JMLR: Workshop and Conference Proceedings 63:254–269, 2016 ACML 2016

A Bayesian Nonparametric Approach for Multi-label Classification

Vu Nguyen V.NGUYEN@DEAKIN.EDU.AU
Sunil Gupta SUNIL.GUPTA@DEAKIN.EDU.AU
Santu Rana SANTU.RANA@DEAKIN.EDU.AU
Cheng Li CHENG.L@DEAKIN.EDU.AU
Svetha Venkatesh SVETHA.VENKATESH@DEAKIN.EDU.AU
Center for Pattern Recognition and Data Analytics, Deakin University

Editors: Robert J. Durrant and Kee-Eung Kim

Abstract

Many real-world applications require multi-label classification where multiple target labels are assigned to each instance. In multi-label classification, there exist the intrinsic correlations between the labels and features. These correlations are beneficial for multi-label classification task since they reflect the coexistence of the input and output spaces that can be exploited for prediction. Traditional classification methods have attempted to reveal these correlations in different ways. However, existing methods demand expensive computation complexity for finding such correlation structures. Furthermore, these approaches can not identify the suitable number of label-feature correlation patterns. In this paper, we propose a Bayesian nonparametric (BNP) framework for multi-label classification that can automatically learn and exploit the unknown number of multi-label correlation. We utilize the recent techniques in stochastic inference to derive the cheap (but efficient) posterior inference algorithm for the model. In addition, our model can naturally exploit the useful information from missing label samples. Furthermore, we extend the model to update parameters in an online fashion that highlights the flexibility of our model against the existing approaches. We compare our method with the state-of-the-art multi-label classification algorithms on real-world datasets using both complete and missing label settings. Our model achieves better classification accuracy while our running time is consistently much faster than the baselines in an order of magnitude.

1. Introduction

Although many supervised learning methods assume only one associated outcome with the input data, several real-world applications have to deal with multiple outcomes. Medical data is an important example of this problem - for instance, a cancer patient when being treated with chemotherapy or radiotherapy may have multiple toxicity outcomes (labels). Prediction of such adverse events before treatment offers opportunity to alter treatment to mitigate such adverse effects.

A straightforward solution to multi-label learning is to decompose the problem into a series of binary classification problems, each for one label. However, the key challenge of learning these binary decomposition is the overwhelming size of output space, i.e. the number of label sets grows exponentially as the number of class labels increases. For example, given

© 2016 V. Nguyen, S. Gupta, S. Rana, C. Li & S. Venkatesh.

a label space with 30 class labels, the number of possible label sets would exceed one billion (i.e. 2^{30}) (Zhang and Zhou, 2014). Such a solution, nevertheless, neglects the fact that information of one label may be helpful for the learning of another related label; especially when some labels have insufficient training examples, the label correlations may provide useful extra information.

To exploit label correlations, external knowledge such as existing label hierarchies can be used (Cai and Hofmann, 2004). However, such external information may be hard to obtain. Many other approaches try to learn label correlations hidden in the training data. Hidden label correlations are exploited either locally (Huang and Zhou, 2012) or by constructing label-specific features using positive and negative instances (Zhang and Wu, 2015). A better approach is to exploit both label and feature correlations - one example is to exhaustively encode the conditional dependencies of the label and feature set through a Bayesian network (Zhang and Zhang, 2010). Other approaches utilize different versions of the classifier chains (Read et al., 2009; Cheng et al., 2010) for learning the feature-label correlation. However, the exhaustive encoding and classifier chain come at computational cost. Moreover, it is challenging in choosing the appropriate number of label-feature patterns. We consider it as the *first challenge* of multi-label classification.

The majority of multi-label classification work (Cheng et al., 2010; Zhang and Zhang, 2010; Huang and Zhou, 2012; Zhang and Wu, 2015) has focused on the supervised settings whose assumption is that a large amount of labeled training data is available. Unfortunately, labeling training example is expensive and time-consuming, especially when it has more than one label. However, abundant unlabeled data is easy to obtain in many cases. For example, unlabeled data are enormously available in electronic medical records while a significant manual effort is required to label them. Therefore, we concern handling missing labels as the *second challenge*.

Very often, we need to perform supervised learning task when the data come in sequence, without revisiting past data. In particular, we consider the task of diagnosing patients to multiple cancers based on the historical data of other patients. These patients come daily and constantly. Retraining predictive models on a daily basis may not be feasible. The promising way to deal with this problem is using online learning. Online learning algorithms (Borodin and El-Yaniv, 1998; Rosenblatt, 1958) allow updating classifiers with new examples, without retraining the whole data. As more and more data points are added into the training set, the multi-label model will be updated accordingly. Proposing the algorithm which can perform online multi-label classification becomes our *third challenge*.

To address three challenges described above in a unified framework, we propose the Bayesian Nonparametric Multi-label Classification (BNMC) model that jointly learns the latent spaces of label-feature and estimates a classifier for each label. Our goal is to find a subset of labels and features that are strongly correlated. Especially, the number of these subsets are unknown in advance. BNMC offers the following points to solve the multi-label challenges. (1) The model jointly estimates the unknown number of latent distribution of label and feature correlations and thus solves the model selection problem. (2) As a by-product of Bayesian setting, it can handle the uncertainty of missing labels appropriately. (3) The model parameters can be updated in an online fashion using stochastic variational inference and stochastic gradient descent. We demonstrate extensive multi-label classification experiments using different settings: batch setting, learning with unlabeled data and

online setting. Our BNMC achieves superior performance in terms of accuracy and speed than the state-of-the-art multi-label classification approaches.

2. Variational Inference and Stochastic Variational Inference

We first briefly describe the variational inference, then present the stochastic variational inference (SVI) for scalable posterior estimation which is later used in the proposed model. Let us split the hidden parameters in our Bayesian model into a global parameter β (shared across all observations) and groups of local parameter $z_{1,2,\ldots,N}$ (each of which is associated to a small group of observations). Variational inference (Blei and Jordan, 2006; Wainwright and Jordan, 2008) turns the posterior inference problem into an optimization problem where a new distribution over the hidden variables $q(z, \beta)$ (called the variational distribution) is introduced. The variational distribution is a function of a set of free parameters that are optimized such that the variational distribution is as close as possible to the actual target posterior distribution where closeness is measured in terms of Kullback–Leibler (KL) divergence. Minimizing the KL divergence between the variational distribution and the target posterior is equivalent to maximizing the evidence lower bound (ELBO) that is

$$\log p(\boldsymbol{x}) = \log\left(\mathop{\mathbb{E}}_{q}\left[\frac{p(\boldsymbol{x}, \boldsymbol{z}, \beta)}{q(\boldsymbol{z}, \beta)}\right]\right) \geq \mathop{\mathbb{E}}_{q}\left[\log p(\boldsymbol{x}, \boldsymbol{z}, \beta)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(\boldsymbol{z}, \beta)\right] \triangleq \mathcal{L}(q) \tag{1}$$

where $\boldsymbol{x}$ is a collection of observations. In addition, each hidden variable is governed by its own variational parameter, e.g., $\tilde{\beta}$ governs for global variable β and $\tilde{z}_i$ governs for z_i. The variational distribution has the property that it can be efficiently computed by making each hidden variable independent of each other: $q(\Theta) = q\left(\beta \mid \tilde{\beta}\right) \prod_{i=1}^{N} q(z_i \mid \tilde{z}_i)$ where $q\left(\beta \mid \tilde{\beta}\right)$ and $q(z_i \mid \tilde{z}_i)$ take the same form as the complete conditionals $p(\beta \mid \boldsymbol{x}, \boldsymbol{z}, \alpha)$ and $p(z_i \mid \boldsymbol{x}, \boldsymbol{z}_{-i}, \beta)$, but the parameters are now $\tilde{\beta}$ and $\tilde{z}_i$.

