Bayesian Multi-label Learning with Sparse Features and Labels, and Label Co-occurrences

He Zhao* **Piyush Rai†** **Lan Du*** **Wray Buntine***
*Faculty of IT, Monash University, Australia
{he.zhao, lan.du, wray.buntine}@monash.edu
†Department of CSE, IIT Kanpur, India
piyush@cse.iitk.ac.in

Abstract

We present a probabilistic, fully Bayesian framework for multi-label learning. Our framework is based on the idea of learning a joint low-rank embedding of the label matrix and the label co-occurrence matrix. The proposed framework has the following appealing aspects: (1) It leverages the sparsity in the label matrix and the feature matrix, which results in very efficient inference, especially for sparse datasets, commonly encountered in multi-label learning problems, and (2) By effectively utilizing the label co-occurrence information, the model yields improved prediction accuracies, especially in the case where the amount of training data is low and/or the label matrix has a significant fraction of missing labels. Our framework enjoys full local conjugacy and admits a simple inference procedure via a scalable Gibbs sampler. We report experimental results on a number of benchmark datasets, on which it outperforms several state-of-the-art multi-label learning models.[1]

1 Introduction

Multi-label learning [Gibaja and Ventura, 2015, Prabhu and Varma, 2014, Jain et al., 2016, Babbar and Schölkopf, 2017] refers to the problem of learning to assign a subset of relevant labels to each object, given a large set of candidate labels. Each object is thus associated with a binary label vector, which denotes the presence/absence of each of the candidate labels. Multi-label learning problems are ubiquitous in a wide variety of applications, such as image/document tagging, recommender system, ad-placement.

In multi-label learning problems encountered in modern applications, it is common to have datasets characterized by instances defined by sparse, high-dimensional feature vectors, in addition to the corresponding label vectors themselves being sparse and high-dimensional. Moreover, often the label vector may be incomplete since it is usually not possible to completely annotate an instance with all of the relevant labels. Multi-label learning problems thus need to routinely deal with missing labels in the label vector of each training instance. Finally, scalability is another challenge in multi-label learning problems. Given the high degree of sparsity of features and labels, it is desirable to have multi-label learning algorithms that can leverage this sparsity during training/test time, and can consequently scale to large-scale problems.

Motivated by these issues and desiderata, we present a probabilistic framework for multi-label learning, which is capable of addressing these issues effectively, in a principled manner. Our framework is based on a generative latent factor model for the binary label matrix. This latent factor model is based on an efficient Poisson-Dirichlet-gamma non-negative factorization [Zhou et al., 2012] of the binary label matrix, which scales in the number of nonzeros in the label matrix. Moreover, we condition the latent factors on the instance features in a way that effectively utilizes the feature sparsity and further improves the scalability. Leveraging both instance label vector as well as instance feature vector sparsity leads to a very efficient inference for our model.

We further augment our model with a latent factor model for the label co-occurrences. Information about label co-occurrences can be obtained from an external source (e.g., a text corpus such as Wikipedia) and this information can be helpful, especially in predicting labels that are rare in the data (e.g., for which there are very training examples) or in cases where the label matrix could have a large fraction of labels as missing.

[1] Code at `https://github.com/ethanhezhao/BMLS`

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

Our latent factor model for the label co-occurrence is learned jointly with the latent factor model for the label matrix, and sharing the latent factors of the label helps in effectively transferring information from the label co-occurrences.

Our model enjoys local conjugacy, which leads to a very simple and highly efficient Bayesian inference via Gibbs sampling. Our model is considerably more scalable as compared to other state-of-the-art Bayesian models for multi-label learning, while achieving comparable and better prediction accuracies.

2 Background and Notation

In the multi-label learning problem, we assume that we are given an $D \times N$ instance feature matrix $\mathbf{X}$ and an $L \times N$ instance label matrix $\mathbf{Y} \in \{0,1\}^{L \times N}$, where N, D, L are the number of instances, the dimension of features, the dimension of labels, respectively. Both matrices are assumed to be highly sparse. In this paper, we focus on binary features, which are quite common, especially in large-scale multi-label learning tasks. An example would be in document classification: each instance is a text document which is associated with a binary feature vector indicating the presence/absence of words. The goal of multi-label learning is to use the feature matrix and the label matrix to learn a model that can predict the label vector $\boldsymbol{y}_*$, given the feature vector $\boldsymbol{x}_*$ of a new instance.

Our model is based on the idea of factorizing the label matrix $\mathbf{Y}$, which is equivalent to learning a low-dimensional embedding $\boldsymbol{\theta}_i$ for the label vector $\boldsymbol{y}_i$ (i.e., the i^{th} column vector of $\mathbf{Y}$) of each instance i [Yu et al., 2014, Rai et al., 2015, Mineiro and Karampatziakis, 2015]. The embedding $\boldsymbol{\theta}_i$ is, in turn, conditioned on the feature vector $\boldsymbol{x}_i$ (i.e., the i^{th} column vector of $\mathbf{X}$) associated with that instance. Given the feature vector of a new instance $\boldsymbol{x}_*$, its embedding $\boldsymbol{\theta}_*$ can be computed and it label vector $\boldsymbol{y}_*$ can be predicted/decoded from $\boldsymbol{\theta}_*$. Different label embedding models vary in how the embeddings are conditioned on the features and how the embeddings are decoded to produce the label vector at test time.

