Deep Topic Models for Multi-label Learning

Rajat Panda[*†] Goldman Sachs
Ankit Pensia[*†] UW-Madison
Nikhil Mehta Duke University
Mingyuan Zhou UT-Austin
Piyush Rai IIT Kanpur

Abstract

We present a probabilistic framework for multi-label learning based on a deep generative model for the binary label vector associated with each observation. Our generative model learns deep multi-layer latent embeddings of the binary label vector, which are conditioned on the input features of the observation. The model also has an interesting interpretation in terms of a deep topic model, with each label vector representing a bag-of-words document, with the input features being its meta-data. In addition to capturing the structural properties of the label space (e.g., a near-low-rank label matrix), the model also offers a clean, geometric interpretation. In particular, the nonlinear classification boundaries learned by the model can be seen as the union of multiple convex polytopes. Our model admits a simple and scalable inference via efficient Gibbs sampling or EM algorithm. We compare our model with state-of-the-art baselines for multi-label learning on benchmark data sets, and also report some interesting qualitative results.

1 INTRODUCTION

Multi-label learning (Gibaja and Ventura, 2014, 2015) refers to the problem of annotating an object with a subset of *relevant* labels from a potentially massive set of labels. Multi-label learning problems in modern-day applications such as recommender systems involve very high-dimensional feature and label spaces, and also dubbed "extreme" multi-label learning problems (Babbar and Schölkopf, 2017; Jain et al., 2016; Prabhu and Varma, 2014). Since, in such problem settings, the labels tend to be related with each other, the naïve approach of learning a classifier for each label independently is usually sub-optimal and suitable structural assumptions need to be imposed, e.g., learning an embedding of the label matrix with a low-rank (Rai et al., 2015; Yu et al., 2014) or near-low-rank (Bhatia et al., 2015; Xu et al., 2016) assumption. Other notable approaches for multi-label learning include tree-based methods (Jain et al., 2016; Prabhu and Varma, 2014) and carefully regularized one-vs-all approaches (Babbar and Schölkopf, 2017). We provide a more detailed discussion of prior work on multi-label learning in the Related Work section.

While some of the recently proposed approaches to multi-label learning have led to impressive results on benchmark data sets, these methods still have some key limitations. In particular, most of the existing methods are based on linear models (e.g., linear embeddings of label vectors, or linear classifiers for label prediction models), which may not be sufficiently expressive to learn complex classification boundaries if the underlying problem requires nonlinear classifiers. Moreover, most of the existing methods are non-probabilistic in nature, and lack a proper generative model for the data, which may be useful in exploiting the rich structure in the high-dimensional features and label vectors. A probabilistic, generative framework is also appealing since it opens door to extensions such as semi-supervised learning (Kingma et al., 2014), or active learning (Kapoor et al., 2012; Vasisht et al., 2014).

With these desiderata as our motivation, we present a probabilistic, Bayesian framework based on a deep generative model for the binary label vectors. Our generative model is built using a Bernoulli-Poisson likelihood model (Rai et al., 2015; Zhou, 2015) for each entry of the label vector and constructs a deep hierarchy of gamma-distributed non-negative, interpretable, low-dimensional embeddings for the label vector. This construction is appealing due to several reasons: (1) The deep generative model enables learning *nonlinear*

[*]Equal contribution.

[†]Majority of this work was done when Rajat and Ankit were at IIT Kanpur.

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

embeddings of the label vector; (2) The model can be seen as learning a set of topics over the label space, which implicitly amounts to learning a (overlapping) clustering of the labels, thereby imposing an additional structural assumption into our multi-label learning model; (3) As we show, the resulting classification rule for our model can be seen as learning nonlinear classifier for each label, with the decision boundaries having intuitive geometric interpretation; (4) The model admits efficient inference via Gibbs sampling or expectation maximization, with a computational cost that scales in the number of nonzeros in the label matrix.

In our experiments, our model compares favorably with state-of-the-art methods and outperforms these methods especially on difficult multi-label learning tasks for which linear models do not suffice. Moreover, our model can also be used as a scalable *deep* topic model for document collections, where each document also has some associated meta-data given in the form of arbitrary feature vectors (Mimno and McCallum, 2008).

2 DEEP GENERATIVE MODEL FOR NONLINEAR MULTI-LABEL LEARNING

In multi-label learning, we assume that we are given N training examples $f(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)g$ with $\boldsymbol{x}_n \; 2 \; \mathsf{R}^D$ and $\boldsymbol{y}_n \; 2 \; f0, 1g^L = [y_{1,n}, \ldots, y_{L,n}]$, $n = 1, \ldots, N$. The goal is to learn a model that can predict the label vector $\boldsymbol{y}_* \; 2 \; f0, 1g^L$ for a new test input $\boldsymbol{x}_* \; 2 \; \mathsf{R}^D$. Often, instead of the actual binary labels, we are just interested in predicting the relative scores for the L labels (or the *probabilities* of each binary label $y_{\ell,*}$, $\ell = 1, \ldots, L$ being equal to 1).

We assume that each observation $(\boldsymbol{x}_n, \boldsymbol{y}_n)$ is associated with a *deep* hierarchy of latent factors $\boldsymbol{u}_n^{(t)} \; 2 \; \mathsf{R}_+^{K_t}$, $t = 1, \ldots, T$, which generates $\boldsymbol{y}_n$ as follows: Layer 1 latent factor $\boldsymbol{u}_n^{(1)}$ generates an L-dimensional *latent count* vector $\boldsymbol{m}_n \; 2 \; \mathsf{Z}_+^L$ from a Poisson distribution, and we then generate the $L \quad 1$ binary label vector $\boldsymbol{y}_n$ via a thresholding operation. These two steps can be written as

$$\boldsymbol{m}_n j \boldsymbol{\xi}_n \quad \text{Poisson}(\boldsymbol{\xi}_n) \text{ and } \boldsymbol{y}_n j \boldsymbol{m}_n = \mathsf{I}[\boldsymbol{m}_n > 0] \quad (1)$$