We maximize the ELBO objective function in Eq. (1) with a coordinate ascent procedure. We find its gradient with respect to the global variational parameter $\tilde{\beta}$ and find its value that sets the gradient to zero. We do the same thing for the local parameters $\tilde{z}$. We iterate between these updates until we converge to the maximum of the ELBO. The general procedure is to write the ELBO in terms of parameter of interest (either $\tilde{\beta}$ or $\tilde{z}_i$) then take the gradient and set it to zero.

$$\tilde{\beta} = \mathop{\mathbb{E}}_{q}\left[\eta_g(x, z, \alpha)\right] \qquad\qquad \tilde{z}_i = \mathop{\mathbb{E}}_{q}\left[\eta_l(x, z_{-i}, \beta)\right]$$

Therefore, the updates of each variational parameter are equal to the expected value of the natural parameters of the complete conditionals (η_g and η_l).

Different from variational inference, stochastic variational inference (SVI) (Hoffman et al., 2013) uses a stochastic optimization technique to sequentially maximize the ELBO using unbiased samples from the data set. Instead of updating for the whole batch $\tilde{\beta} = \mathop{\mathbb{E}}_{q}\left[\eta_g(\boldsymbol{x}, \boldsymbol{z}, \alpha)\right]$, the SVI updates are performed with the following formula

$$\tilde{\beta}^{(t)} = \tilde{\beta}^{(t-1)} + \rho_t \nabla_t\left(\tilde{\beta}^{(t-1)}\right)$$

where $\nabla_t\left(\tilde{\beta}^{(t-1)}\right)$ is a noisy gradient of the objective function obtained from a subsample of the entire data and ρ_t is the learning rate. Since the objective function is not convex, it is guaranteed to converge to only local optima.

We are now going to detail how to update the variational parameters using SVI (Hoffman et al., 2013). First we write the ELBO in terms of a global term and a sum of local terms

$$\mathcal{L}(\tilde{\beta}) = \mathop{\mathbb{E}}_{q}\left[\log p\left(\beta\right)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(\beta)\right] + \sum_{i=1}^{N} \max_{\tilde{z}_i}\left(\mathop{\mathbb{E}}_{q}\left[\log p\left(x_i, z_i \mid \beta\right)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(z_i)\right]\right)$$

We consider a randomly chosen data point index I sampled from Uniform $(1, ..., N)$. For this data point x_I let us define

$$\mathcal{L}_I(\tilde{\beta}) = \mathop{\mathbb{E}}_{q}\left[\log p(\beta)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(\beta)\right] + N \max_{\tilde{z}_i}(\mathop{\mathbb{E}}_{q}\left[\log p(x_I, z_I \mid \beta)\right] - \mathop{\mathbb{E}}_{q}\left[\log q(z_I)\right])$$

This is equivalent to the original ELBO if the entire data set was made up of x_I. There are two important facts that one must understand about $\mathcal{L}_I(\tilde{\beta})$. The expectation of $\mathcal{L}_I(\tilde{\beta})$ with respect to the data point x_I is equivalent to the original ELBO. As a consequence, the gradient of $\mathcal{L}_I(\tilde{\beta})$ can be thought of as a noisy gradient of the original ELBO $\mathcal{L}(\tilde{\beta})$ because it is unbiased. The usual gradient assumes that the parameter space is Euclidean but it turns out that it is better to assume that it has a Riemannian metric structure (in the context of minimizing KL divergence) which is what the natural gradient (Amari, 1998) does. Thus, we take the natural gradient of $\mathcal{L}_I(\tilde{\beta})$. The natural gradient of $\mathcal{L}_I(\tilde{\beta})$ is

$$\nabla\mathcal{L}_I(\tilde{\beta}) = \mathop{\mathbb{E}}_{q}\left[\eta_g\left(x_I^{(N)}, z_I^{(N)}, \alpha\right)\right] - \tilde{\beta}$$

where $x_I^{(N)}, z_I^{(N)}$ are a data set formed by N replicates of observation x_I and hidden variable z_I. We set the above gradient to zero giving the update

$$\hat{\beta} \triangleq \mathop{\mathbb{E}}_{q}\left[\eta_g\left(x_I^{(N)}, z_I^{(N)}, \alpha\right)\right] = N \times \mathop{\mathbb{E}}_{q}\left[\eta_g\left(x_I, z_I, \alpha\right)\right].$$

where $\hat{\beta}$ is the intermediate global parameter of $\tilde{\beta}$. Then, we update the current estimate of the global variational parameters $\tilde{\beta}^{(t)} = (1 - \rho_t)\,\tilde{\beta}^{(t-1)} + \rho_t\hat{\beta}$. This process is repeated until the algorithm is converged or reaching maximum number of iterations.

3. Bayesian Nonparametric Multi-label Classification

We present the Bayesian Nonparametric Multi-label Classification (BNMC). We first motivate our approach. Then, we introduce our model and posterior inference in batch and online settings.

3.1. Motivation

We observe that inferring the hidden label-feature correlation in the data is not trivial. The number of correlation patterns (detailed in Sec. 3.2) is unknown and changing with the growing data. Bayesian nonparametric approaches have received increasing attention

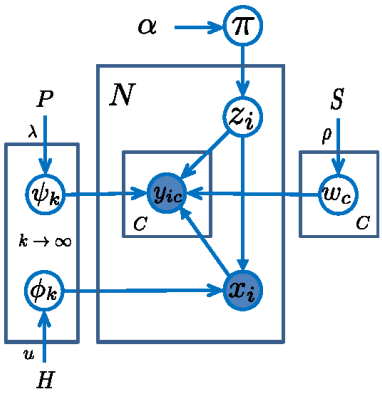

Figure 1: BNMC graphical model.