Our model has the following distinguishing aspects as compared to other existing label embedding methods for multi-label learning: (1) Learning the embeddings by our model scales in the number of nonzeros in the label and feature matrices, and (2) The model can effectively leverage the label co-occurrence matrix, if available. The latter property is especially useful when a significant fraction of the labels are missing in the label matrix and/or if the number of training instances are very small.

3 The Model

Our model assumes that each entry $y_{l,i} \in \{0,1\}$ of the label matrix $\mathbf{Y}$ is generated by first drawing a *latent* count $z_{l,i}$ from the Poisson distribution with rate parameter $\psi_{l,i}$ and then thresholding the count at 1.

$$y_{l,i} = \mathbf{1}_{z_{l,i}>0} \quad (1)$$

$$z_{l,i} \sim \text{Poisson}(\psi_{l,i}) \quad (2)$$

where $\mathbf{1}.$ is the indicator function. To assist clarity, we further denote the latent count matrix as $\mathbf{Z} \in \mathbb{Z}^{L \times N}$ and the Poisson rate matrix as $\boldsymbol{\Psi} \in {\mathbb{R}_+}^{L \times N}$.

By integrating $z_{l,i}$ out, the above generative process for $y_{l,i}$ can be shown to be equivalent to

$$y_{l,i} \sim \text{Bernoulli}[1 - \exp(-\psi_{l,i})] \quad (3)$$

which is the Bernoulli-Poisson (BP) link function [Zhou, 2015] for binary observations. A particularly appealing aspect of the BP link (as opposed to other link function for binary observations, such as logistic/probit) is that the inference cost only depends on the number of nonzeros in the data [Zhou, 2015], making it an ideal choice for the problems involving the large-scale multi-label learning problems with sparsity. Specifically, if the $y_{i,l} = 0$, $z_{i,l} = 0$ with probability one. Therefore we only need to infer the latent count $z_{i,l}$ for those labels $y_{i,l}$ that are nonzero. That is how the sparsity of the label matrix is leveraged in our model.

3.1 A Low-Rank Model for Label Matrix

Most real-world multi-label learning datasets consist of high-dimensional labels vectors. However, the labels tend to be related to each other. Therefore, a popular assumption used in multi-label learning is to use a low-rank approximation for the label matrix, as also used in recent work [Yu et al., 2014, Rai et al., 2015, Mineiro and Karampatziakis, 2015, Bhatia et al., 2015]. To this end, we assume that the Poisson parameter matrix $\boldsymbol{\Psi}$ admits a low-rank factorization as follows:

$$\boldsymbol{\Psi} = \boldsymbol{\Phi}^\top \boldsymbol{\Theta} \quad (4)$$

where $\boldsymbol{\Theta} \in {\mathbb{R}_+}^{K \times N}$ and $\boldsymbol{\Phi} \in {\mathbb{R}_+}^{K \times L}$.

For one instance i, the model can be written as:

$$\boldsymbol{y}_i \sim \text{Bernoulli}[1 - \exp(-\boldsymbol{\psi}_i)] \quad (5)$$

$$\boldsymbol{\psi}_i = \boldsymbol{\Phi}^\top \boldsymbol{\theta}_i = \sum_{k=1}^{K} \boldsymbol{\phi}_k \theta_{i,k} \quad (6)$$

The model can be interpreted as follows: The label vector $\boldsymbol{y}_i$ is associated with an embedding $\boldsymbol{\theta}_i$ and $\boldsymbol{\Phi}$

can be considered as K "topics", each a distribution over the L labels. The label vector $\boldsymbol{y}_i$ of instance i can then be thought of as being generated via a linear combination of these K topics through the BP link. The combination weights given by the embedding vector $\boldsymbol{\theta}_i$, with $\theta_{k,i}$ representing the weight of topic k, where $\phi_{k,l}$ represents the weight of label l in topic k. Finally, we impose Dirichlet prior on $\boldsymbol{\phi_k}$:

$$\boldsymbol{\phi_k} \sim \text{Dirichlet}_L(\beta_0, \cdots, \beta_0) \tag{7}$$

3.2 Conditioning Embeddings on Features

To condition the label vector embeddings $\boldsymbol{\theta}_i$ on the feature vector $\boldsymbol{x}_i$, we model $\theta_{k,i}$ a log-linear combination of the instance's features as follows:

$$\theta_{k,i} = b_k \prod_d^D h_{k,d}^{x_{d,i}} \tag{8}$$

where $h_{k,d} \in \mathbb{R}_+$ is a latent variable controlling the influence of feature d on topic k and $b_k \in \mathbb{R}_+$ is a feature-independent bias term. Both $h_{k,d}$ and b_k are drawn from a gamma distribution:

$$h_{k,d}, b_k \sim \text{Gamma}(\mu_0, 1/\mu_0) \tag{9}$$

Figure 1 shows the graphical model for the above construction. Given our model construction, $h_{k,d}$ is expected to have mean 1. The intuition is that, in multi-label learning problems, the number of features D is usually very large but, for most of the instances, only a small subset of these features is discriminative. Therefore, if feature d does not contribute to topic k or is not very informative, then $h_{k,d}$ should be dominated by the prior and expected to be near 1, in order to have little influence on $\theta_{k,i}$. Note that the variance of $h_{k,d}$ is $\frac{1}{\mu_0}$, which is a hyperparameter of our model.

