Scalable Inference in Max-margin Topic Models

Jun Zhu, Xun Zheng, Li Zhou, Bo Zhang
State Key Lab of Intelligent Technology and Systems
Tsinghua National Lab for Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing, 100084, China
dcszj@mail.tsinghua.edu.cn; vforveri.zheng@gmail.com; lizhou.info@gmail.com
dcszb@mail.tsinghua.edu.cn

ABSTRACT

Topic models have played a pivotal role in analyzing large collections of complex data. Besides discovering latent semantics, supervised topic models (STMs) can make predictions on unseen test data. By marrying with advanced learning techniques, the predictive strengths of STMs have been dramatically enhanced, such as max-margin supervised topic models, state-of-the-art methods that integrate max-margin learning with topic models. Though powerful, max-margin STMs have a hard non-smooth learning problem. Existing algorithms rely on solving multiple latent SVM subproblems in an EM-type procedure, which can be too slow to be applicable to large-scale categorization tasks.

In this paper, we present a highly scalable approach to building max-margin supervised topic models. Our approach builds on three key innovations: 1) *a new formulation of Gibbs max-margin supervised topic models* for both multi-class and multi-label classification; 2) *a simple "augment-and-collapse" Gibbs sampling algorithm* without making restricting assumptions on the posterior distributions; 3) *an efficient parallel implementation* that can easily tackle data sets with hundreds of categories and millions of documents. Furthermore, our algorithm does not need to solve SVM subproblems. Though performing the two tasks of topic discovery and learning predictive models jointly, which significantly improves the classification performance, our methods have comparable scalability as the state-of-the-art parallel algorithms for the standard LDA topic models which perform the single task of topic discovery only. Finally, an open-source implementation is also provided[1].

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Statistical Computing

General Terms

Algorithms, Experimentation, Performance

Keywords

Inference, Topic Models, Large-scale Systems, Max-margin Learning

1. INTRODUCTION

Topic models such as latent Dirichlet allocation (LDA) [5] have been successful in discovering the latent factors underlying observed data. The latent topic representations can be used for many subsequent tasks, such as classification, clustering or merely as a tool to structurally browse the data. To handle large-scale applications, scalable inference algorithms [16, 19, 1] have been developed, of which the current state-of-the-art approaches can easily tackle hundreds of millions of documents and thousands of topics with hundreds of machines and thousands of CPU cores.

In many cases, we are interested in predictive tasks besides discovering latent topic representations. For example, for document data, we may be interested in predicting which categories a new document belongs to [27]; and for social network data, people have been interested in building predictive models that can suggest friends to social network users or recommend products [7, 8]. To improve the predictive ability of topic models, people have been interested in learning supervised topic models (STMs) [4, 27] which can perform the two tasks of discovering latent topic structures and learning predictive models jointly.

Max-margin STMs (e.g., maximum entropy discrimination LDA or MedLDA [27]) are the state-of-the-art methods for classification, which integrate the discriminative max-margin learning with topic models and have shown great promise in text categorization and image annotation [25, 23]. Unfortunately, the resulting learning problems normally involve a non-smooth objective, to tackle which an EM-type procedure is normally applied in the current variational or Monte Carlo solvers [12]. The EM-type algorithms need to solve many latent SVM subproblems, whose efficiency can be a bottle-neck to make these models unscalable to large scale categorization tasks, such as the PASCAL large-scale hierarchical classification challenge (LSHTC)[2] and the ImageNet large scale visual recognition challenge (ILSVRC)[3] which normally involve large data sets consisting of thousands of categories and millions of data samples.

