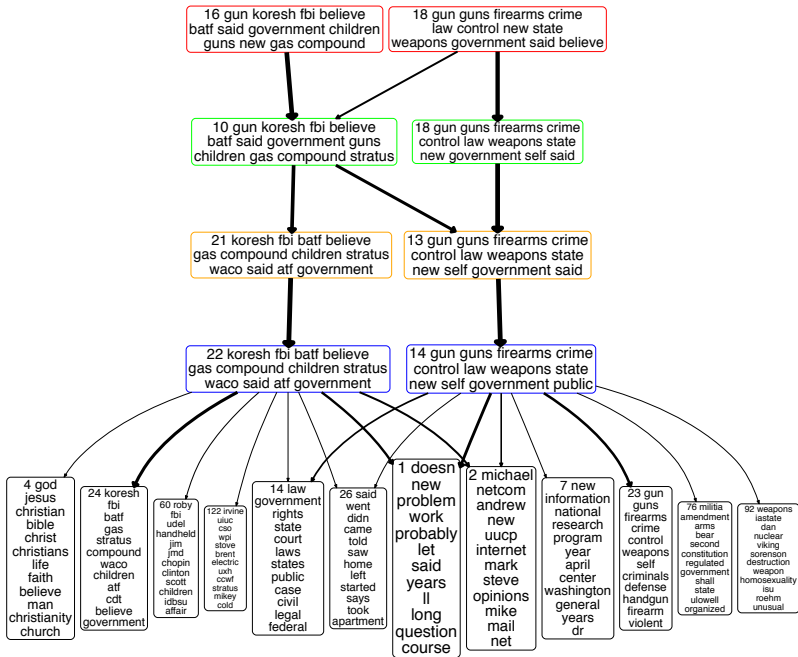


Figure: A subnetwork on "gun."

Upward-downward Gibbs sampling

Lemma (Augment-and-conquer the gamma belief network)

With $p_j^{(1)} := 1 - e^{-1}$ and

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

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

$$\mathbf{x}_j^{(t)} \sim \text{Pois} \left[-\Phi^{(t)} \theta_j^{(t)} \ln \left(1 - p_j^{(t)} \right) \right].$$

Upward propagate latent counts

Corollary (Propagate the latent counts upward)

With $m_{kj}^{(t)(t+1)} := x_{\cdot jk}^{(t)} := \sum_{v=1}^{K_{t-1}} x_{vjk}^{(t)}$ representing the number of times that factor $k \in \{1, \dots, K_t\}$ of layer t appears in observation j , we can propagate the latent counts $x_{vj}^{(t)}$ of layer t upward to layer $t+1$ as

$$\left\{ \left(x_{vj1}^{(t)}, \dots, x_{vjK_t}^{(t)} \right) \mid x_{vj}^{(t)}, \phi_{v\cdot}^{(t)}, \theta_j^{(t)} \right\} \\ \sim \text{Mult} \left(x_{vj}^{(t)}, \frac{\phi_{v1}^{(t)} \theta_{1j}^{(t)}}{\sum_{k=1}^{K_t} \phi_{vk}^{(t)} \theta_{kj}^{(t)}}, \dots, \frac{\phi_{vK_t}^{(t)} \theta_{K_tj}^{(t)}}{\sum_{k=1}^{K_t} \phi_{vk}^{(t)} \theta_{kj}^{(t)}} \right),$$

$$(\phi_k^{(t)} \mid -) \sim \text{Dir} \left(\eta_1^{(t)} + x_{1\cdot k}^{(t)}, \dots, \eta_{K_{t-1}}^{(t)} + x_{K_{t-1}\cdot k}^{(t)} \right),$$

$$\left(x_{kj}^{(t+1)} \mid m_{kj}^{(t)(t+1)}, \phi_{k\cdot}^{(t+1)}, \theta_j^{(t+1)} \right) \sim \text{CRT} \left(m_{kj}^{(t)(t+1)}, \phi_{k\cdot}^{(t+1)} \theta_j^{(t+1)} \right).$$

Downward sample the hidden units

Using the latent counts propagated upward and the gamma-Poisson conjugacy, we downward sample the hidden units as

$$(r_k | -) \sim \text{Gam} \left(\gamma_0 / K_T + x_k^{(T+1)}, \left[c_0 - \sum_j \ln(1 - p_j^{(T+1)}) \right]^{-1} \right),$$