One of the particularly appealing aspects of our parameterization in Eq. 8 is its computational efficiency when the features are sparse (which is usually the case with most multi-label learning datasets). In contrast, the existing label embedding models [Yu et al., 2014, Rai et al., 2015, Mineiro and Karampatziakis, 2015] learn an explicit regression model from the D dimensional feature vector $\boldsymbol{x}_i$ to $\theta_{i,k}$, which is computationally very expensive for large D. At the same time, the choice of parameterization in Eq. 8 also facilitates in retaining the conjugacy of our model, leading to a simple and efficient inference algorithm. We will study the details of how the inference leverages the sparsity of the feature matrix in Section 4.

3.3 Leveraging Label Co-occurrences

In addition to the labels of the instances, it is often possible to get *label co-occurrence* statistics [Mensink et al., 2014] from an external source, such as a text corpus (e.g., Wikipedia). Suppose the label co-occurrence statistics are provided in form of an $L \times L$ count matrix $\mathbf{C} \in \mathbb{Z}^{L \times L}$, where each entry of $\mathbf{C}$ denotes the number of times a pair of labels co-occurs. Note that in the absence of an external source of information, one possible way to construct the matrix $\mathbf{C}$ could be to use the label matrix $\mathbf{Y}$ itself, i.e., as $\mathbf{C} = \mathbf{Y}^\top \mathbf{Y}$. In this case, even though $\mathbf{C}$ reuses the information already present in $\mathbf{Y}$, this "re-encoding" of information can still help the model, as also corroborated by recent work [Liang et al., 2016].

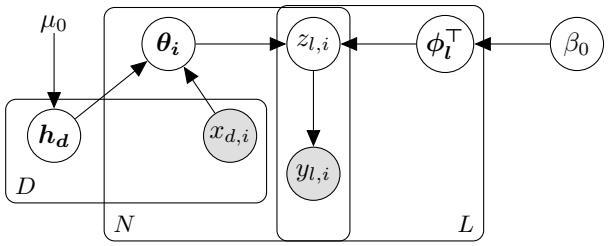

Figure 1: The graphical model for factorizing the label matrix. $\boldsymbol{h_d}$, $\boldsymbol{\theta_i}$, $\boldsymbol{\phi_l}$ is the d^{th} column of $\mathbf{H}$, the i^{th} column of $\boldsymbol{\Theta}$, the l^{th} row of $\boldsymbol{\Phi}$ respectively. All of them are K dimensional vectors.

It is natural to model label co-occurrences by the Poisson distribution:

$$c_{l,m} \sim \text{Poisson}(\psi'_{l,m}) \tag{10}$$

where $c_{l,m}$ denotes the number of times a pair of labels l and m co-occurs, $\psi'_{l,m}$ denotes the $(l,m)^{th}$ entry in the Poisson rate matrix $\boldsymbol{\Psi'} \in {\mathbb{R}_+}^{L \times L}$. We further apply a low-rank factorization of $\boldsymbol{\Psi'}$ as follows:

$$\boldsymbol{\Psi'} = \boldsymbol{\Phi}^\top \boldsymbol{\Lambda} \boldsymbol{\Phi} \tag{11}$$

Here $\boldsymbol{\Lambda} \in {\mathbb{R}_+}^{K \times K}$ is a diagonal matrix, whose diagonal elements are denoted by the vector $\boldsymbol{\lambda} \in {\mathbb{R}_+}^K$. We assume λ_k to have a gamma prior distribution:

$$\lambda_k \sim \text{Gamma}(\gamma_0/K, f_0) \tag{12}$$

where γ_0, f_0 are given uninformative gamma priors.