Figure 2: Generative process for BNMC

1: $\pi \sim \text{Stick}(\alpha)$
2: $\phi_k \overset{\text{iid}}{\sim} H(u),\ \forall k = 1, ..., \infty$
3: $\psi_{k,c} \overset{\text{iid}}{\sim} \text{Mult}(\lambda)\ \ \forall k, \forall c = 1...C$
4: $w_c \overset{\text{iid}}{\sim} S(\rho)\ \ \forall c = 1...C$
5: **for** each data point $i = 1, ..., N$ **do**
6: $\quad z_i \overset{\text{iid}}{\sim} \pi$ and $\boldsymbol{x}_i \sim F(\phi_{z_i})$
7: $\quad T_i = \mathcal{N}(\bar{Y}, \sigma_{\bar{Y}})$
8: $\quad \boldsymbol{y}_{i,1....C} \sim \text{Mult}\left(\psi_{z_i,1....C} \times \sigma\left(\boldsymbol{x}_i^T w_{1...C}\right), T_i\right)$
9: **end for**

recently due to their capability to perform automatic model selection (Orbanz and Teh, 2010; Nguyen et al., 2014). Nevertheless, to the best of our knowledge, there is no previous work attempting to solve the multi-label classification using BNP due to computational burden. To overcome the computational issue for BNP models, stochastic variational inference (SVI) (Hoffman et al., 2013) is introduced to approximate the posterior distribution in an online setting using stochastic optimization. In this paper, we use the idea of SVI to develop the scalable inference for learning label-feature correlation. In addition, as a Bayesian model, our proposed method allows handling the uncertainty of missing label data.

3.2. Model

We have a set of N data points $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$, where $\boldsymbol{x}_i \in \mathcal{R}^D$ is the feature and $\boldsymbol{y}_i \in (0,1)^C$ is the label. We assume that an observed feature-label pair $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$ is drawn from a pair of latent parameters $\{\phi_k, \psi_k\}$, which represents label-feature correlation. This correlation indicates that if we observe the feature vector $\boldsymbol{x}_j \sim \phi_k$, we are likely to know the label $\hat{\boldsymbol{y}}_j$ (by the corresponding pattern ψ_k). Taking an example in the healthcare domain, from the training data, we learn that the symptoms $\{a, b, c\}$ often come with the diseases $\{u, v\}$. Then, in a testing set, if we know that a new patient with symptoms $\{a, b, c\}$, we infer that he will have diseases $\{u, v\}$ with high probability.

Since the underlying label-feature correlation is not observed, we observe the raw label and feature instead. The key idea is to learn the lower dimensional representation such as $\{\phi_k, \psi_k\}_{k=1}^{K\to\infty}$ which capture the inter-dependencies of the features and labels. However, the number K of these patterns are unknown and may be changing over time.

Using Dirichlet process (Ferguson, 1973) as a nonparametric prior for the unbounded space of label-feature correlations, we describe the graphical representation in Fig. 1. Then, we present the generative process in Alg. 2 indicates how the latent parameters and observations in our model are generated. There are two key ingredients to characterize our model. The first one is the label-feature correlation $\{\phi_k, \psi_k\}$ where the number of K is inferred by the Bayesian nonparametric setting. The second one is the classifier w_c (such as Bayesian Logistic Regression (BLR) and Bayesian Support Vector Machine (BSVM) (Polson et al., 2011, 2013; Nguyen et al., 2015)) to discriminate for each class, the number of classes C is

known and fixed. In this paper, we have used the classifier as either BSVM or BLR, other classifiers under Bayesian setting could also be used.

The label vectors are assumed to follow Multinomial distribution as the product of these two views, $\boldsymbol{y}_{i,1....C} \sim \text{Mult}\left(\psi_{z_i,1....C} \times \sigma\left(\boldsymbol{x}_i^T w_{1...C}\right), T_i\right)$. The number of trials in Multinomial distribution is defined as $T_i = \mathcal{N}\left(\bar{Y}, \sigma_{\bar{Y}}\right)$ where $\bar{Y}$ is the average number of labels per data point and $\sigma_{\bar{Y}}$ is the standard deviation. These statistics ($\bar{Y}$ and $\sigma_{\bar{Y}}$) are obtained from the training set. Generating labels using Multinomial distribution includes the following benefits. The number of labels (per data point) is bounded by the mean $\bar{Y}$ and standard deviation of the number of labels $\sigma_{\bar{Y}}$ observed in the training set (i.e. ranging from $\bar{Y} - \sigma_{\bar{Y}}$ to $\bar{Y} + \sigma_{\bar{Y}}$). In addition, the predicted labels can be concentrated on the most certain label with highest probability by the probability simplex ν later defined in Section 3.6.

3.3. Posterior Inference

Using the idea of SVI, we derive stochastic variational inference for our model given the ELBO $\mathcal{L}(q)$ defined in Eq. (1). Our model parameters, defined in the previous section, include $\Theta = \{\boldsymbol{z}, \mathbf{w}, \boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\pi}\}$ where $\boldsymbol{z}$ is the local parameter and the others are the global parameters. We note that the observations include both the feature $\boldsymbol{x}$ and the label $\boldsymbol{y}$.

Using SVI, we learn the variational parameters in our model as follows. We update $\tilde{z}_i, \tilde{\phi}, \tilde{\psi}, \tilde{\pi}$ in a standard SVI form while updating $\tilde{w}$ is more complicated since it is not in an exponential family distribution. Hence, we develop two schemes to compute $\tilde{w}$. Firstly, we estimate the local conditional distribution of $\tilde{\eta}$ using augmented Gibbs approach (Polson et al., 2011, 2013). We note that using Gibbs sampling within SVI to maintain certain posterior dependencies is extremely effective (Shah et al., 2015; Hoffman and Blei, 2015). However, iteratively sampling $\tilde{w}$ will demolish the online nature of SVI. Therefore, we propose an alternative technique to compute $\tilde{w}$ using Stochastic Gradient Descent by a noise gradient vector evaluated at local data point.

Figure 3: BNMC algorithm.

Input $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{N^{\text{Train}}}$ and $(\boldsymbol{x}_j)_{j=1}^{N^{\text{Test}}}$
for $i = 1, 2, \ldots N^{\text{Train}}$ **do**
  Estimate $\tilde{z}_i^k$ using Eq. (2)
  Estimate $\tilde{\phi}_k^{(i)}$ using Eq. (3)
  Estimate $\tilde{\psi}_k^{(i)}$ using Eq. (4)
  Estimate $\tilde{\pi}_k^{(i)}$ using Eq. (5)
  Estimate $\tilde{w}_c^{(i)}$ using Eq.(7)
end for
for $j = 1, 2, ...N^{\text{Test}}$ **do**
  Compute ν_{jc} using Eq. (8)
  $T_j = \mathcal{N}\left(\bar{Y}, \sigma_{\bar{Y}}\right)$
  Predict $\hat{y}_{j,1..C}$ using Eq. (9)
end for
Output: $(\boldsymbol{y}_j)_{j=1}^{N^{\text{Test}}}$

With a slight abuse of notation, we denote the parameters as follows: ϕ_k is the variable in the original distribution. $\tilde{\phi}_k$ is the parameter in variational distribution $q\left(\phi_k \mid \tilde{\phi}_k\right)$. $\hat{\phi}_k$ is the natural gradient to update $\tilde{\phi}_k$. Other variables are used in similar notations. We summarize the posterior inference below. We refer to the **supplementary material** for detailed derivations.