where $\mathsf{I}[m > 0]$ is equal to 1 if $m > 0$ and 0 otherwise. The Poisson rate parameter $\boldsymbol{\xi}_n$ is defined as $\boldsymbol{\xi}_n = \sum_{k_1=1}^{K_1} \lambda_{k_1}^{(1)} u_{n,k_1}^{(1)} \boldsymbol{v}_{k_1}^{(1)} = \mathbf{V}^{(1)} \; {}^{(1)} \boldsymbol{u}_n^{(1)}$, which is a weighted combination of K_1 "basis vectors" $\mathbf{V}^{(1)} = [\boldsymbol{v}_1^{(1)}, \ldots, \boldsymbol{v}_{K_1}^{(1)}]$, where $\mathbf{V}^{(1)} \; 2 \; \mathsf{R}_+^{L \times K_1}$. Note that the Poisson(.) and $\mathsf{I}[.]$ in Eq. 1 denote point-wise operations defined on vectors, i.e., $m_{\ell,n} \quad \text{Poisson}(\xi_{\ell,n})$ and $y_{\ell,n} = \mathsf{I}[m_{\ell,n} > 0]$. The likelihood model given by Eq. (1) is known as the Bernoulli-Poisson (BP) link (Rai et al., 2015; Zhou, 2015) and is preferred over the commonly used logistic/probit likelihood since it only requires evaluating the likelihood for the nonzero entries in $\boldsymbol{y}_n$ (Rai et al., 2015; Zhou, 2015). This is especially appealing for multi-label learning problems since the label vector tends to be highly sparse. Also note that, for the BP likelihood, we can write $y_{\ell,n} j \xi_{\ell,n} \quad \text{Bernoulli}[1 \quad \exp(\quad \xi_{\ell,n})]$.

We can view $\mathbf{V}^{(1)} = [\boldsymbol{v}_1^{(1)}, \ldots, \boldsymbol{v}_{K_1}^{(1)}] \; 2 \; \mathsf{R}_+^{L \times K_1}$ as a set of K_1 *topics*, with each of its columns representing a (unnormalized) *distribution* over the L labels. As we will show later, our model actually learns a deep hierarchy of topics with $\mathbf{V}^{(1)}$ denoting the set of layer-1 topics.

Fig. 1 shows the graphical model in plate notation. For brevity, we do not show the global parameters $\mathbf{V}^{(1)}$, $^{(1)}$ that the latent count vector $\boldsymbol{m}_n$ depends on, as well as the other global parameters that the latent factors $\boldsymbol{u}_n^{(t)} \; 2 \; \mathsf{R}_+^{K_t}$, $t = 1, \ldots, T$ in the deep hierarchy depend on (we collectively denote those by $f\boldsymbol{\beta}^{(t)} g_{t=1}^T$). We also condition the latent factors $f\boldsymbol{u}_n^{(t)} g_{t=1}^T$ on the input features $\boldsymbol{x}_n \; 2 \; \mathsf{R}^D$.

We now describe the rest of the generative model which we showed concisely in Fig. 1. First note that the latent factors $f\boldsymbol{u}_n^{(t)} g_{t=1}^T$ form a deep hierarchy with the following generative story:

$$u_{n,k_T}^{(T)} \quad \text{Gamma}\left(r_{k_T}, \exp(\boldsymbol{x}_n' \boldsymbol{w}_{k_T}^{(T+1)})\right) \quad (2)$$
$$k_T = 1, \ldots, K_T$$
$$\ldots$$
$$u_{n,k_t}^{(t)} \quad \text{Gamma}\Big(\mathbf{V}_{k_t,:}^{(t+1)} \; {}^{(t+1)} \boldsymbol{u}_n^{(t+1)}, \quad (3)$$
$$\exp(\boldsymbol{x}_n' \boldsymbol{w}_{k_t}^{(t+1)})\Big), k_1 = 1, \ldots, K_t$$
$$\ldots$$
$$u_{n,k_1}^{(1)} \quad \text{Gamma}\left(\mathbf{V}_{k_t,:}^{(2)} \; {}^{(2)} \boldsymbol{u}_n^{(2)}, \exp(\boldsymbol{x}_n' \boldsymbol{w}_{k_1}^{(2)})\right) \quad (4)$$
$$k_1 = 1, \ldots, K_1$$

Note that, in the above generative model, the shape parameter of the gamma prior on each component of the latent factor $\boldsymbol{u}_n^{(t)}$, for layers $t = 1, \ldots, T \quad 1$, depends on $\boldsymbol{u}_n^{(t+1)}$, the latent factors of the layer above. The gamma scale parameter is a function of the inputs $\boldsymbol{x}_n$ via a *regression model* with weight vectors $\boldsymbol{w}_{k_t}^{(t+1)} \; 2 \; \mathsf{R}^D$, $k_t = 1, \ldots, K_t$. For the top layer T, the shape parameters are given by $r_{k_T} \; 2 \; \mathsf{R}_+$, $k_T = 1, \ldots, K_T$. Also note that $\mathbf{V}_{k_t,:}^{(t+1)}$ is a row vector of size K_{t+1}, $^{(t+1)}$ is a diagonal matrix of size $K_{t+1} \quad K_{t+1}$. We assume the following priors on these parameters:

$$\lambda_{k_t}^{(t)} \quad \text{Gamma}(a_\lambda^{(t)}, 1/b_\lambda^{(t)})$$

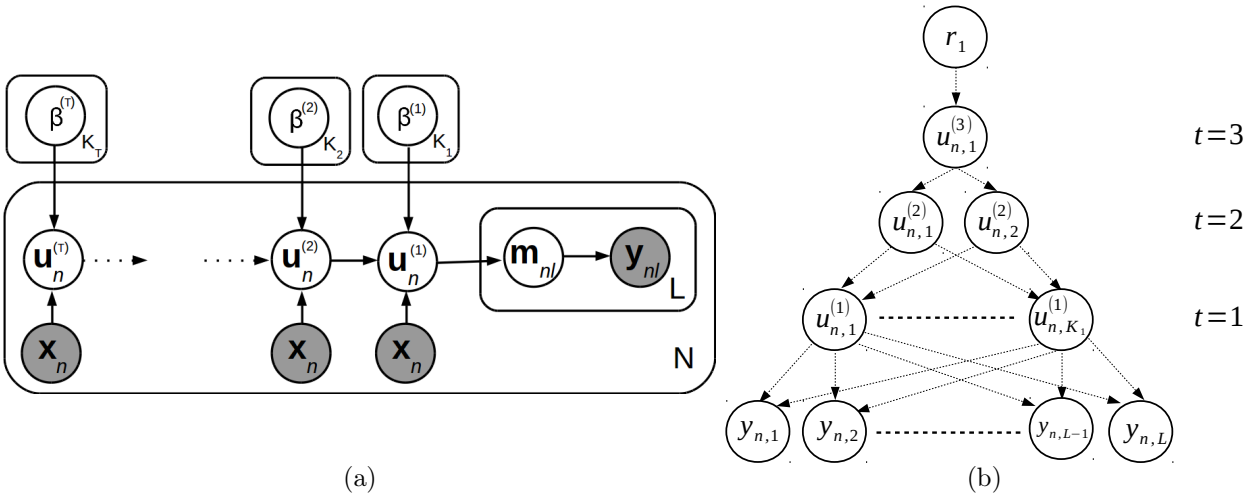

Figure 1: (a) Our generative model in the plate notation. Note: Some of the global variables are not shown for brevity. The $f\boldsymbol{\beta}^{(t)}g_{t=1}^{T}$ collectively denote the global parameters that $f\boldsymbol{u}_n^{(t)}g_{t=1}^{T}$ depend on. The latent factors are also conditioned on the input features $\boldsymbol{x}_n$ (explained in more detail when we describe the full generative model). (b) Visualization as a Bayesian network.

$$k_t = 1, \ldots, K_t \tag{5}$$

$$v_{k_t\ 1,k_t}^{(t)} \quad \text{Gamma}(a_v^{(t)}, 1/b_v^{(t)})$$
$$k_{t-1} = 1, \ldots, K_{t-1}, \quad k_t = 1, \ldots, K_t \tag{6}$$

Each of the regression weight vectors $w_{k_t}^{(t+1)}$ 2 R^D is given a Gaussian prior with zero mean and a diagonal precision matrix $^{(t+1)}$. We assume a gamma prior on the diagonal entries of the precision matrix, which allows us to learn the relevance of different features in the input $\boldsymbol{x}_n$.

We also place gamma priors on each of the gamma shape parameters r_{k_T} needed at the topmost layer. Note that the deep stacked construction of the latent factors is similar in spirit to other deep Bayesian generative models, such as deep exponential family (Ranganath et al., 2015) and gamma belief networks (Zhou et al., 2015), that define a deep hierarchy of latent variables using a top-down generative model.

The proposed deep hierarchy (Eq.(2)-(4)) of latent factors that eventually generates the binary label vector $\boldsymbol{y}_n$ via Eq. (1) is appealing as it enables learning a nonlinear embedding of $\boldsymbol{y}_n$. This is in contrast with most of the existing methods that can only learn a linear embedding (Rai et al., 2015; Yu et al., 2014).

In addition to learning a nonlinear embedding via a deep hierarchy of latent factors, our model has several other distinguishing properties: (1) It allows conditioning the latent factor in each layer on arbitrary features (which in our case is $\boldsymbol{x}_n$, where our goal is to learn a multi-label learning model); and (2) The overall model has a nice geometric interpretation (Sec. 3) in terms of defining *nonlinear* classification boundaries which can be seen as the union of multiple convex polytopes .

For the architectural choice of the deep network, we choose K_t K_{t-1} which leads to a pyramid-like structure of the deep network. The pyramid-like structure of the network ensures low intrinsic dimensionality on the label vectors and, at the same time, leads to parsimony in terms of the number of model parameters to be learned, while still being expressive enough.

Another appealing aspect of our framework is that it is not limited to multi-label learning. Note that the deep generative construction for the label vectors can also be seen as learning a (deep) topic model over a document collection, where the documents are also associated with arbitrary meta-data (Mimno and McCallum, 2008). To see this connection, note that the label vector $\boldsymbol{y}_n$ 2 $f0, 1g^L$ can be thought of as a bag-of-words representation of a document (labels being the words) and $\boldsymbol{x}_n$ 2 R^D can be thought of as additional features or "meta-data" associated with this document. Our generative model can leverage both the document as well as the meta-data to infer a deep hierarchy of topics and a deep topic based representation $f\boldsymbol{u}_n^{(t)}g_{t=1}^{T}$ of each document $\boldsymbol{y}_n$. The hierarchy of topics is given by the matrices $f\mathbf{V}^{(t)}g_{t=1}^{T}$, with $\mathbf{V}^{(t)}$ 2 $\mathsf{R}_+^{K_t\ \ 1\times K_t}$ representing a set of K_t topics, with each topic being a distribution over K_{t-1} "tokens". At layer 1 ($K_0 = L$), these tokens are the L labels themselves, and at higher layers, these token correspond to topics of a "super-topic". Deep topic models have been explored recently using the idea of gamma belief networks (Zhou et al., 2015). However, a key difference is that our model is able to additionally condition on the document meta-

data $\boldsymbol{x}_n$ via a regression model.

3 MODEL PROPERTIES AND GEOMETRIC INTERPRETATION

Our deep generative framework for multi-label learning has several appealing properties. Firstly, predicting the labels for a test example $\hat{\boldsymbol{x}}$ does not require inferring its latent factors $\{\boldsymbol{\theta}^{(t)}\}_{t=1}^{T}$. Instead, the expression for the label probabilities can be obtained in closed-form and it only depends on the test input $\hat{\boldsymbol{x}}$ and the global parameters of our model. This leads to faster inference at test time.