Figure 2 shows the graphical model of this part. Note that $\boldsymbol{\Phi}$ in Eq. (11) is the same "K topics" matrix that we have used in the low-rank modeling of the label matrix $\mathbf{Y}$ (Sec. 3.1). This is essentially a co-factorization model, such as the *collective matrix factorization* Singh and Gordon [2010], Klami et al. [2013], for joint low-rank modeling of multiple matrices with shared latent factors. In our case, these matrices are the label matrix $\mathbf{Y}$ and the label co-occurrence matrix $\mathbf{C}$, with the topic matrix $\boldsymbol{\Phi}$ shared by the latent factor models of both $\mathbf{Y}$ and $\mathbf{C}$. Note however that

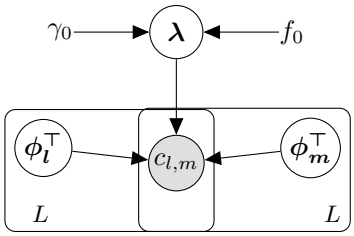

Figure 2: The graphical model for leveraging co-occurrences. $\boldsymbol{\phi_l}$ is the l^{th} row of $\boldsymbol{\Phi}$. $\boldsymbol{\phi_l}$ and $\boldsymbol{\lambda}$ are K dimensional vectors. Note: In the overall model, this part is learned jointly with the factorization of the label matrix.

unlike collective matrix factorization Singh and Gordon [2010], Klami et al. [2013], our gamma-Poisson generative model can effectively leverage the sparsity of these matrices and results in very efficient inference, with complexity that scales in the number of nonzeros.

4 Inference

Exact inference in our Bayesian model is intractable. However, one of the most appealing properties of our model is that it admits very simple yet efficient approximate inference via closed form Gibbs sampling updates. Leveraging data augmentation techniques Zhou et al. [2012], the proposed model enjoys full local conjugacy and facilitates deriving efficient Gibbs sampling updates for all the latent variables of our model. Moreover, the inference in our model scales in the number of nonzeros in both the label matrix as well as the feature matrix, which makes the model work efficiently for multi-label learning problems that involve large but highly sparse feature and label matrices.

4.1 Sampling Latent Counts $z_{l,i,k}$

Given a binary label $y_{l,i}$, according to our model construction in Eq. (2), we first need to sample the corresponding latent count $z_{l,i}$, which can be drawn from a truncated Poisson distribution:

$$(z_{l,i}|y_{l,i},\psi_{l,i}) \sim y_{l,i} \cdot \text{Poisson}_+(\psi_{l,i}) \tag{13}$$

The above equation indicates that we only need to sample $z_{l,i}$ if $y_{l,i} > 0$, i.e., the sparsity of the label matrix.

Given Eq. (2) and the additivity of Poisson, the latent count $z_{l,i}$ can be written as a sum of K smaller latent counts, each of which is contributed by the corresponding topic:

$$z_{l,i} = \sum_k^K z_{l,i,k} \tag{14}$$

$$z_{l,i,k} \sim \text{Poisson}(\phi_{k,l}\theta_{k,i}) \tag{15}$$

where $z_{l,i,k}$ is the counts for each topic k.

Moreover, using the relationship of the Poisson and multinomial distributions, we can express the decomposition in Eq. (14) and Eq. (15) as a draw from a multinomial:

$$[z_{l,i,1},\cdots,z_{l,i,K}] \sim \text{Multi}\left\{z_{l,i}; \frac{[\phi_{1,l}\theta_{1,i},\cdots,\phi_{K,l}\theta_{K,i}]}{\sum_k^K \phi_{k,l}\theta_{k,i}}\right\} \tag{16}$$

4.2 Sampling Latent Counts $c_{l,m,k}$

To infer the latent factors defining the generative model of the count-valued label co-occurrences $c_{l,m}$ (Fig. 2), we leverage a similar latent variable augmentation scheme to the one used for sampling the latent counts associated with the label matrix (cf., Section 4.1). In particular, we assume the *observed* label co-occurrence $c_{l,m}$ for two labels l and m as a sum of K smaller latent counts (each of which can be attributed to one of these K topics) as follows

$$c_{l,m} = \sum_k^K c_{l,m,k} \tag{17}$$

$$c_{l,m,k} \sim \text{Poisson}(\phi_{k,l}\lambda_k\phi_{k,m}) \tag{18}$$

where $c_{l,m,k}$ is the latent counts for topic k.

Again, given $c_{l,m}$, which is observed, $c_{l,m,k}$ can be sampled from multinomial, similar to the sampling of $z_{l,i,k}$ in Eq. (16).

4.3 Sampling $h_{k,d}$ and b_k

As $\boldsymbol{\phi_k}$ is normalized (sums to 1), summing over l of Eq. (15) and using the additivity of Poisson, we get:

$$z_{\cdot,i,k} \sim \text{Poisson}(\theta_{k,i}) \tag{19}$$

where $z_{\cdot,i,k} = \sum_l^L z_{l,i,k}$. Thus, the likelihood of θ is

$$\prod_{k,i} e^{-\theta_{k,i}}\theta_{k,i}^{z_{\cdot,i,k}} \tag{20}$$

Given Eq. (8), recall that all the features are binary and $h_{k,d}$ influences $\theta_{k,i}$ iff $x_{d,i} = 1$. This gives us a direct way of extracting $h_{k,d}$ from $\theta_{k,i}$. We can derive the likelihood of $h_{k,d}$ as:

$$e^{-h_{k,d}\sum_{i:x_{d,i}=1}^N \frac{\theta_{k,i}}{h_{k,d}}}(h_{k,d})^{\sum_i^N x_{d,i}z_{\cdot,i,k}} \tag{21}$$

which is conjugate to its Gamma prior. Therefore, it is straightforward to yield the following sampling strategy for $h_{k,d}$:

$$h_{k,d} \sim \text{Gamma}\left(\mu_0 + \sum_{i:x_{d,i}=1}^{N} z_{\cdot,i,k}, \frac{1}{\mu_0 + \sum_{i:x_{d,i}=1}^{N} \frac{g_{k,i}}{h_{k,d}}}\right) \tag{22}$$

b_k can be sampled using the same formula by adding an extra row of ones in the feature matrix $\mathbf{X}$ (which serve as the default features).