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

$$(\theta_j^{(T-1)} | -) \sim \text{Gam} \left(\Phi^{(T)} \theta_j^{(T)} + \mathbf{m}_j^{(T-1)(T)}, \left[c_j^{(T)} - \ln(1 - p_j^{(T-1)}) \right]^{-1} \right),$$

⋮

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

⋮

$$(\theta_j^{(1)} | -) \sim \text{Gam} \left(\Phi^{(2)} \theta_j^{(2)} + \mathbf{m}_j^{(1)(2)}, \left[c_j^{(2)} - \ln(1 - p_j^{(1)}) \right]^{-1} \right).$$

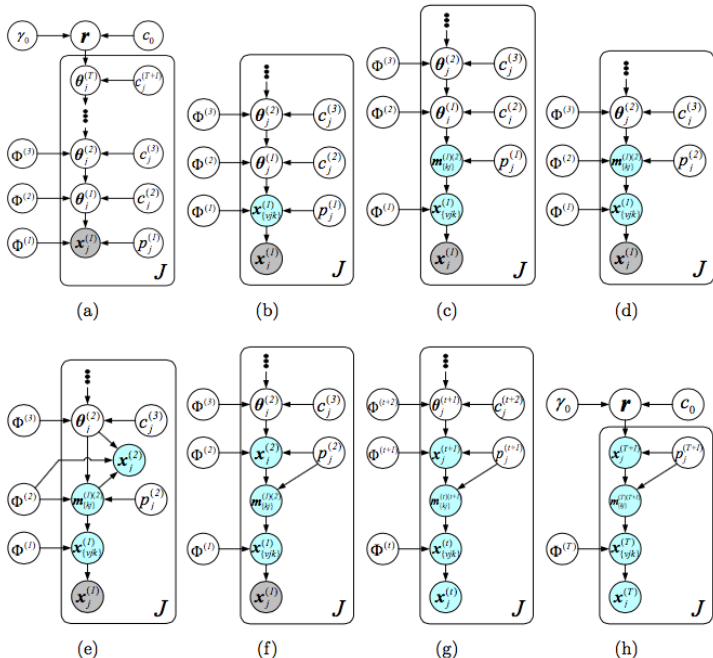


Figure: Graphical representation of the model and inference scheme.

Modeling overdispersion with distributed representation

Assuming $\Phi^{(t)} = \mathbf{I}$ for all $t \in 3, \dots, T$, we have

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

- ▶ In comparison to PFA with $m_{kj}^{(1)(2)} \sim \text{NB}(r_k, p_j^{(2)})$, the PGBN increases $\text{VMR}[m_{kj}^{(1)(2)} | r_k]$ by a factor of

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

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 variance into more layers, the multilayer structure could increase its capability to model data variability.

- ▶ For the GBN with $T = 1$, given the shared weight vector \mathbf{r} , we have

$$\mathbb{E}[\mathbf{x}_j^{(1)} \mid \Phi^{(1)}, \mathbf{r}] = \Phi^{(1)} \mathbf{r} / c_j^{(2)};$$

- ▶ For the GBN with $T \geq 2$, given the weight vector $\theta_j^{(2)}$, we have

$$\mathbb{E}[\mathbf{x}_j^{(1)} \mid \Phi^{(1)}, \Phi^{(2)}, \theta_j^{(2)}] = \Phi^{(1)} \Phi^{(2)} \theta_j^{(2)} / c_j^{(2)}.$$

- ▶ Thus in the prior, the co-occurrence patterns of the columns of $\Phi^{(1)}$ are captured in a single vector \mathbf{r} when $T = 1$, and are captured in the columns of $\Phi^{(2)}$ 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 $\prod_{\ell=1}^t \Phi^{(\ell)}$ will be captured in the columns of the $K_t \times K_{t+1}$ matrix $\Phi^{(t+1)}$.