To elaborate more on this, note that, given a new input $\hat{\boldsymbol{x}}$ and global parameters, we can easily marginalize the local variable $\theta_{n,k_t}^{(t)}$ of every node in the deep hierarchy, in a recursive fashion, by using the moment-generating function of Gamma distribution (Rai et al., 2015; Zhou, 2016). The marginalization scheme associates a confidence score $q_{k_t,l}^{(t)}$ to every node $\theta_{n,k_t}^{(t)}$ that depends on all its children nodes and corresponding outgoing edges. The confidence score for a node in the first layer (t=1) is given by

$$q_{k_1,l}^{(1)} = \log\left(1 + \lambda_{k_1}^{(1)} v_{l,k_1}^{(1)} \exp(\hat{\boldsymbol{x}}' \boldsymbol{w}_{k_1}^{(2)})\right) \tag{7}$$

Confidence score of the nodes in the upper layers ($t > 1$) can then be recursively defined as follows:

$$q_{k_t,l}^{(t)} = \log\left[1 + \lambda_{k_t}^{(t)} \left(\sum_{k_{t-1}=1}^{K_{t-1}} v_{k_{t-1},k_t}^{(t)} q_{k_{t-1},l}^{(t-1)}\right) \exp\left(\hat{\boldsymbol{x}}' \boldsymbol{w}_{k_t}^{(t+1)}\right)\right] \tag{8}$$

The predictive distribution of the i^{th} label in the label vector $\hat{\boldsymbol{y}}$ is then calculated by aggregating the confidence scores of all the K_T nodes in the top layer as

$$P(\hat{y}_l = 1 | \hat{\boldsymbol{x}}) = 1 - \exp\left(-\sum_{k_T=1}^{K_T} r_{k_T} q_{k_T,l}^{(T)}\right) \tag{9}$$

For the single layer case ($T = 1$), the above expression is equivalent to the predictive distribution of the Bernoulli-Poisson multi-label learning model from (Rai et al., 2015). However, in the multi-layer construction of our model, Eq. (9) also has an interesting geometric interpretations. It can be observed from the above expression that if there exists atleast one *confident path* from the top layer upto the label l, then the probability of this label being 1 will be large. A confident path is essentially a sequence of nodes $[l, k_1', k_2', \ldots, k_T', r_{k_T^0}]$ in the deep hierarchy such that the model is confident at each step t , that is, the edge weight $(\lambda_{k_{t+1}^0} v_{k_t^0, k_{t+1}^0})$ as well as the regression model based score $(\hat{\boldsymbol{x}}' w_{k_t^0}^{(t+1)})$ is sufficiently large. Consequently, the region of the input space satisfying $\hat{\boldsymbol{x}}' w_{k_t^0}^{(t+1)} > c_t \; \forall t$, for the layer-specific thresholds c_t, leads to a convex-polytope-like region in the feature space that encloses positive labels.

Another interesting geometric interpretation and explanation of our model's ability to learn highly nonlinear classification boundaries comes from the fact that the model can be seen as a union of $(\prod_{t=0}^{T} K_t)$ committees of T experts where experts are shared across committees. Moreover, each expert uses a linear hyperplane to generate a score. For predicting a specific label, the model uses a union of $(\prod_{t=1}^{T} K_t)$ committees of experts. All the experts of at least one committee must agree simultaneously to generate high probability of success and therefore each committee resembles a convex polytope-like region of positive labels. Note that all the labels share $(\prod_{t=2}^{T} K_t)$ experts in their committees. Moreover, K_1 hyperplanes are shifted in a parallel fashion to generate $K_0 \times K_1$ label-specific experts. Sharing experts leads to a highly flexible model without increasing the number of parameters.

4 INFERENCE

Although our deep generative model is not natively conjugate, we are able to leverage auxiliary-variable-augmentation techniques (Rai et al., 2015; Zhou, 2016; Zhou et al., 2015) to obtain a model that is locally conjugate and admits simple inference via closed-form Gibbs sampling or Expectation-Maximization . For each training example $(\boldsymbol{x}_n, \boldsymbol{y}_n)$, we augment latent counts $m_{n,k_t,k_{t+1}}^{(t+1)}$ corresponding to each edge in the deep hierarchy (see Fig. 1b). Using these latent counts, we associate two latent counts with each node $u_{n,k_t}^{(t)}$: (1) *outgoing* latent counts $m_{n,\_,k_t}^{(t)}$; (2) *incoming* latent counts $m_{n,k_t,\_}^{(t+1)}$. The outgoing latent count is defined as the sum of latent counts associated with all the outgoing edges from that node, $m_{n,\_,k_t}^{(t)} = \sum_{k_{t-1}=1}^{K_{t-1}} m_{n,k_{t-1},k_t}^{(t)}$. Similarly, the incoming latent count associated with node $u_{n,k_t}^{(t)}$, that is $m_{n,k_t,\_}^{(t+1)}$, is defined as $\sum_{k_{t+1}=1}^{K_{t+1}} m_{n,k_t,k_{t+1}}^{(t+1)}$. The latent count, $m_{n,l}^{(1)}$, described in Eq. 1 can be thought of as an incoming latent counts $m_{n,k_0,\_}^{(1)}$ for a node k_0 of the 0^{th} layer (i.e., the observed label vector layer).