We can compute and cache the value of $\theta_{k,i}$ first. After $h_{k,d}$ is sampled, we can update $\theta_{k,i}$ for the instances where feature d is on:

$$\theta_{k,i} \leftarrow \frac{\theta_{k,i} h'_{k,d}}{h_{k,d}} \tag{23}$$

where $h'_{k,d}$ is the newly-sampled value of $h_{k,d}$.

To sample h and compute θ, according to Eq. (8) and Eq. (22), one only iterates over the instances where feature d is on (i.e., $x_{d,i} = 1$) instead of iterating over all the instances. This demonstrates how the sparsity in the feature matrix is leveraged. Note that the inference simplicity only exists with binary features.

4.4 Sampling $\boldsymbol{\phi_k}$

If the co-occurrence matrix is not incorporated, using Eq. (16) and the Dirichlet-multinomial conjugacy, $\boldsymbol{\phi_k}$ can be sampled as:

$$\boldsymbol{\phi_k} \sim \text{Dirichlet}_L(\beta_0 + z_{1,\cdot,k}, \cdots, \beta_0 + z_{L,\cdot,k}) \tag{24}$$

where $z_{l,\cdot,k} = \sum_i^N z_{l,i,k}$.

Otherwise, ϕ is also involved in the generative process of $\mathbf{C}$. According to Eq. (18), the likelihood of $\mathbf{C}$ is

$$e^{-\sum_{l,m,k} -\phi_{k,l}\lambda_k\phi_{k,m}} \prod_{l,m,k} (\phi_{k,l}\lambda_k\phi_{k,m})^{c_{l,m,k}} \tag{25}$$

Given the fact that $\boldsymbol{\phi_k}$ is normalized, the likelihood term related to $\phi_{k,l}$ is: $\phi_{l,k}^{c_{l,\cdot,k}}$ where $c_{l,\cdot,k} = \sum_m^L c_{l,m,k} + \sum_m^L c_{m,l,k}$. Therefore, we can sample $\boldsymbol{\phi_k}$ as:

$$\boldsymbol{\phi_k} \sim \text{Dirichlet}_L(\cdots, \beta_0 + z_{l,\cdot,k} + c_{l,\cdot,k}, \cdots) \tag{26}$$

4.5 Sampling λ_k

According to Eq. (25), λ_k has the Poisson likelihood, which is conjugate to its Gamma prior. Therefore, we can sample λ_k as:

$$\lambda_k \sim \text{Gamma}[\gamma_0/K + c_{\cdot,\cdot,k}, 1/(f_0+1)] \tag{27}$$

where $c_{\cdot,\cdot,k} = \sum_l^L c_{l,\cdot,k}$.

Recall that γ_0 and f_0 have uninformative Gamma prior. For γ_0, we can apply the data augmentation in Zhou et al. [2012], Buntine and Hutter [2012] to get the Gamma likelihood. For f_0, its posterior is directly conjugate to the Gamma likelihood.

4.6 Time-Complexity Analysis

In addition to having a rich generative model for the label and label co-occurrences, one of the key properties of the proposed model is the computational efficiency resulting from taking advantage of the sparsity in both feature and label matrices. This is important because in many multi-label learning problems, the feature and label matrices usually are massive but highly sparse. Specifically, for the label matrix, with the Bernoulli-Poisson link, the models scales in the number of nonzeros in the label matrix. At the same time, sampling h and computing θ scale in the number of nonzeros in the feature matrix. Therefore, in the case where the label co-occurrences are not leveraged, the inference complexity of the proposed model is $\mathcal{O}(KG + KDG')$ where G is the number of nonzeros in the label matrix $\mathbf{Y}$ and G' is the average number of instances where a feature is on (i.e., the column-wise sparsity of $\mathbf{X}$). Even when the label co-occurrences are leveraged, it does not add much overhead since the label co-occurrence matrix is usually highly sparse as well and its low-rank factorization scales in the number of nonzeros in this matrix. The efficiency of our model will be empirically studied in Section 6.4.

5 Related Work

Multi-label learning problems in modern-day applications are usually characterized by a large number of training instances, a large number of features, and a large number of labels (i.e., label-space cardinality). Owing to this, there is a considerable recent interest in designing multi-label learning models that can gracefully scale to handle such large datasets.