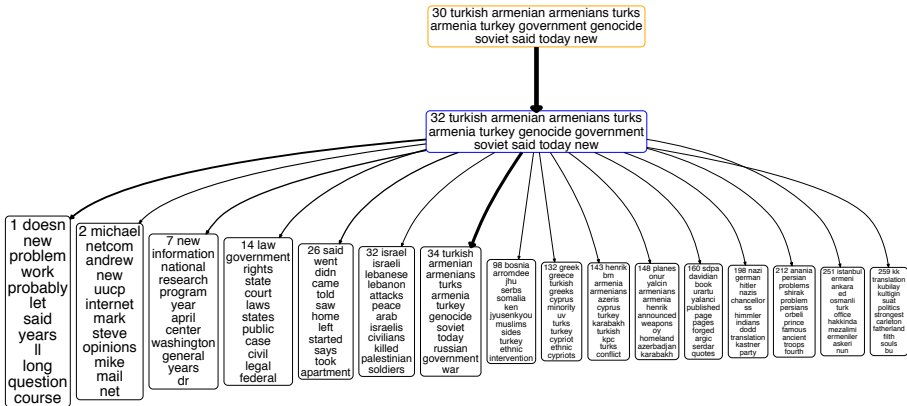


Figure: The tree rooted at node 30 of layer three on "Turkey & Armenia."

Learning the network structure

- ▶ Using greedy layer-wise training together with the gamma-negative binomial process on the top hidden layer.

Modeling binary and nonnegative real observations

- ▶ We link binary observations to the latent counts at layer one as

$$b_{vj}^{(1)} = \mathbf{1}(x_{vj}^{(1)} \geq 1).$$

- ▶ For inference, we sample the latent counts at layer one from the truncated Poisson distribution as

$$(x_{vj}^{(1)} \mid -) \sim b_{vj}^{(1)} \cdot \text{Pois}_+ \left(\sum_{k=1}^{K_1} \phi_{vk}^{(1)} \theta_{kj}^{(1)} \right).$$

- ▶ We link nonnegative real observations to the latent counts at layer one using

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

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

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

if $y_{vj}^{(1)} > 0$.

Multivariate count data

- ▶ Train on the 20 newsgroups dataset, each document of which is summarized as a word count vector over the vocabulary.
 - ▶ We use all 11,269 training documents to infer a five layer network.
 - ▶ After removing stopwords and terms that appear less than five times, we obtain a vocabulary with $K_0 = 33,420$.
 - ▶ With $\eta^{(t)} = 0.05$ for all t , the inferred network widths by the PGBN are $[K_1, K_2, K_3, K_4, K_5] =$
 - ▶ $[50, 50, 50, 50, 50]$ for $K_{1\max} = 50$
 - ▶ $[100, 99, 99, 94, 87]$ for $K_{1\max} = 100$
 - ▶ $[200, 161, 130, 94, 63]$ for $K_{1\max} = 200$
 - ▶ $[396, 109, 99, 82, 68]$ for $K_{1\max} = 400$
 - ▶ $[528, 129, 109, 98, 91]$ for $K_{1\max} = 600$
 - ▶ $[608, 100, 99, 96, 89]$ for $K_{1\max} = 800$
 - ▶ Test on the 7,505 documents in the testing set.

Feature learning for multi-class classification

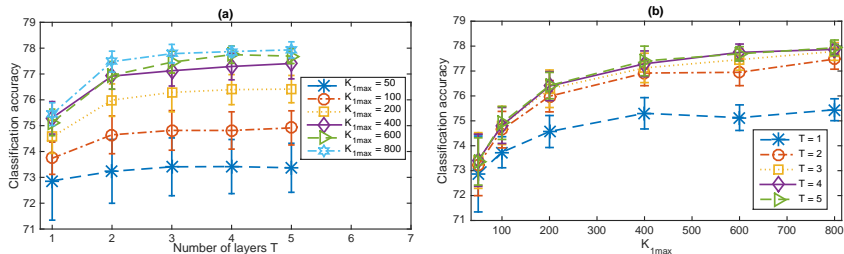


Figure: Classification accuracy (%) of the PGBNs with Algorithm 1 for 20newsgroups multi-class classification (a) as a function of the depth T with various $K_{1\max}$ and (b) as a function of $K_{1\max}$ with various depths, with $\eta^{(t)} = 0.05$ for all layers.