The latent counts $m_{n,k_0,\_}^{(1)}$ are drawn from a truncated Poisson distribution as

$$(m_{n,k_0,\_}^{(1)} | -) \sim y_{k_0,n} \; \text{Pois}_+\Big(\sum_{k_1} \lambda_{k_1}^{(1)} v_{k_0,k_1}^{(1)} u_{n,k_1}^{(1)}\Big) \tag{10}$$

The latent counts of the successive layer can be computed recursively: In particular, given $m^{(t+1)}_{n,k_t,\_}$ for a layer t, we can obtain $m^{(t+1)}_{n,k_t,k_{t+1}}$ using Poisson and Multinomial-Poisson equivalence (Zhou et al., 2015):

$$\left(\ldots, m^{(t+1)}_{n,k_t,k_{t+1}}, \ldots\right) \quad \text{Mult}\left(m^{(t+1)}_{k_t,n,-}\middle|\ldots, \frac{\lambda^{(t+1)}_{k_{t+1}} v^{(t+1)}_{k_t,k_{t+1}} u^{(t+1)}_{n,k_{t+1}}}{\sum_{k_{t+1}} \lambda_{k_{t+1}} v^{(t+1)}_{k_t,k_{t+1}} u^{(t+1)}_{n,k_{t+1}}}, \ldots\right) \tag{11}$$

The Eq. (11) and Eq. (12) hold trivially for $t = 0$ and it also holds for $t > 1$ by using the Chinese Restaurant Table-negative Binomial (CRT-NB) equivalence with Poisson-SumLog(Zhou and Carin, 2015).

The subsequent $m^{(t+1)}_{n,\_,k_{t+1}}$ can be obtained by aggregating the $m^{(t+1)}_{n,k_t,k_{t+1}}$ over k_t. By using the additive property of the Poisson distribution, we have

$$m^{(t+1)}_{n,\_,k_{t+1}} \quad \text{Pois}(u^{(t+1)}_{n,k_{t+1}} q^{(t+1)}_{n,k_{t+1}}) \tag{12}$$

where $q^{(t+1)}_{n,k_{t+1}}$ can be thought of as a proportionality factor similar to $q^{(t)}_{k_t,l}$ in Eq. (8) and is defined as: $q^{(t+1)}_{n,k_{t+1}} = \lambda^{(t+1)}_{k_{t+1}} \sum_{k_t} v^{(t+1)}_{k_t,k_{t+1}} \log\left(1 + q^{(t)}_{n,k_t} \exp(\boldsymbol{x}'_n w^{(t+1)}_{k_t})\right)$, $q^{(0)}_{n,k_0} = \exp(1) \quad 1$ and $w^{(1)}_{k_0} = \mathbf{0}$.

Once the outgoing latent count $m^{(t)}_{n,\_,k_t}$ of a node is available, its incoming latent count $m^{(t+1)}_{n,k_t,\_}$ can be calculated as $m^{(t+1)}_{n,k_t,\_} \quad \text{CRT}\left(m^{(t)}_{\_,n,k_t}, r^{(t)}_{n,k_t}\right)$, where $r^{(t)}_{n,k_t}$ is $\sum_{k_{t+1}=1}^{K_{t+1}} v^{t+1}_{k_t,k_{t+1}} \lambda^{(t+1)}_{k_{t+1}} u^{(t+1)}_{n,k_{t+1}}$ when $t < T$ and r_{k_T} otherwise. This procedure is repeated to find latent counts at each layer.

Given these latent counts, we can now leverage Poisson-gamma conjugacy to find the posterior distributions of the local variables $u^{(t)}_{n,k_t}$ and the global variables $v^{(t)}_{k_t\ 1,k_t}$ and $\lambda^{(t)}_{k_t}$, as follows:

$$u^{(t)}_{n,k_t} \quad \text{Gamma}\left(r^{(t)}_{n,k_t} + m^{(t)}_{n,\_,k_t}, \frac{\exp\left(\boldsymbol{x}'_n w^{(t+1)}_{k_t}\right)}{\exp\left(\boldsymbol{x}'_n w^{(t+1)}_{k_t}\right) q^{(t)}_{n,k_t} + 1}\right) \tag{13}$$

$$v^{(t)}_{k_t\ 1,k_t} \quad \text{Gamma}\left(a^{(t)}_v + \sum_{n=1}^{N} m^{(t)}_{n,k_t\ 1,k_t}, \left(b^{(t)}_v + \lambda_{k_t} \sum_n u^{(t)}_{n,k_t} \psi^{(t-1)}_{n,k_t\ 1}\right)^{-1}\right) \tag{14}$$

$$\lambda^{(t)}_{k_t} \quad \text{Gamma}\left(a^{(t)}_\lambda + \sum_{n=1}^{N} m^{(t)}_{n,\_,k_t}, \left(b^{(t)}_\lambda + \sum_{k_t\ 1} \sum_n (v^{(t)}_{k_t\ 1,k_t} u^{(t)}_{n,k_t} \psi^{(t-1)}_{n,k_t\ 1})\right)^{-1}\right) \tag{15}$$

where $\psi^{(t-1)}_{n,k_t\ 1}$ is equal to $\log(1 + q^{(t-1)}_{n,k_t\ 1} \exp(\boldsymbol{x}'_n w^{(t)}_{k_t\ 1}))$. Although, the posterior of the regression weight $\boldsymbol{w}^{(t)}_{k_t}$ does not have any closed-form solution, Poisson-gamma marginalization leads to a negative binomial distribution which can be transformed into a Gaussian likelihood using Pólya-Gamma augmentation (Polson et al., 2013). Given the Pólya-Gamma variables

$$(\omega^{(t)}_{n,k_t} j \ ) \quad \text{PG}\left(m^{(t)}_{n,\_,k_t} + r^{(t)}_{n,k_t}, \boldsymbol{x}'_n w^{(t+1)}_{k_t} + \ln(q^{(t)}_{n,k_t})\right) \tag{16}$$

the posterior of $w^{(t+1)}_{k_t}$ can be written as $N(\mu^{(t+1)}_{w_{k_t}}, \ ^{(t+1)}_{w_{k_t}})$ where

$$^{(t+1)}_{w_{k_t}} = \left((\ ^{(t+1)})^{-1} + \sum_{n=1}^{N} \omega^{(t)}_{n,k_t} \boldsymbol{x}_n \boldsymbol{x}'_n\right)^{-1} \tag{17}$$

and

$$\mu^{(t+1)}_{w_{k_t}} = \ ^{(t+1)}_{w_{k_t}} \left[\sum_{n=1}^{N} \left(\omega^{(t)}_{n,k_t} \ln q^{(t)}_{n,k_t} + 0.5 \quad (m^{(t)}_{n,\_,k_t} \quad r^{(t)}_{n,k_t})\right) \boldsymbol{x}_n\right] \tag{18}$$

We omit the equations for hyperparameter inference for the sake of brevity.