Label embedding methods offer an appealing solution to the large label-space cardinality problem. These methods project the high-dimensional sparse label vectors of each instance into a low-dimensional space. This corresponds to learning a low-rank embedding of the label matrix. However, learning the embedding itself is a computationally challenging problem, especially when the label matrix is massive. This has led to a lot of recent interest in embedding based models for multi-label learning that can learn label matrix embeddings efficiently [Yu et al., 2014, Mineiro and Karampatziakis, 2015]. However, most of these methods do not exploit the sparsity of the label matrix while learning the embeddings. Recently, [Rai et al., 2015]

proposed a Bayesian label matrix embedding method that scales in the number of nonzeros in the label matrix. Their approach is similar in spirit to our approach. However, the approach in [Rai et al., 2015] conditions the embeddings on the feature vectors via a regression model. Learning this regression model is challenging due to non-conjugacy, and is computationally expensive. In contrast, our approach of learning the label matrix embedding also scales in the number of nonzeros in the label matrix. However, the embeddings are conditioned on the feature vector not via a regression model used in [Rai et al., 2015] but via a log-linear combination of the features. If the features are binary and sparse, such an approach of conditioning on the features leads to significant speed-ups. In our experiments, we compare the per iteration computational cost of our approach with the approach of [Rai et al., 2015] and observe significant speed-ups. Moreover, unlike our model, the model of [Rai et al., 2015] cannot leverage label co-occurrences.

Other prominent Bayesian approaches to multi-label include the Bayesian compressed sensing (BCS) based approach [Kapoor et al., 2012]. However, inference in BCS is expensive. Moreover, it does not exploit the sparsity of label matrix or feature matrix, and is therefore not suitable for large-scale multi-label datasets.

Leveraging label co-occurrences to improve multi-label learning has not received much attention so far, except for some recent works such as [Mensink et al., 2014, Gaure et al., 2017]. One key difference of our model as compared to these models is that the computational cost scales in the number of nonzeros in the label and feature matrix. Moreover, the Poisson-Dirichlet-gamma based latent factor model offers a nice interpretability of our model, making it also suitable for other tasks, such as topic discovery (e.g., group of related labels representing a topic). In our experiments, we show such a qualitative analysis on a real dataset.

Our approach of constructing embeddings via conditioning on features is related to the models that incorporate auxiliary information in Poisson factorization or topic models such as the ones in Hu et al. [2016], Zhao et al. [2017a,b,c]. Features in those models are used to construct the prior of the embeddings. However, in our model, the embeddings are directly constructed using the features (Eq. 8), which allows efficiently computing the embeddings of test instances.

6 Experiments

In our experiments, we compare the proposed **B**ayesian **M**ulti-label **L**earning with **S**parse Features and Labels (abbreviated BMLS) with various state-of-the-art multi-label learning models, which include both Bayesian and non-Bayesian models. We evaluate the proposed model on four benchmark multi-label datasets with binary features: Bibtex, Delicious, Movielens, and NIPS.

The statistics of the datasets are listed in Table 1. The datasets cover a wide range of feature and label sizes. Moreover, both the feature vectors as well as the label vectors are highly sparse, reflecting real-world multi-label learning problems. Our model can effectively exploit the sparsity in these vectors, which results in a fast inference procedure.

We compare the following models: **(1) BMLS:** Our proposed model. We experiment with two variants - with and without the label co-occurrences. If the label co-occurrences are leveraged, we refer to the model as BMLS-co. **(2) LEML:** Low rank Empirical risk minimization for Multi-label Learning Yu et al. [2014]. Similar to our model, LEML factorizes the label matrix $\mathbf{Y}$ with two matrices and one of them is further factorized with the feature matrix $\mathbf{X}$. LEML considers various types of loss functions such as squared loss, logistic loss, hinge loss, etc. **(3) BMLPL:** Bayesian Multi-label Learning via Positive Labels Rai et al. [2015]. As one of the most related models to BMLS, BMLPL applies the Bernoulli-Poisson factorization on $\mathbf{Y}$ as well. However, unlike our model, BMLPL uses a regression based approach to condition on the features. **(4) BCS:** Bayesian Compressed Sensing for multi-label learning Kapoor et al. [2012]. BCS is a Bayesian method that uses the idea of doing compressed sensing on the label vectors Hsu et al. [2009], and relies on variational inference. **(5) BNMC:** Bayesian Nonparametric model for Multi-label Classification Nguyen et al. [2016]. BNMC is a Bayesian model that automatically learns and exploit the unknown number of multi-label correlation.

We report the Area Under the ROC Curve (AUC) on the test data to measure the prediction performance on new instances for all the models being compared. In particular, for our model, we can obtain $\mathbf{H}, \boldsymbol{b}, \boldsymbol{\Phi}$ from the training phase. Given a new instance i', we can compute $\theta_{k,i'}$ by Eq. (8) using its feature vector $\boldsymbol{x}_{i'}$. The labels can be predicted as follows:

$$\Pr(y_{l,i'} = 1) = 1 - e^{-\sum_k^K \phi_{k,l}\theta_{k,i'}}$$

In the experiments, we set the hyperparameters for our model as $\mu_0 = 10$, $\beta_0 = 0.01$, $K = 100$ and γ_0, f_0 are given uninformative gamma priors. We use 5000 Gibbs sampling iterations to train the model and report the average results over the last 2500 iterations. For the baseline models, we use their default parameter settings.