Estimating $\tilde{z}_i$ The Multinomial conditional distribution for z_i^k in the original distribution p is defined as $p\left(z_i^k \mid .\right) \propto \exp\left\{\log \pi_k + \log p\left(\boldsymbol{x}_i \mid \phi_k\right) + \log p\left(\boldsymbol{y}_i \mid \psi_k\right)\right\}$. The variational distribution for z_i is $q\left(z_i \mid \tilde{z}_i\right) = \text{Mult}\left(\tilde{z}_i\right)$. The local variational parameter is set equal to

the expected natural parameter of its complete conditional distribution, that is

$$\tilde{z}_i^k = \mathop{\mathbb{E}}_{q}\left[\eta_l\left(\boldsymbol{x}_i, \boldsymbol{y}_i, \phi, \psi, \pi\right)\right] = \exp\left\{\mathbb{E}\left[\log \pi_k\right] + \mathbb{E}\left[\log \phi_{k,\boldsymbol{x}_i}\right] + \mathbb{E}\left[\log \psi_{k,\boldsymbol{y}_i}\right]\right\}. \tag{2}$$

We utilize the property of the exponential family, Dirichlet distribution in our case, to compute these expectations (see the supplement for details). Explicitly, we have that $\mathbb{E}\left[\log \phi_{k,\boldsymbol{x}_i}\right] = \sum_{d=1}^{D} x_{id}\left[\Psi(\tilde{\phi}_{k,d}) - \Psi(\tilde{\phi}_{k,*})\right]$, $\mathbb{E}\left[\log \psi_{k,\boldsymbol{y}_i}\right] = \sum_{c=1}^{C} y_{ic}\left[\Psi(\tilde{\psi}_{k,c}) - \Psi(\tilde{\psi}_{k,*})\right]$ and $\mathbb{E}\left[\log \pi_k\right] = \Psi\left(\tilde{\pi}_k\right) - \Psi\left(\sum \tilde{\pi}_*\right)$ where $*$ denotes for the sum and Ψ is the first derivative of the log Gamma function.

Estimating $\tilde{\phi}_k$ This is a global variable that its complete conditional depends on the feature $\boldsymbol{x}$ and latent assignments $\boldsymbol{z}$. The conditional distribution for the topics are defined as $p\left(\phi_k \mid \boldsymbol{z}, \boldsymbol{x}, H\right) \propto \mathrm{Dir}\left(\omega_\phi + \sum_{i=1}^{N} z_i^k x_i\right)$. The variational distribution for each topic is a D-dimensional Dirichlet $q\left(\phi_k\right) = \mathrm{Dir}\left(\tilde{\phi}_k\right)$. Then, the natural gradient is computed as $\hat{\phi}_k = \omega_\phi + N\tilde{z}_i^k \boldsymbol{x}_i$. Finally, the variational parameter $\tilde{\phi}_k$ is updated as

$$\tilde{\phi}_k^{(i+1)} = \left(1 - \rho_i\right)\tilde{\phi}_k^{(i)} + \rho_i \hat{\phi}_k \tag{3}$$

where ρ_i is the learning rate.

Estimating $\tilde{\psi}_k$ We estimate $\tilde{\psi}_k$ similar to the case of $\tilde{\phi}_k$. First we compute the natural gradient $\hat{\psi}_k = \omega_\psi + N\tilde{z}_i^k \boldsymbol{y}_i$ and update $\tilde{\psi}_k$ as

$$\tilde{\psi}_k^{(i+1)} = \left(1 - \rho_i\right)\tilde{\psi}_k^{(i)} + \rho_i \hat{\psi}_k. \tag{4}$$

Estimating $\tilde{\pi}$ The full conditional for the proportions follows a standard stick-breaking construction $p\left(\pi_k \mid \alpha, z\right) = \mathrm{Beta}\left(1 + \sum_{i=1}^{N} z_i^k, \alpha + \sum_{i=1}^{N}\sum_{j>k} z_i^j\right)$. Then, the natural gradient (two dimensional) vector is estimated as $\hat{\pi} = \left(1 + N\mathop{\mathbb{E}}_{q}\left[\tilde{z}_i^k\right], \alpha + N\sum_{j=k+1}^{K}\mathop{\mathbb{E}}_{q}\left[\tilde{z}_i^j\right]\right)$ and the stochastic updates are given

$$\tilde{\pi}_k^{(i+1)} = \left(1 - \rho_t\right)\tilde{\pi}^{(i)} + \rho_t \tilde{\pi}. \tag{5}$$

Estimating $\mathbf{w}_c$ in batch setting We consider the local conditional distribution given by $q\left(\mathbf{w}_c \mid \tilde{w}\right) = p\left(\mathbf{w}_c \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0, \boldsymbol{x}, \boldsymbol{y}\right)$. We use MCMC samples to compute w_c using Gibbs sampler. Using Gibbs sampling within SVI to maintain certain posterior dependencies is also highlighted in recent works (Shah et al., 2015; Hoffman and Blei, 2015).

We assume the prior distribution $\mathbf{w}_c \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right)$, then the posterior distribution of the classifier $\mathbf{w}_c$ is given as

$$\begin{aligned} p\left(\mathbf{w}_c \mid\right) &\propto \mathcal{N}\left(\mathbf{w}_c \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right) \prod_{\forall i \mid y_{ic}=1} p\left(y_{ic} = 1 \mid \boldsymbol{x}_i, \mathbf{w}_c\right) \times \prod_{\forall j \mid y_{jc}=0} p\left(y_{jc} = 0 \mid \boldsymbol{x}_j, \mathbf{w}_c\right) \\ &= \mathcal{N}\left(\mathbf{w}_c \mid \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N\right). \end{aligned} \tag{6}$$

We consider $\mathbf{w}_c$ for both SVM and LR. For SVM, we have $\boldsymbol{\Sigma}_N^{-1} = \boldsymbol{\mu}_0\boldsymbol{\Sigma}_0^{-1} + \sum_{i=1}^{N}\frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{\lambda_i}$ and $\boldsymbol{\mu}_N = \boldsymbol{\Sigma}_N\left(\sum_{i=1}^{N}\frac{\lambda_i+1}{\lambda_i}\boldsymbol{x}_i\right)$. For LR case: $\boldsymbol{\Sigma}_N^{-1} = \boldsymbol{\mu}_0\boldsymbol{\Sigma}_0^{-1} + \sum_{i=1}^{N}\lambda_i\boldsymbol{x}_i\boldsymbol{x}_i^T$ and $\boldsymbol{\mu}_N = \Sigma_N\left(\sum_{i=1}^{N}\boldsymbol{x}_i\left[y_{ic} - \frac{1}{2}\right] + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0\right)$. We refer to the supplement for details.