5 RELATED WORK

A prominent class of methods for multi-label learning has been based on learning low-dimensional embeddings of label vectors (Bhatia et al., 2015; Kapoor et al., 2012; Rai et al., 2015; Yu et al., 2014). These methods perform multi-label learning by assuming that the label vectors have a low-dimensional representation (label embeddings), which amounts to the label matrix being a low-rank matrix. However, these methods are usually limited to learning linear (or locally linear) embeddings. Another way to incorporate nonlinearity in the embeddings is by using kernel-based label embeddings (Li and Guo, 2015). However, kernel-based methods tend to be slow at training as well as test time, due to the requirement of storing all the training data.

Probabilistic/Bayesian models for multi-label learning are relatively fewer. These are usually based on latent factor models for the binary label vectors. However, the inference cost for these models is usually

prohibitive (Kapoor et al., 2012) and these models do not scale to large data sets. Since the label vectors are high-dimensional but extremely sparse in nature (i.e., have very few nonzeros), it is desirable to have models whose computational cost scales in the number of nonzeros in the label matrix. In (Rai et al., 2015), a Bayesian latent factor model (BMLPL) based on a Bernoulli-Poisson-gamma generative model was proposed, which exploited the label matrix sparsity to design efficient inference algorithms for multi-label learning. Our model is similar in spirit to the work in (Rai et al., 2015). However, it is significantly more general and considerably more flexible due to the deep generative model on the label vectors. The BMLPL model proposed in (Rai et al., 2015) is a special case of our model when the number of layers is equal to 1. We use this model as one of the baselines in our experiments. In contrast to the model in (Rai et al., 2015), our deep generative model allows learning nonlinear label embeddings, provides interesting geometric interpretation/explanation for the nonlinearity of our model (in terms of what kind of nonlinear decision boundaries it learns), while still enjoying simplicity of the inference procedure. The inference cost in our model still scales in the number of nonzeros in the label matrix, which is appealing for multi-label learning problems involving large label matrices.

Among other related work, recently (Cissé et al., 2016) proposed a deep architecture for multi-label learning. This model clusters the labels into a Markov Blanket Chain and then leverages these clusters to train a deep neural network. While this work is also similar in spirit to our model, there are a few key differences: (1) It has a feedforward architecture and lacks a proper generative model for the labels; (2) It cannot leverage the label matrix sparsity; and (3) It lacks the nice geometric interpretability offered by our approach. Moreover, unlike the model in (Cissé et al., 2016), our model can be applied to not only multi-label learning problems, but also to topic modeling tasks where the documents also have meta-data associated with them. Recent work has also explored multi-label learning for text documents using text CNN (Liu et al., 2017). However, to the best of our knowledge, none of the existing deep learning models, including (Cissé et al., 2016; Liu et al., 2017) are based on generative models.

Deep generative models based on the Poisson-gamma latent factor model construction (Zhou et al., 2015) have also been proposed for modeling high-dimensional count data (e.g., documents represented as vector of word-counts). These models have also been extended to multimodal settings (Henao et al., 2016). Although the model proposed in (Henao et al., 2016) can be used in multi-label setting, it is (1) limited to count-valued features and moreover, (2) difficult to do inference on at test time, and (3) does not provide any geometric interpretation unlike our model. Our framework can be seen as a generarization of these models where we condition the latent factors on *arbitrary* feature vectors, and the deep architecture naturally leads to a *nonlinear* classification model from the feature space to the label space. To the best of our knowledge, ours is the first fully Bayesian method, based on a deep generative model for the problem of multi-label learning, that can handle arbitrary type of features, while also leveraging the sparsity of label matrix for computational speed-ups.

6 Experiments

In this section, we present the quantitative and qualitative results of our model. First, inspired by the geometric interpretation of decision boundaries, we generate a synthetic dataset to show that shallow models can't compete with deep models for a non-linear dataset. Then, we benchmark our model with other state-of-the-art models on the benchmark datasets. Moreover, we illustrate the ability of our model to learn a hierarchical topic model over the documents.

We have implemented an EM algorithm (Scott and Sun, 2013) to generate the results in the following experiments. We use the fact that the simple closed-form expressions of expectations of the local variables are available including Polya-Gamma and Chinese Restaurant Table random variables. Note that a straightforward implementation to calculate the updates of $w_{k_t}^{(t+1)}$ would require inversion of a $D \quad D$ matrix. We avoid this computationally heavy step by calculating the approximate solution to the following linear system for $w_{k_t}^{(t+1)}$ by using only a few iterations of Conjuate Gradient Method (Bertsekas, 1999) (see Eq. (17),(18)) $\left(\begin{smallmatrix} (t+1) \\ w_{k_t} \end{smallmatrix} \right)^{-1} w_{k_t}^{(t+1)} = d_{k_t}^{(t+1)}$ where $d_{k_t}^{(t+1)} = \left[\sum_{n=1}^{N} \left(\quad \omega_{n,k_t}^{(t)} \ln q_{n,k_t}^{(t)} + 0.5 \quad (m_{n,\_,k_t}^{(t)} \quad r_{n,k_t}^{(t)}) \right) \boldsymbol{x}_n \right]$. We run our EM-CGS algorithm for 500 iterations and the algorithm converged in all these cases.