Prediction of heldout words

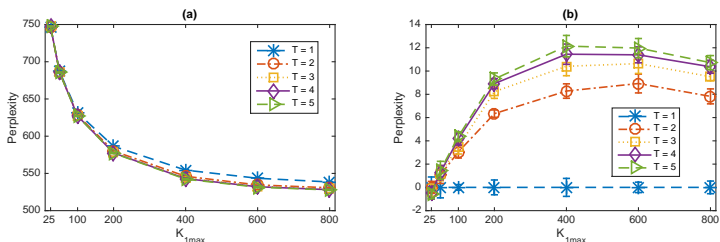
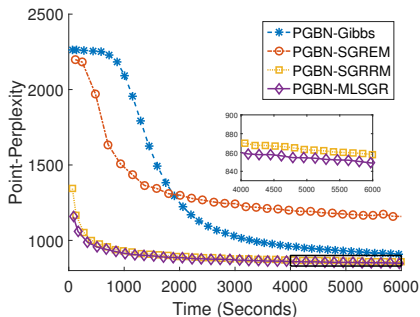
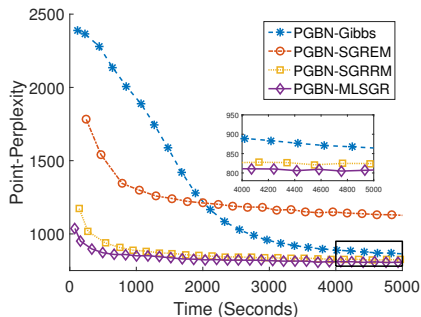


Figure: (a) per-heldout-word perplexity (the lower the better) for the NIPS12 corpus (using the 2000 most frequent terms) as a function of the upper bound of the first layer width $K_{1\max}$ and network depth T , with 30% of the word tokens in each document used for training and $\eta^{(t)} = 0.05$ 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.

Perplexity using stochastic gradient-MCMC



(a)



(b)

Figure: Point-Perplexity results versus time. (a) RCV1. (b) Wiki.

- ▶ Gibbs: Gibbs sampling, batch learning
- ▶ SGREM: stochastic gradient Riemannian (expanded mean) [Patterson & Ten, 2013; Chen, Fox & Guestrin, 2014]
- ▶ SGRRM: stochastic gradient Riemannian (reduced mean) [Cong, Bo & Zhou, 2016]
- ▶ MLSGR: multilayer stochastic gradient Riemannian (reduced mean) [Cong, Bo & Zhou, 2016]

Simulate documents

- ▶ Draw $\boldsymbol{\theta}^{(T)}, \boldsymbol{\theta}^{(T-1)}, \dots, \boldsymbol{\theta}^1$ using

$$\boldsymbol{\theta}_{j'}^{(T)} \sim \text{Gam} \left(\mathbf{r}, \left[c_{j'}^{(T+1)} \right]^{-1} \right),$$

$$\boldsymbol{\theta}_{j'}^{(T-1)} \sim \text{Gam} \left(\boldsymbol{\Phi}^{(T)} \boldsymbol{\theta}_{j'}^{(T)}, \left[c_{j'}^{(T)} \right]^{-1} \right),$$

⋮

$$\boldsymbol{\theta}_{j'}^{(t)} \sim \text{Gam} \left(\boldsymbol{\Phi}^{(t+1)} \boldsymbol{\theta}_{j'}^{(t+1)}, \left[c_{j'}^{(t+1)} \right]^{-1} \right),$$

⋮

$$\boldsymbol{\theta}_{j'}^{(1)} \sim \text{Gam} \left(\boldsymbol{\Phi}^{(2)} \boldsymbol{\theta}_{j'}^{(2)}, \left[c_{j'}^{(2)} \right]^{-1} \right).$$

- ▶ Calculate $E[\mathbf{x}_{j'}^{(1)}] = \boldsymbol{\Phi}^{(1)} \boldsymbol{\theta}_{j'}^{(1)}$
- ▶ Display the top 100 words

Simulate documents

- ▶ mac apple bit mhz ram simms mb like memory just don cpu people chip chips think color board ibm speed does know se video time machines motherboard hardware lc cache meg ns simm need upgrade built vram good quadra want centris price dx run way processor card clock slots make fpu internal did macs cards ve pin power really machine say faster said software intel macintosh right week writes slot going sx performance things edu years nubus possible thing monitor work point expansion rom iisi ll add dram better little slow let sure pc ii didn ethernet lciii case kind