As a part of sampling $\mathbf{w}_c$ above, we need to sample the auxiliary variable λ_i as follows:

$$\lambda_i \sim \begin{cases} \left[IG\left(\left|1-\boldsymbol{x}_i^T\mathbf{w}_c\right|^{-1},1\right)\right]^{-1} & \text{SVM} \\ PG\left(1,\boldsymbol{x}_i^T\mathbf{w}_c\right) & \text{LR} \end{cases}$$

where IG is Inverse Gaussian distribution and PG is Polya-Gamma distribution.

Estimating $\mathbf{w}_c$ in online setting For online learning, we use Stochastic Gradient Descent to estimate the approximate local conditional distribution given by $q\left(w_c \mid \tilde{w}\right)$ as

$$\mathbf{w}_c^{(i+1)} = \mathbf{w}_c^{(i)} - \frac{1}{\lambda \times i} g_i \tag{7}$$

where $\frac{1}{\lambda \times i}$ is the learning rate and g_i is the gradient w.r.t. $\mathbf{w}_c^i$ evaluated at data point i.

We summarize the learning algorithm for BNMC in Alg. 3.

3.4. Handling Missing Labels in Training Data

As a Bayesian model, the proposed framework naturally handles the case of missing labels in the training set. Given a incomplete data point $\boldsymbol{x}_i$ without $\boldsymbol{y}_i$, we will compute the latent assignment using the prior and feature information $\tilde{z}_i^k \propto \exp\left\{\mathbb{E}\left[\log \pi_k\right] + \mathbb{E}\left[\log \phi_{k,\boldsymbol{x}_i}\right]\right\}$ where these expectations are defined in Eq. (2). Then, we update $\tilde{\phi}_k = \omega_\phi + \sum_{i=1}^N \tilde{z}_i^k \boldsymbol{x}_i$ using the estimated $\tilde{z}_i^k$ and the observed feature $\boldsymbol{x}_i$. We note that the label topic $\tilde{\psi}_k$ is only updated where $\boldsymbol{y}_i$ is available. For estimating the classifier $\mathbf{w}_c$, we only use the training samples which are fully observed in both $\boldsymbol{x}$ and $\boldsymbol{y}$ to reduce the uncertainty. In BNMC, the missing labels data points are still beneficial to build a good label-feature correlation.

3.5. Model complexity

The cost of estimating the label-feature correlation (steps $2-5$ in Alg. 3) is $\mathcal{O}\left(N\left[D+C\right]\right)$ where the number of hidden topics K is assumed to be smaller than the feature size D and the label size C. The complexity of estimating the classifier $\mathbf{w}_c$ using augmented approaches (cf. Section 3) is $\mathcal{O}\left(ND^2 + CD^{2.3}\right)$ where $\mathcal{O}\left(D^{2.3}\right)$ is the complexity of solving a linear system equations for each class c in Eq. (6). When we compute $\mathbf{w}_c$ using stochastic gradient descent as in step 6 of Alg. 3, the complexity is reduced to $\mathcal{O}\left(N\left[D+C\right]\right)$.

3.6. Prediction

Given the estimated model $\Theta = \{\phi_k, \psi_k, w_c, \pi\}$ and the testing observation $\boldsymbol{x}_i^{\text{Test}}$, we predict the label $\boldsymbol{y}_i^{\text{Test}} = \left[y_{i1}^{\text{Test}}, ...y_{iC}^{\text{Test}}\right]$, assumed to follow a Multinomial distribution parameterized by $\boldsymbol{\nu}_i = [\nu_{i1}, ...\nu_{iC}]$. Concretely, the probability of each element $\nu_{ic} \triangleq p\left(y_{ic}^{\text{Test}} = 1 \mid \boldsymbol{x}_i, \Theta\right)$ is computed as:

$$\begin{aligned} \nu_{ic} &\propto \sum_{k=1}^K p\left(y_{ic}^{\text{Test}} = 1 \mid z_i = k, \boldsymbol{x}_i^{\text{Test}}, \Theta\right) \times p\left(z_i = k \mid \boldsymbol{x}_i^{\text{Test}}, \Theta\right) \\ &= \sum_{k=1}^K p\left(y_{ic}^{\text{Test}} = 1 \mid \mathbf{w}_c, \boldsymbol{x}_i^{\text{Test}}\right) \times p\left(y_{ic}^{\text{Test}} = 1 \mid \psi_{z_i}\right) \times p\left(z_i = k \mid \pi\right) \times p\left(\boldsymbol{x}_i^{\text{Test}} \mid z_i = k, \phi_k\right). \end{aligned} \tag{8}$$

Given $T_i = \mathcal{N}\left(\bar{Y}, \sigma_{\bar{Y}}\right)$ (cf. Section 3.2), we predict the labels from Multinomial distribution:

$$\hat{y}_{j,1..C} \sim \text{Mult}\left(\boldsymbol{\nu}_{j,1...C}, T_i\right). \tag{9}$$

4. Experiments

In this section, we demonstrate that the proposed BNMC embodies two major merits that is desirable in any practical useful algorithm. 1) Effectiveness and efficiency: BNMC is consistently faster and obtaining high prediction accuracy than the baselines. 2) Flexibility: BNMC is well applicable to handle the unlabeled data and for online learning.

Table 1: Dataset statistics.

Datasets	#Data	#Feat	#Label
Emotions	593	72	6
Medical	978	1449	45
Scene	2,407	294	6
Corel5k	5,000	500	373
Bibtex	7,395	2,515	159
Cancer	16,397	95	33
MediaMill	43,907	120	102

First, we conduct the multi-label classification task and compare with the state-of-the-art methods. Next, we present our model's behavior. Then, we consider learning with unlabeled data. Finally, we demonstrate the proposed model in online learning setting.

Towards the open science and repeatability of our experiments, we make available all of our **source codes** at the URL[1].

Competitors We compare the proposed method with six well-established multi-label learning algorithms including:

- Binary Relevance (BR) (Boutell et al., 2004): We use LibLinear (Fan et al., 2008) toolbox to train C independent binary classification problems.
- LIFT (Zhang and Wu, 2015): The ratio parameter is set as 0.1.
- LEAD (Zhang and Zhang, 2010): The directed acyclic graph is randomly generated as the prior structure.
- ML-LOC (Huang and Zhou, 2012): The parameters are set as default $\lambda_1 = 1, \lambda_2 = 100, \sigma = 0.1, m = 15$ as recommended (Huang and Zhou, 2012).
- ML-kNN (Zhang and Zhou, 2007): The number of nearest neighbors considered is set to the average number of label per data point $\overline{L}$ and Euclidean distance is used.
- BML-CS (Kapoor et al., 2012)[2]. The compressed rate is set default at 3, $\log \chi = -0.1$ and $\log \sigma = -0.2$.
- Probabilistic Classifier Chain (PCC) (Cheng et al., 2010): We use the Java software from the author.

The Matlab codes are downloaded from the author's website. For LEAD and ML-LOC, we use linear kernel as other kernels are costly for large scale datasets.