Table 1: The statistics of the datasets used in the experiments. N_{train}: number of training instances, N_{test}: number of test instances, D: number of features, L: number of labels.

Dataset	N_{train}	N_{test}	D	L
Bibtex	4880	2515	1836	159
Delicious	12920	3185	500	983
Movielens	4000	2040	29	3952
NIPS	2292	573	2484	14036

Table 2: Comparison of the various methods in terms of AUC scores with all the instances in the training sets. "-" denotes either these results were not available or the method was infeasible to run on that data set.

Model	Bibtex	Delicious	Movielens	NIPS
LEML	0.9040	0.8894	**0.8787**	0.8777
BMLPL	0.9210	0.8950	0.8582	0.9002
BCS	0.8614	0.8000	-	-
BNMC	0.8318	-	-	-
BMLS	**0.9379**	**0.9062**	0.8682	**0.9009**

6.1 Results using Complete Training Set

In the first experiment, we train all the models using all the instances in the training set. The AUC scores are reported in Table 2. The result shows that the proposed model performs better than the other models in three out of four datasets, which evidences the effectiveness of our model. Note that BMLS-co performs comparably to BMLS in this setting (possibly because training data is plenty), so its results are not reported.

6.2 Results using Missing Labels and Limited Training Instances

One common problem of multi-label learning is missing labels. As a Bayesian model, the proposed model naturally handles this problem. Furthermore, it is reasonable to assume that the label co-occurrences shall play a more important role in the case of missing labels. To examine this, we randomly remove 80% entries from the label matrix in the training data of Bibtex, Delicious, and Movielens to mimic the situation where a significantly large fraction of the labels are missing. The AUC scores of this experiment are shown in Table 3. From the results, it can be observed that BMLS-co gains better results than BMLS, especially on the Bibtex dataset, demonstrating that the label co-occurrences do help in the case with missing labels. Moreover, both of our proposed models outperform the others significantly in this case. It is also noteworthy that although LEML gets better AUC score on the Movielens dataset with all the training instances, the proposed models have a clear advantage when there is a high fraction of missing labels.

Table 3: AUC scores with only 20% labels.

Model	Bibtex	Delicious	Movielens
LEML	0.8452	-	0.8406
BMLPL	0.7879	0.8082	0.8574
BMLS	0.8598	0.8933	0.8619
BMLS-co	**0.8764**	**0.8978**	**0.8643**

Table 4: AUC scores with only 20% instances of the training set.

Model	Bibtex	Delicious	Movielens
LEML	0.8649	0.7325	0.8429
BMLPL	0.8167	0.8484	0.8437
BNMC	0.7549	-	-
BMLS	0.8651	0.8888	**0.8629**
BMLS-co	**0.8723**	**0.8921**	0.8562

Another situation where the label co-occurrences may benefit is the case where there are not sufficient training examples in the data. We mimic this situation by reducing the size of training instances to 20% on Bibtex, Delicious, and Movielens. The AUC scores in this case in shown in Table 4. Here we can observe a similar trend as for the missing label case: BMLS has significantly better performance as compared to the baseline models and BMLS-co further improves the prediction accuracies using the label co-occurrences.

6.3 Qualitative Analysis: Topic Modeling on NIPS Dataset

Recall that in our model, $\boldsymbol{\phi}_k$ represents a distribution (i.e., a "topic") over the labels. To assess our model's ability to discover meaningful topics, we run an experiment on the NIPS dataset with $K = 100$ and examine each topic. The NIPS dataset consists of 14036 labels (each of which is a word; each author (i.e., instance) has a subset of words), so $\boldsymbol{\phi}_k$ is of that size. In Table 5, we show five of the topics with their top words (ranked by $\phi_{k,l}$) and the top authors (ranked by $\theta_{k,i}$). As shown in the table, our model is able to discover clear and meaningful topics of the authors, which shows its usefulness as a topic model when each document $\boldsymbol{y}_i \in \{0,1\}^L$ has features in form of meta data $\boldsymbol{x}_i \in \{0,1\}^D$ associated with it.

Table 5: The top words and authors with the largest weights in the topics.