Simulate documents

- ▶ image jpeg gif file color files images format bit display convert quality formats colors programs program tiff picture viewer graphics bmp bits xv screen pixel read compression conversion zip shareware scale view jpg original save quicktime jfif free version best pcx viewing bitmap gifs simtel viewers don mac usenet resolution animation menu scanner pixels sites gray quantization displays better try msdos tga want current black faq converting white setting mirror xloadimage section ppm fractal amiga write algorithm mpeg pict targa arithmetic export scodal archive converted grasp lossless let space human grey directory pictures rgb demo scanned old choice grayscale compress

Simulate documents

- ▶ medical health disease doctor pain patients treatment medicine cancer edu hiv blood use years patient writes cause skin don like just aids symptoms number article help diseases drug com effects information doctors infection physician normal chronic think taking care volume condition drugs page says cure people tobacco hicnet know newsletter effective therapy problem common time women prevent surgery children center immune research called april control effect weeks low syndrome hospital physicians states clinical diagnosed day med age good make caused severe reported public safety child said cdc usually diet national studies tissue months way cases causing migraine smokeless infections does

Simulate documents

- ▶ men homosexual sex gay sexual homosexuality male don people partners promiscuous number just bi like study homosexuals percent cramer heterosexual think did dramatically numbers straight church reform population know report pythagorean life man good accept time said considered kinsey posted general optilink irrational social gays behavior way children make published johnson survey table new activity showing million statistics american sexuality shows want right women ve article ago exclusively eating virginia masters repent really say purpose member clayton apparent kolodny writes going press society evil function engaged relationships ryan evolutionary different does compared person join figure community edu chose interesting things

Simulate documents

- ▶ nissan electronics wagon altima delcoelect kocrsv station gm subaru sumax delco spiros hughes wax pathfinder legacy kokomo wagons smorris scott toyota seattleu don just like strong silver software luxury derek proof stanza seattle cisco morris cymbal triantafyllopoulos sportscar think people know near fool ugly proud claims flat statistics lincoln sedans bullet karl lee perth puzzled miata sentra maxima acura infiniti corolla mgb untruth verbatim good time consider way based make stand guys writes noticed want ve heavy suggestion eat steven horrible uunet studies armor fisher lust designs study definately lexus remove conversion embodied aesthetic elvis attached honey stole designing wd

Modeling high-dimensional sparse binary vectors

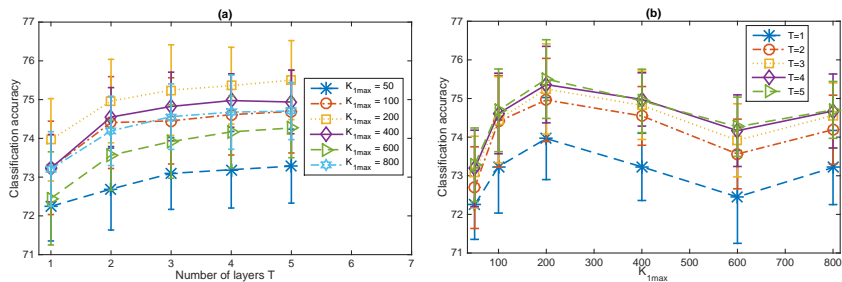


Figure: Results of the BerPo-GBNs on the binarized 20newsgroups term-document count matrix. The widths of the hidden layers are automatically inferred. In a random trial with Algorithm 2, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1\max} = 50, 100, 200, 400, 600,$ and 800 are $[50, 50, 50, 50, 50]$, $[100, 97, 95, 90, 82]$, $[178, 145, 122, 97, 72]$, $[184, 139, 119, 101, 75]$, $[172, 165, 158, 138, 110]$, and $[156, 151, 147, 134, 117]$, respectively.

Modeling high-dimensional sparse nonnegative real vectors

- ▶ Train on the 60,000 MNIST digits in the training set, each digit of which is represented as a 784 dimensional nonnegative real vector.
 - ▶ We use all 60,000 to infer a five layer network.
 - ▶ With $\eta^{(t)} = 0.05$ for all t , the inferred network widths by the gamma-PGBN are $[K_1, K_2, K_3, K_4, K_5] =$
 - ▶ [50, 50, 50, 50, 50] for $K_{1\max} = 50$
 - ▶ [100, 100, 100, 100, 100] for $K_{1\max} = 100$
 - ▶ [200, 200, 200, 200, 200] for $K_{1\max} = 200$
 - ▶ [400, 400, 399, 385, 321] for $K_{1\max} = 400$
 - ▶ Test on the 10,000 MNIST digits in the test set.