1. https://github.com/ntienvu/ACML2016_BNMC
2. https://github.com/yalesong/BGCS

Table 2: Multi-label classification evaluation using F1 (%) score (mean±std). The highest score per dataset is in bold and the second highest is in *italic*.

	Datasets					
Methods	Emotions	Scene	Corel5K	Bibtex	Cancer	MediaMill
LIFT	22.1±.4	5.1±.5	6.5±.2	31.1±.6	54.0±.5	-
LEAD	23.2±.2	23.5±.7	5.4±.3	32.8±.5	-	-
ML-LOC	.3±.3	30.7±.5	13.9±1.0	37.1±.8	23.9±.3	-
ML-kNN	32.0±.2	22.3±.6	2.3±.6	16.8±1.0	48.2±.3	***52.3±.3***
BML-CS	30.5±.3	19.3±.5	**26.3±.7**	33.4±.9	6.4±.4	12.8±.3
BR	14.0±6.0	28.4±.6	13.3±.2	37.4±1.0	44.3±.3	10.1±.3
PCC	***34.6±9.0***	70.21±8.0	15.87±.4	***40.9±.4***	54.5±.2	**55.5±.2**
BNMC-S	**36.0±.3**	***70.4±.5***	***21.1±.2***	**41.0±.4**	**56.2±.1**	47.4±.*2*
BNMC-L	32.0±.2	**71.5±.4**	19.8±.2	40.3±.4	***55.3±.4***	48.3±.*2*

Datasets We use 6 multi-label datasets representing different kinds of the real-world data obtained from the URL [3]. The dataset statistics are summarized in Table 1. In particular, we have collected the *Cancer dataset* from a regional hospital in Australia. This cohort consists of 2,869 patients who visited the hospital during 2000-2015 and diagnosed with toxicity. We extract admissions of patients as data points and obtain approximately 16,000 data points. The features of each data point comprise of patient-specific attributes (e.g. age, gender, cancer types, cancer stage) and treatment attributes (e.g. radiotherapy durations, chemotherapy drugs, past toxicities). The multiple labels of each data point include 32 types of toxicities. Our goal is to predict the presence of toxicities for a new admission.

The training and testing data are already available by these standard datasets, except the Cancer data where we split it into 90% training and 10% testing sets.

Evaluation It is an advantage that our model can estimate the number of labels for each data point in Eq. (9) without specifying the top k labels. Given a predicted label vector $\hat{\boldsymbol{y}}_i$ and the ground truth vector $\boldsymbol{y}_i$ where each element y_{ic} is a binary value, we compute F1 score and Exact Match. The F1 formula is given as follows $\text{F1} = \frac{1}{N}\sum_{i=1}^{N}\frac{2\sum_{c=1}^{C} y_{ic}\hat{y}_{ic}}{\sum_{c=1}^{C}\hat{y}_{ic}+\sum_{c=1}^{C} y_{ic}}$. Exact Match evaluates how many times the ground truth labels and the predicted labels are exactly matched. Exact Match score is very important in some applications, e.g., in healthcare and cancer prediction: $\text{Exact Match} = \frac{1}{N}\sum_{i=1}^{N}[\boldsymbol{y}_i = \hat{\boldsymbol{y}}_i]$.

Implementation and Parameters Setting Our implementation is done in Matlab. All experiments are run on a Windows PC with 3.40 GHz Intel i7 CPU and 24 GB RAM. We repeat the experiments 10 times, then report the mean and standard deviation. The hyperparameters used in stochastic variational inference for ϕ_k, ψ_k and π_k are initialized as $\omega_\phi = 0.1, \omega_\psi = 0.01$ and $\alpha = 1$. The learning rate for SVI is set as $\rho = 0.001$ (Hoffman et al., 2013). We use the standard learning rate for SGD as $\frac{1}{\lambda t}$ where $\lambda = \frac{32}{\#Train}$. We note that the optimal λ can be selected using cross-validation or Bayesian optimization techniques (Nguyen et al., 2016).

3. http://mulan.sourceforge.net/datasets-mlc.html

Table 3: Multi-label classification evaluation using Exact Match (%) score (mean±std).

	Datasets					
Methods	Emotions	Scene	Corel5K	Bibtex	Cancer	Media Mill
LIFT	2.9±2	1.2±.9	0.6±.02	14.8±.5	41.1±1	-
LEAD	4.4±2	20.0±.9	0.4±.02	15.6±.4	-	-
ML-LOC	1.0±2	28.2±1.2	***0.8±.02***	14.8±.6	16.8±2	-
ML-kNN	6.4±2.1	19.8±1.0	0.4±.02	6.8±.5	35.0±2	*3.2±.05*
BML-CS	3.9±1.7	11.2±.9	**1.0**±.01	4.0±.6	4.0±2	3.4±.02
BR	4.9±1.5	20.5±.9	0.6±.01	13.6±.5	20.3±3	0.1±.01
PCC	1.49±1.4	46.15±.7	0.1±.01	13.7±.5	**58.54±3**	5.3±.01
BNMC-S	**10.2±1**	**66.2±.8**	0.3±.01	***16.5***±**.3**	***45.43±1***	**5.9±.01**
BNMC-L	***9***.**0±*1***	***63.2±.7***	0.4±.01	**17.6±.3**	41.4±1	***5.9±.01***

4.1. Multi-label Classification

We next conduct experiments on the multi-label classification. We use six datasets to evaluate the proposed algorithm and compare with the baseline approaches. Due to high complexity, some methods (e.g., LIFT (Zhang and Wu, 2015) and LEAD (Zhang and Zhang, 2010)) can not run on large scale datasets (e.g., MediaMill). Therefore, we set the time limit of 50,000 seconds (or 14 hours). If the algorithms exceed this limit, we will ignore them.

We report and compare the classification performance using F1 score in Table 2 and using Exact Match score in Table 3. Our BNMC beats all of the baselines in most of the datasets, except BML-CS does better for Corel5k dataset and PCC obtains the best score in Exact Match in Cancer dataset. From our observation, ML-kNN performs relatively well and robust among the baselines. We note that the numerical evaluation using Exact Match is smaller than F1 since Exact Match is a strict criteria. Especially, Exact Match criteria is hard to achieve when the output space for matching is large.

The different effects of "using label-feature correlation" against "not using it" can be seen through the performance of BR and BNMC. Learning each class independently in BR results in poor performance and can not exploit effectively the label-feature correlations.

Running Time By running time, we mean the total time of training and testing. We compare our model with other multi-label methods, except BR because BR treats each class independently that is obviously the fastest algorithm. We present the computational results in Table 4. BNMC is absolute faster than all the baselines by orders of magnitudes. Particularly, BNMC is 10-50 times faster than LIFT, LEAD and ML-LOC while 2-5 times faster than BML-CS and ML-kNN. The reason is that the training time of other algorithms, except BML-CS, is quadratic in the number of data points N. Therefore, these algorithms are not scalable to situations where N is high.