Topic:	1	2	3	4	5
Top words:	input	problem	image	posterior	optimal
	neural	theorem	dimensional	distributions	control
	networks	theory	system	log	current
	network	bound	vision	likelihood	actions
	training	result	images	monte	dynamic
	set	exists	visual	inference	programming
	learning	positive	object	bayesian	learn
	output	dimension	computer	joint	action
	weights	proof	pattern	carlo	state
	information	assume	position	variance	machine
Top authors:	Mozer_M	Sontag_E	Sejnowski_T	Jordan_M	Sejnowski_T
	Hinton_G	Venkatesh_S	Hinton_G	DeFreitas_J	Dayan_P
	Sejnowski_T	Bartlett_P	Baluja_S	Hinton_G	Hinton_G
	Bengio_Y	Jordan_M	Zemel_R	Doucet_A	Mozer_M
	Giles_C	Meir_R	Poggio_T	Bishop_C	Jordan_M

6.4 Running Time

In this section, we empirically compare the running time of our model with BMLPL[2], with a similar low-rank embedding approach. Note that BMLPL uses a regression approach to condition the embeddings on the features, while in our model, the embeddings are conditioned on the features via a log-linear combination of the features. This makes our model much more scalable, while also enjoying closed form, highly efficient Gibbs sampling.

Both the models are implemented in MATLAB running on a desktop with 3.40 GHz CPU and 16GB RAM. We report the running time per MCMC iteration on the four datasets and we also vary the size of training instances from 20% to 80% to fully exam the efficiency. Shown in Table 6, the proposed model runs much faster than BMLPL, supporting the time-complexity analysis in Section 4.6.

Table 6: Running time per iteration (seconds) of BMLS and BMLPL. $K = 100$ for both models.

Dataset	% training	BMLPL	BMLS
Bibtex	20%	18.14	0.04
	40%	22.54	0.06
	60%	26.75	0.09
	80%	29.80	0.11
Delicious	20%	12.18	0.09
	40%	14.45	0.16
	60%	17.82	0.24
	80%	20.70	0.33
Movielens	20%	19.19	0.16
	40%	21.86	0.27
	60%	24.08	0.37
	80%	26.27	0.49
NIPS	20%	35.50	0.66
	40%	38.51	1.10
	60%	40.31	1.55
	80%	43.06	2.01

7 Conclusion and Discussion

Despite the considerable amount of recent progress on the problem of multi-label learning, Bayesian approaches to this problem have received relatively little attention. This is primarily due to the lack of scalable approaches that can handle large datasets and can be efficient at training and test time. With this motivation, in this paper, we presented a framework for multi-label learning that leverages some of the key characteristics of multi-label learning datasets (in particular, the sparsity of label and feature matrix) to design a scalable Bayesian multi-label learning model. Unlike most existing multi-label learning models that are based on learning a low-rank factorization of the label matrix, our model performs a joint factorization of the label matrix and the label co-occurrence matrix and, by sharing latent factors between the two factorizations, it can address problems such as lack of training data and/or a high fraction of missing labels in the label matrix. The topic-based interpretation of our label embedding approach is intuitive and we hope it would motivate the application of similar topic model based approaches for the problem of multi-label learning. Finally, making such models more scalable would be an interesting direction of future work. Although in this paper, we have presented Gibbs sampling for doing inference in the model, developing variational inference or stochastic variational inference would further improve the scalability of our model.

[2]We only compare the running time with BMLPL because (1) it is a Bayesian model with the similar base framework like ours, (2) its inference is done by Gibbs sampling and implemented in MATLAB as well.

Acknowledgements

PR acknowledges support from IBM Faculty Award, DST-SERB Early Career Research Award, and Dr. Deep Singh and Daljeet Kaur Faculty Fellowship, IIT Kanpur.

References

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

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

W. Buntine and M. Hutter. A Bayesian view of the Poisson-Dirichlet process. *arXiv preprint arXiv:1007.0296v2 [math.ST]*, 2012.

A. Gaure, A. Gupta, V. K. Verma, and P. Rai. A probabilistic framework for zero-shot multi-label learning. In *UAI*, 2017.

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

D. J. Hsu, S. M. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009.

C. Hu, P. Rai, and L. Carin. Non-negative matrix factorization for discrete data with hierarchical side-information. In *19th International Conference on Artificial Intelligence and Statistics*, pages 1124–1132, 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.

A. Klami, G. Bouchard, and A. Tripathi. Group-sparse embeddings in collective matrix factorization. *CoRR*, abs/1312.5921, 2013.

D. Liang, J. Altosaar, L. Charlin, and D. M. Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *RecSys*, 2016.

T. Mensink, E. Gavves, and C. G. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014.

P. Mineiro and N. Karampatziakis. Fast label embeddings via randomized linear algebra. In *ECML*, 2015.

V. Nguyen, S. Gupta, S. Rana, C. Li, and S. Venkatesh. A Bayesian nonparametric approach for multi-label classification. In *ACML*, 2016.

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 *NIPS*, 2015.

A. P. Singh and G. J. Gordon. A Bayesian matrix factorization model for relational data. In *UAI*, 2010.

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

H. Zhao, L. Du, and W. Buntine. Leveraging node attributes for incomplete relational data. In *ICML*, 2017a.

H. Zhao, L. Du, and W. Buntine. A word embeddings informed focused topic model. In *ACML*, 2017b.

H. Zhao, L. Du, W. Buntine, and G. Liu. MetaLDA: A topic model that efficiently incorporates meta information. In *ICDM*, 2017c.

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

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