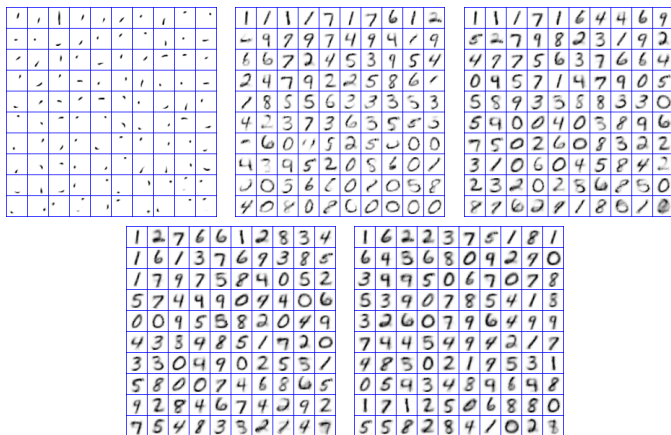
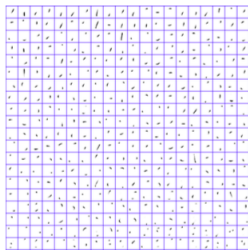


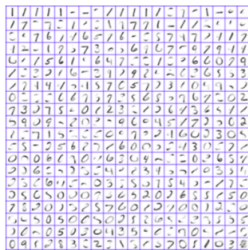
Figure: Visualization of the inferred $\{\phi^{(1)}, \dots, \phi^{(T)}\}$ with $K_{1max} = 100$. All ϕ 's are projected to the first layer.

First row from left to right: $\phi^{(1)}$, $\phi^{(1)}\phi^{(2)}$, $\phi^{(1)}\phi^{(2)}\phi^{(3)}$

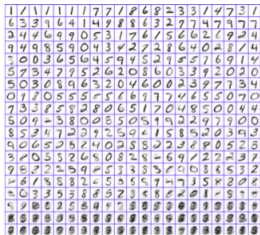
Second row from left to right: $\phi^{(1)}\phi^{(2)}\phi^{(3)}\phi^{(4)}$, $\phi^{(1)}\phi^{(2)}\phi^{(3)}\phi^{(4)}\phi^{(5)}$.



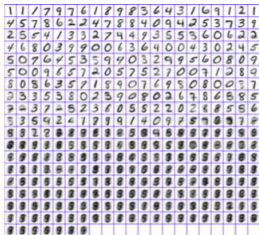
(a)



(b)



(c)



(d)



(e)

Figure: Visualization of the inferred $\{\Phi^{(1)}, \dots, \Phi^{(T)}\}$ with $K_{1max} = 400$. All Φ 's are projected to the first layer. From (a) to (e): $\Phi^{(1)}$, $\Phi^{(1)}\Phi^{(2)}$, $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}$, $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}$, $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}\Phi^{(5)}$.

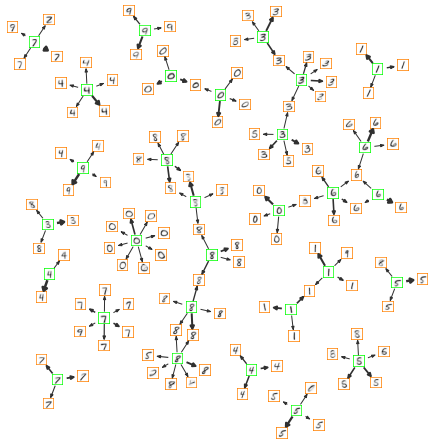
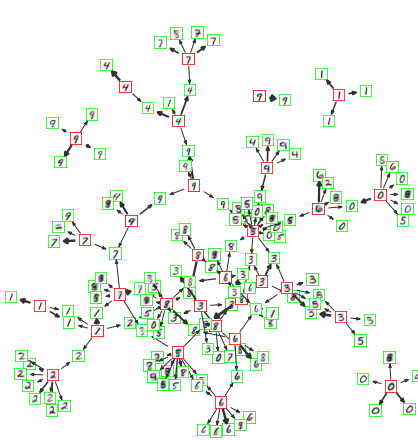