6.1 Synthetic Dataset

We generate a synthetic dataset to validate the claim made in Section 3 that the model learns a union of multiple convex polytopes. With a model of depth T, each of the inferred polytope will be the intersection of T half-spaces. Therefore, if the region (in the covariate space R^D) corresponding to $l = 1$, for a label l, is intersection of multiple half-spaces, then the performance of a shallow model would saturate after a particular width. With this idea, we generate the synthetic data that has $L = 10$, $d = 50$ and contains 5000 instances. The label $y_{n,l}$ is set to be 1 if x_n lies in the convex

Table 1: Performance on synthetic data with varying depth and width of the deep architecture

Layer Sizes = $[K_1, K_2, K_3]$	[4]	[10]	[4,2]	[4,2,1]	[4,2,2]
AUC	0.898	0.909	0.930	0.930	0.935

Table 2: Statistics of the data sets. D and L denote the average number of non-zero elements per example in features and labels respectively.

Data set	D	L	N	L	D
bibtex	1836	159	4880	2.40	68.74
delicious	500	983	12920	19.03	18.17
eurlex	5000	3993	17413	5.30	236.69

Table 3: AUC-ROC scores on real datasets

Dataset	CPLST	BCS	WSABIE	LEML	BMLPL	ADIOS	DBMLPL		
							T= 1	T = 2	T = 3
bibtex	0.888	0.861	0.918	0.904	0.921	0.876	0.927	0.930	0.932
delicious	0.883	0.800	0.856	0.889	0.895	0.904	0.899	0.903	0.906
eurlex	-	-	0.865	0.946	0.952	0.877	0.956	0.961	0.962

polytopes defined by 3 hyperplanes $w^{(3)}_{k_l}$, $w^{(2)}_{k^0_l}$ and $w^{(1)}_{k^{00}_l}$. The hyperplanes are shared across labels such that number of distinct $w^{(1)}_k, w^{(2)}_k$ and $w^{(3)}_k$ are 4, 2 and 2 respectively. The results are reported in Table 1 which clearly shows that in absence of a sufficient deep model that captures the intrinsic dimensionality of data, the performance suffers adversely.

6.2 Real Datasets

We first report our results on standard benchmark datasets for multi-label learning. We use three benchmark datasets(Yu et al., 2014) — Bibtex, Eurlex-4k, and Delicious — to compare our model with other state-of-the-art models. Note that all these datasets have highly sparse features and labels (see Table 2).

The choice of the number of layers and sizes of these layers is the most important decision that has to be made while using this model, which is further shown in Section 6.1. It is worth mentioning that between two architectures with the same number of nodes, the model with a higher depth will have a lower number of parameters in our case. We restrict the size of t^{th} layer for t 2 f2, 3g to half of the size of the $(t \quad 1)^{\text{th}}$ layer. The results are only reported for DBMLPL with the total number of nodes between 0.1L and 0.5L. On the other hand, for the rest of the models we have picked their best performance with the embedding size between 0.2L and L. We compare our model with the depth as T=1, 2 and 3 with other state-of-the-art baselines in Table 3.

Among the baselines used in Table 3, BMLPL(Rai et al., 2015) is a Bayesian model based on learning low-dimensional embeddings of the labels, that are conditioned on the features (similar to our model). It can be thought of as a special case of our model with a single layer. LEML(Yu et al., 2014) minimizes various loss functions like logistic, squared and hinge loss for low dimensional embedding in an ERM framework. WSABIE(Weston et al., 2011) tries to learn a joint embedding of both the features as well as the labels. BCS(Kapoor et al., 2012) is also a Bayesian model that uses random projection of labels and regression of the features against that random projection in a single probabilistic framework. CPLST(Chen and Lin, 2012) also learns label embeddings, which are conditioned on the features, but it uses hamming loss as its loss function. We also compare our model with a state-of-the-art deep learning architecture ADIOS (Cissé et al., 2016) which leverages the relationship among labels by creating a deep output space to perform multi-label learning. As shown in Table 3, our model outperforms the various baselines on all the datasets. Also, the classification accuracies improve as we increase the number of layers.

We would like to highlight that our baselines consist of very strong, state-of-the-art Bayesian models and deep learning models for multi-label learning, and our model yields moderate but consistent performance gains across all the datasets. Moreover, our experiment on the carefully simulated difficult nonlinear dataset (Table 1) demonstrates that the model is able to learn difficult nonlinear boundaries, which single-layer models like BMLPL cannot.