As our model's complexity is $\mathcal{O}\left(N\left[D^2+C\right]+CD^{2.3}\right)$ which is less sensitive to the number of training instances N. Due to this complexity, BNMC (batch setting) will take longer for the datasets with high dimensional feature. To have matters concrete, in Bibtex dataset ($N = 7,395; D = 2,515$), BNMC takes 295 secs while it consumes 60 secs for MediaMill ($N = 43,907; D = 120$) which contains more number of data points, but in

Table 4: Computational time comparison. Time is recorded in seconds.

	Datasets					
Methods	Emotions	Scene	Corel5K	Bibtex	Cancer	MediaMill
LIFT	1.65	206	3192	7668	32,207	>14 h
LEAD	2.51	17	3672	22,728	>14 h	>14 h
ML-LOC	6.16	398	2730	13,955	48,494	>14 h
ML-kNN	1.53	14.8	***367***	835	967	5564
BML-CS	***0.8***	4.92	1271	1160	***98***	***477***
PCC	2.07	***3.08***	4093	***345***	137	1743
BNMC-Batch	**0.6**	**2.5**	**117**	**295**	**30**	**60**

lower dimensions. Later, we show that BNMC in online setting will overcome the curse of dimensionality and be much faster with the complexity of $\mathcal{O}(N[D+C])$.

4.2. Model Analysis

In our model, a data point i includes a pair of feature and label $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}$, assumed to draw from a pair of parameter $\{\phi_k, \psi_k\}$. Intuitively, if we observe the feature vector $\boldsymbol{x} \sim \phi_k$, we are likely to know the $\hat{\boldsymbol{y}}$ (by the corresponding pattern ψ_k). This is what we mean *label-feature correlation* in the paper. To have better understanding about this correlation, we examine the Scene dataset in which BNMC automatically identifies that there are $K = 4$ latent label-feature correlations representing by a pair of $\{\phi_k, \psi_k\}$ in Fig. 4. We manually pick two correlations (1st and 4th) in Fig. 4 where we learn that the classes $1, 2$ and 3 are strongly correlated while the class 4 always exists alone. These labels patterns ψ (Right Fig. 4) link to the feature patterns ϕ (Left Fig. 4) to form label-feature correlations. We note that the feature patterns ϕ_k look visually similar to each other because (1) the feature is more noisy and complex than the label and (2) some columns of the feature matrix have high numerical values than the other columns.

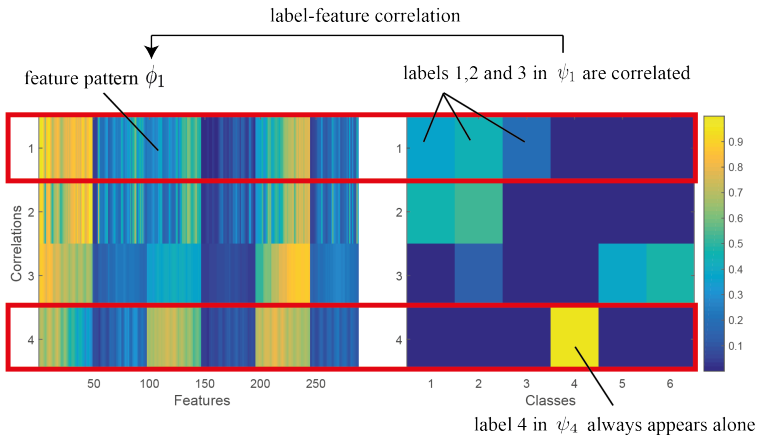

Figure 4: Label-feature correlations from Scene dataset

In addition to the label-feature correlation estimation, our BNMC also learns a set of classifiers using Support Vector Machine and Logistic Regression. Given the model parameters $\Theta = \{\phi_k, \psi_k, \eta_c\}$ and testing data point $\boldsymbol{x}$, we aim to predict the label $\boldsymbol{y} = [y_1 y_2 ... y_C] \sim \text{Mult}(\boldsymbol{\nu})$ where $\boldsymbol{\nu}$ is computed in Eq. (8). For ease of interpretation, we illustrate and compare the effects of the classifier $\mathbf{w}_c$ and the label-feature correlations to predict the final

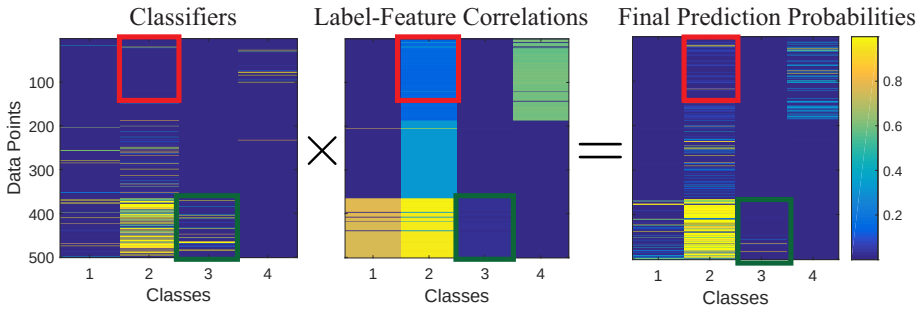

Figure 5: The final predictive probabilities of labels (Right) is a product of classifiers (Left) and label-feature correlations (Middle). The red and green boxes highlight two examples of the effects in two views to the final prediction.

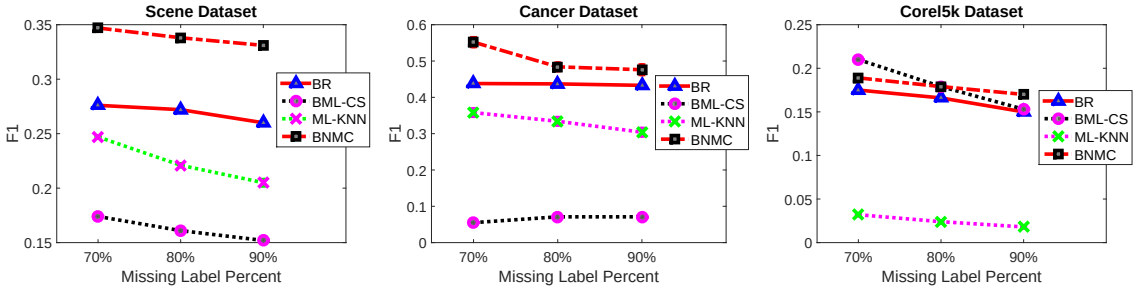

Figure 6: Multi-label classification with unlabeled data.

labels in Fig. 5 where the predictive likelihood by the classifier $\mathbf{w}_c$ is in Left Fig. 5 and the predictive likelihood by the label-feature correlation is in Middle Fig. 5. We highlight this effects in the red and green boxes. In red boxes, the classifiers (Left) give low probabilities while the label-feature correlations returns high probabilities, then the final probability will be calibrated. The reverse story can be seen for the green boxes.