Figure: Left: layers 5 to 4;

Right: layers 4 to 3

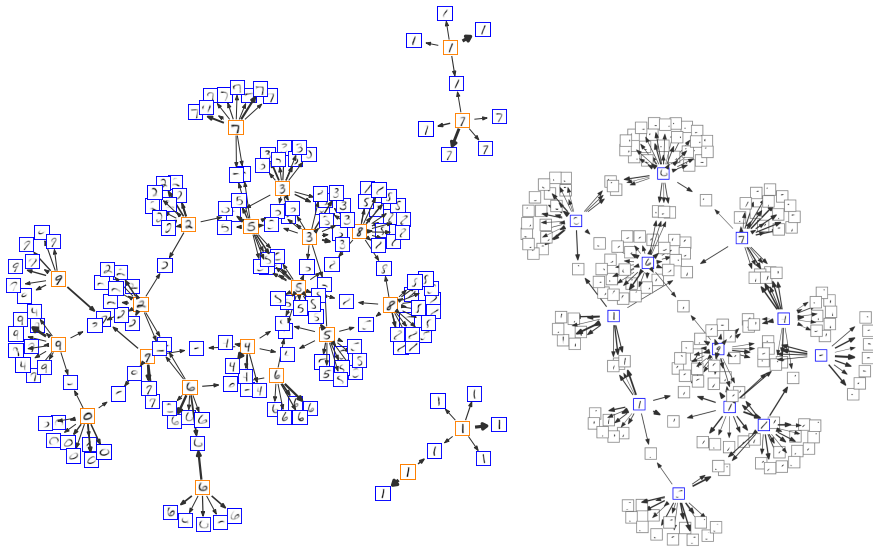


Figure: Left: layers 3 to 2;

Right: layers 2 to 1

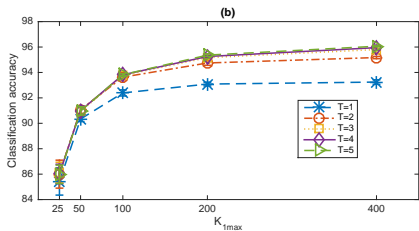
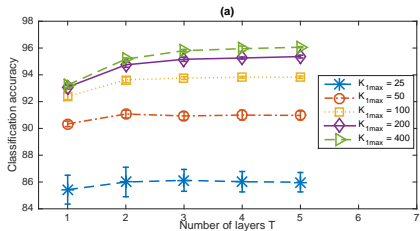









Figure: Left: Classification accuracy (%) of the PRG-GBNs with various K_{1max} as a function of T_{1max} . Right: Classification accuracy (%) of the PRG-GBNs with various depths as a function of K_{1max} . ($T_{max} \in \{1, \dots, 5\}$).

Conclusions

- ▶ The Poisson gamma belief network is proposed to extract a multilayer representation for high-dimensional count vectors.
- ▶ Upward-downward Gibbs sampling to jointly train all layers.
- ▶ A layer-wise training strategy to infer the network structure.
- ▶ Deep latent Dirichlet allocation and multilayer stochastic gradient Riemannian MCMC.
- ▶ Extension to modeling binary and nonnegative real data.
- ▶ Understanding the data by examining the features of different layers and their relationships using the structure of the network.
- ▶ For big data problems, in practice one may rarely has 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 widths and depth of the network.

Main References

-  D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.
-  T. Chen, E. B. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. *arXiv preprint arXiv:1402.4102*, 2014.
-  S. Patterson and Y. W. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *NIPS*, pages 3102–3110, 2013.
-  M. Zhou, Y. Cong, and B. Chen. The Poisson gamma belief network. In *NIPS*, 2015.
-  M. Zhou, Y. Cong, and B. Chen. Gamma belief networks. *arXiv:1512.03081*, Dec. 2015.
-  M. Zhou, O. Padilla, and J. G. Scott. Priors for random count matrices derived from a family of negative binomial processes. *To appear in JASA*, 2016.
-  Y. Cong, B. Chen, and M. Zhou. Learning deep latent Dirichlet allocation via multilayer stochastic gradient Riemannian MCMC. *Preprint*, June 2016.