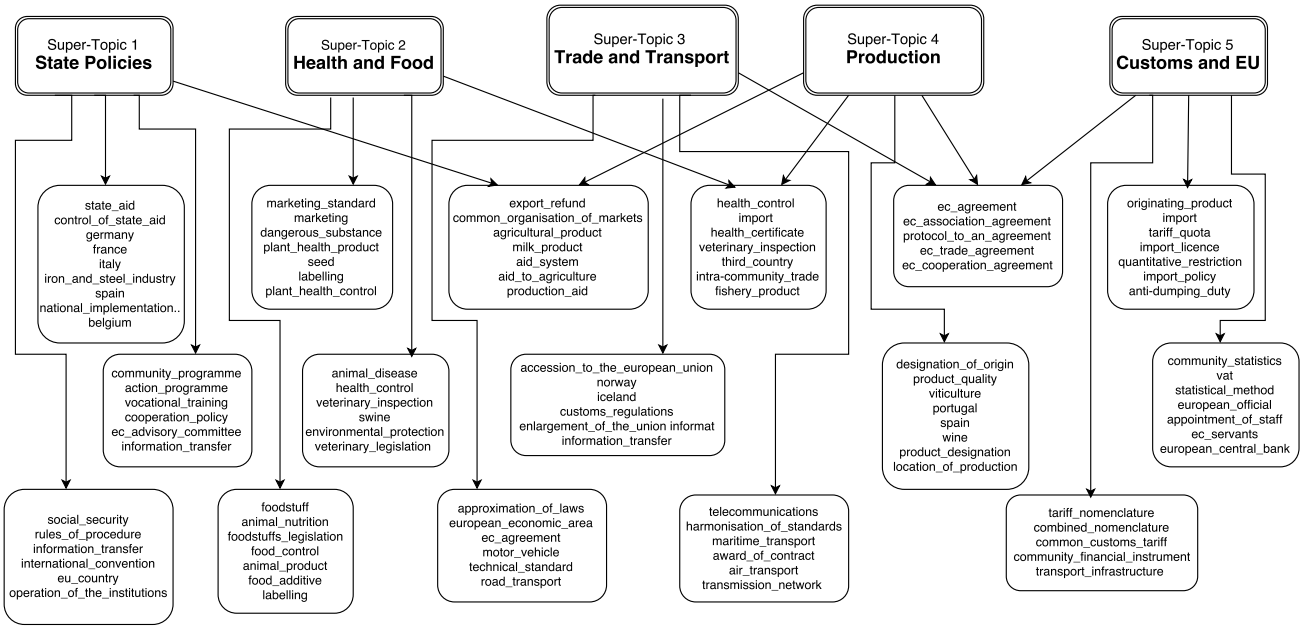

Figure 2: Topics and Super topics inferred from eurlex data

6.3 Qualitative Analysis: Deep Topic Modeling

The model explicitly imposes a low-dimensional structure on labels through a factor-loading matrix $V^{(1)}$ $^{(1)}$. Moreover, each layer (t) learns an unnormalized distribution on the layer below $(t \quad 1)$ through $V^{(t)}$ $^{(t)}$ with 0^{th} layer being the observed labels. We use the eurlex-4k dataset that contains about 5000 documents related to European Union Law where each document is annotated with possibly multiple tags. There are a total of 3993 different labels, i.e. tags, spanning diverse areas. We use this dataset to test efficacy of deep topic modeling aspect of our model. We set the $K_1 = 20$ and $K_2 = 5$ for our model and hence expect to learn 20 topics and 5 super-topics over the topics. The obtained topics and super-topics are shown in Fig. 2 where we show the top few labels for each topic as well as show the top 4 topics for each super-topic. The model groups semantically similar labels in each topic and is able to learn super-topics such as "Health and Food".

7 Conclusion and Discussion

We have presented a deep generative model for nonlinear multi-label learning. Our Bayesian model is based on learning a deep hierarchy of gamma-distributed latent factors that represent the embeddings of binary label vectors. The model is built using a clear, deep generative model, and admits a simple and efficient inference procedure due to full local conjugacy. Moreover, the sparsity of label vectors further reduces the computational cost of inference. The model also offers a nice geometric interpretation, which explains its effectiveness in learning complex nonlinear decision boundaries.

While multi-layer neural networks can also be tried for multi-label learning (in fact, the ADIOS baseline we compared against is precisely that!), taking a generative approach enables us to construct a likelihood model that is appropriate for high-dimensional binary label vectors, and also leverage the sparsity of the label vectors for computational efficiency via the Bernoulli-Poisson link. Extensions to semi-supervised and active learning would be an interesting future directions.

Although, in this paper, we used a Gibbs sampling based inference procedure, other inference methods such as stochastic variational inference can allow applying our model on very large-scale data sets. Developing such inference algorithms for our model will be an interesting direction of future work. Finally, our fully Bayesian framework also opens the door to other interesting extensions such as performing active learning (Vasisht et al., 2014) to acquire the most informative labels.

Acknowledgements: PR acknowledges support from Visvesvaraya Young Faculty Fellowship by DST India, and P.K. Kelkar Young Faculty Fellowship by IIT Kanpur, and grants from IBM Research, Microsoft Research India, and Google.

References

R. Babbar and B. Schölkopf. DiSMEC- distributed sparse machines for extreme multi-label classification. In *WSDM*, 2017.

D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, 2015.

Y.-N. Chen and H.-T. Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, 2012.

M. Cissé, C. M. Al-Shedivat, and S. Bengio. Adios: Architectures deep in output space. In *ICML*, 2016.

E. Gibaja and S. Ventura. Multilabel learning: A review of the state of the art and ongoing research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2014.

E. Gibaja and S. Ventura. A tutorial on multilabel learning. *ACM Comput. Surv.*, 2015.

R. Henao, J. T. Lu, J. E. Lucas, J. Ferranti, and L. Carin. Electronic health record analysis via deep poisson factor models. *The Journal of Machine Learning Research*, 17(1):6422–6453, 2016.

H. Jain, Y. Prabhu, and M. Varma. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *KDD*, 2016.

A. Kapoor, R. Viswanathan, and P. Jain. Multilabel classification using bayesian compressed sensing. In *NIPS*, 2012.

D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *NIPS*, 2014.

X. Li and Y. Guo. Multi-label classification with feature-aware non-linear label space transformation. In *IJCAI*, 2015.

J. Liu, W.-C. Chang, Y. Wu, and Y. Yang. Deep learning for extreme multi-label text classification. In *SIGIR*, 2017.

D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *UAI*, 2008.

N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013. doi: 10.1080/01621459.2013.829001.

Y. Prabhu and M. Varma. FastXML: a fast, accurate and stable tree-classifier for extreme multi-label learning. In *KDD*, 2014.

P. Rai, C. Hu, R. Henao, and L. Carin. Large-scale bayesian multi-label learning via topic-based label embeddings. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3222–3230, Cambridge, MA, USA, 2015. MIT Press.

R. Ranganath, L. Tang, L. Charlin, and D. Blei. Deep exponential families. In *AISTATS*, 2015.

J. G. Scott and L. Sun. Expectation-maximization for logistic regression, 2013.

D. Vasisht, A. Damianou, M. Varma, and A. Kapoor. Active learning for sparse bayesian multilabel classification. In *KDD*, 2014.

J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In *IJCAI*, 2011.

C. Xu, D. Tao, and C. Xu. Robust extreme multi-label learning. In *KDD*, 2016.

H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon. Large-scale multi-label learning with missing labels. In *ICML*, 2014.

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

M. Zhou. Softplus Regressions and Convex Polytopes. *ArXiv e-prints*, Aug. 2016.

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

M. Zhou, Y. Cong, and B. Chen. The poisson gamma belief network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3043–3051. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5645-the-poisson-gamma-belief-network.pdf`.