4.3. Learning with Unlabeled Data

We conduct experiments for multi-label classification with unlabeled data. By unlabeled samples, we only observe the feature $\boldsymbol{x}$ while the label $\boldsymbol{y}$ is missing for these data points. We remove a fixed fraction of training labels randomly from each dataset considered. We then apply our method to such training data. BNMC can utilize feature information from the samples with missing labels to have better estimation of pattern ϕ_k. Similar setting can be achieved for missing feature, but observing labels. However, within the scope of this paper, we focus on the missing labeled case, not missing feature case.

We compare our approach with BR, ML-KNN (Zhang and Zhou, 2007) and BML-CS (Kapoor et al., 2012) for handling missing labels. While BML-CS can utilize the missing label directly, BR and ML-KNN simply remove samples with missing labels. We present the results in Fig. 6 with the percentage of labels missing ranging from 70% to 90%. It is expected that as the amount of missing labels increases, there is a smooth dip in the F1

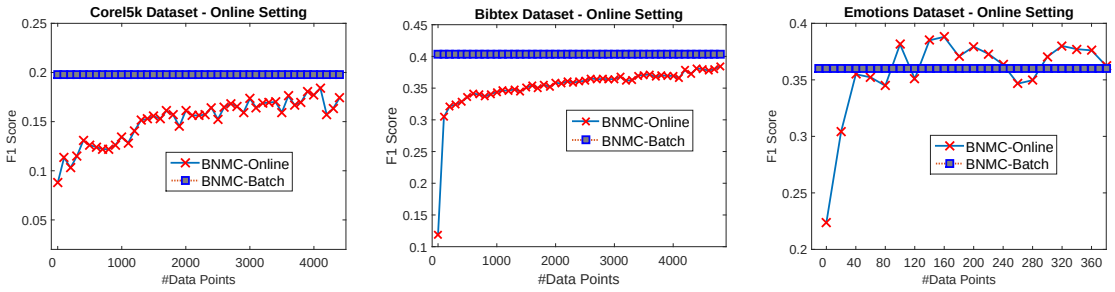

Figure 7: Multilabel classification in online setting.

score of the models. Although BML-CS obtains better performance than our model when the labels are fully observed in Corel5k dataset (cf. Table 2), when the number of missing labels increases, we equal it at 80% and surpass at 90% of missing (cf. Right Fig. 6).

4.4. Online Multi-label Classification

We evaluate the proposed BNMC in online setting of multi-label classification problem that data becomes available in a sequential order. At each step we use the new data to update our model parameters for future prediction. Although online classification (binary and multi-class setting) (Le et al., 2016) is a well-studied field, online multi-label classification is somewhat premature. We compare our BNMC-Online against our batch counterpart. BNMC-Online updates the model parameters when a data point comes in while BNMC-Batch is applied on the whole data. The more samples taken, the better our model learned (evaluated by F1 score). We plot the results in Fig. 7.

There is a trade-off in speed and accuracy in our model using online and batch setting. On the one hand, our batch version is estimated in closed-form and gains good accuracy. Its complexity of $\mathcal{O}\left(N\left[D^2+C\right]+CD^{2.3}\right)$ is affected by the feature size and slower than the BNMC-Online although it is still significantly faster than other baselines. On the other hand, BNMC-Online is approximated sequentially using the noisy gradient vectors at each data point and gets slightly worse accuracy than its batch counterpart. However, the complexity of the online algorithm is smaller at $\mathcal{O}\left(N\left[C+D\right]\right)$ and thus runs faster. We plot the running time comparison between BNMC-Online and BNMC-Batch in Fig. 8. Because of the high dimensional feature ($D = 2,515$) in Bibtex dataset, our online version highlights its superiority in running time that is 6 times faster than the batch version.

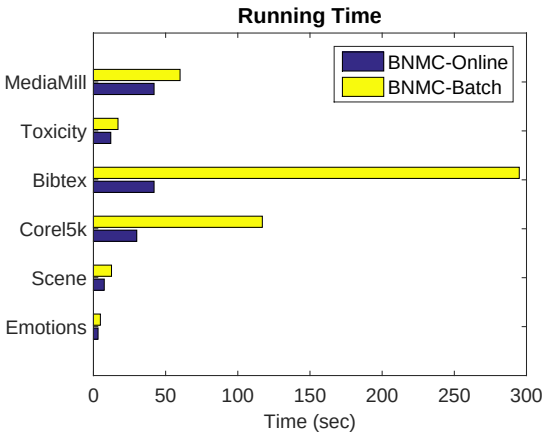

Figure 8: BNMC-Online is much faster than the BNMC-Batch counterpart.

5. Summary

We presented a BNP framework for multi-label classification that jointly learns the label-feature correlation over the low dimensional latent space. We develop an algorithm to estimate the model in batch and online settings. We carry out extensive experiments to highlight the efficacy of the proposed method. BNMC runs fast and gains high accuracy. Additionally, it can handle unlabeled data and perform online learning. BNMC is appealing for large-scale multi-label task with the ideal complexity is $\mathcal{O}(N[C+D])$.

References

S.-I. Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.

D. Blei and M. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.

A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998.

M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87. ACM, 2004.

W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

T. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973. URL http://www.jstor.org/stable/pdfplus/2958008.pdf.

M. D. Hoffman and D. M. Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, 2015.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

S.-J. Huang and Z.-H. Zhou. Multi-label learning by exploiting label correlations locally. In *AAAI*, 2012.

A. Kapoor, R. Viswanathan, and P. Jain. Multilabel classification using bayesian compressed sensing. In *Advances in Neural Information Processing Systems*, pages 2645–2653, 2012.

T. Le, V. Nguyen, T. D. Nguyen, and D. Phung. Nonparametric budgeted stochastic gradient descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 654–572, 2016.

T. Nguyen, D. Phung, S. Venkatesh, X. Nguyen, and H. Bui. Bayesian nonparametric multilevel clustering with group-level contexts. In *Proc. of International Conference on Machine Learning (ICML)*, pages 288–296, Beijing, China, 2014.

T. D. Nguyen, V. Nguyen, T. Le, and D. Phung. Sparkling vector machines. In *Workshop on Machine Learning Systems at Neural Information Processing Systems (NIPS)*, 2015.

V. Nguyen, S. Rana, S. K. Gupta, C. Li, and S. Venkatesh. Budgeted batch bayesian optimization. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, Spain, 2016.

P. Orbanz and Y. W. Teh. Bayesian nonparametric models. 2010.

N. G. Polson, S. L. Scott, et al. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1–23, 2011.

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.

J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

A. Shah, D. Knowles, and Z. Ghahramani. An empirical study of stochastic variational inference algorithms for the beta bernoulli process. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1594–1603, 2015.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

M.-L. Zhang and L. Wu. Lift: Multi-label learning with label-specific features. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):107–120, 2015.

M.-L. Zhang and K. Zhang. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 999–1008. ACM, 2010.

M.-L. Zhang and Z.-H. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 26(8):1819–1837, 2014.