To meet the requirements of large-scale document categorization as well as topic discovery, in this paper we present to our knowledge the first highly scalable max-margin super-

[1] http://www.ml-thu.net/∼jun/gibbs-medlda.shtml

[2] http://lshtc.iit.demokritos.gr/LSHTC2_CFP

[3] http://www.image-net.org/challenges/LSVRC/2012/index

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...$15.00.

vised topic model. Our method relies on three key innovations. *First*, unlike conventional max-margin STMs [27] that minimize a margin-loss of an expected prediction rule, we present a multi-task Gibbs max-margin STM that optimizes an expected margin loss of many latent predictive rules, each of which is randomly drawn from a posterior distribution. The method is a substantial extension of the recent work on binary classification [28] to the tasks of both single-label multi-class and multi-label [21] classification. *Second*, we present a simple collapsed Gibbs sampling algorithm without making any restricting assumptions on the posterior distributions, by exploring the classical ideas of data augmentation in statistics [20, 22] and its recent developments on learning large-margin classifiers [17]. *Third*, we present a scalable parallel implementation by leveraging the modularity property of our algorithm and the recent advances in scalable inference methods for LDA.

We apply our methods to large-scale text categorization data sets. Experimental results demonstrate significant improvements on classification performance compared to the SVM classifiers built on raw features and on the latent topic features discovered by LDA; while the time efficiency is comparable to the state-of-the-art parallel LDA [1]. In summary, our work substantially extends [28] by introducing:

- A multi-task Gibbs MedLDA with efficient sampling algorithms for handling both single-label multi-class and multi-label classification;
- A highly scalable parallel implementation for both binary and multi-task Gibbs MedLDA;
- An extensive evaluation with large-scale document categorization data sets.

Outline: We introduce the binary Gibbs MedLDA in Section 2 and present the multi-task formulation in Section 3. We present the parallel implementation in Section 4, and present the large-scale experiments in Section 5. Finally, Section 6 concludes.

2. GIBBS MEDLDA

We begin by a brief overview of the Gibbs MedLDA for binary classification.

2.1 Learning with an Expected Margin Loss

We denote the labeled training set by $\mathcal{D} = \{(\mathbf{w}_d, y_d)\}_{d=1}^D$, where the category variable Y takes values from the binary space $\mathcal{Y} = \{-1, +1\}$. Basically, a Gibbs MedLDA model consists of two parts—an LDA model for describing input documents $\mathbf{W} = \{\mathbf{w}_d\}_{d=1}^D$, where $\mathbf{w}_d = \{w_{dn}\}_{n=1}^{N_d}$ denote the words appearing in document d, and a Gibbs classifier for considering the supervising signal $\mathbf{y} = \{y_d\}_{d=1}^D$. Below, we introduce each of them in turn.

LDA: LDA is a hierarchical Bayesian model that posits each document as an admixture of K topics, where each topic $\mathbf{\Phi}_k$ is a multinomial distribution over a V-word vocabulary. For document d, the generating process can be described as

1. draw a topic proportion $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$
2. for each word n ($1 \leq n \leq N_d$):
   (a) draw a topic assignment[4] $z_{dn} \sim \mathrm{Mult}(\boldsymbol{\theta}_d)$
   (b) draw the observed word $w_{dn} \sim \mathrm{Mult}(\mathbf{\Phi}_{z_{dn}})$

where $\mathrm{Dir}(\cdot)$ is a Dirichlet distribution; $\mathrm{Mult}(\cdot)$ is multinomial; and $\mathbf{\Phi}_{z_{dn}}$ denotes the topic selected by the non-zero entry of z_{dn}. For Bayesian LDA, the topics are random samples drawn from a Dirichlet prior, $\mathbf{\Phi}_k \sim \mathrm{Dir}(\boldsymbol{\beta})$.

Given a set of documents $\mathbf{W}$, we let $\mathbf{z}_d = \{z_{dn}\}_{n=1}^{N_d}$ denote the set of topic assignments for document d and let $\mathbf{Z} = \{\mathbf{z}_d\}_{d=1}^D$ and $\mathbf{\Theta} = \{\boldsymbol{\theta}_d\}_{d=1}^D$ denote all the topic assignments and mixing proportions for the whole corpus, respectively. Then, LDA infers the posterior distribution using the Bayes' rule

$$p(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}|\mathbf{W}) = \frac{p_0(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}) p(\mathbf{W}|\mathbf{Z}, \mathbf{\Phi})}{p(\mathbf{W})},$$

where $p_0(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}) = \prod_k p_0(\Phi_k|\boldsymbol{\beta}) \prod_d p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \prod_n p(z_{dn}|\boldsymbol{\theta}_d)$ and $p(\mathbf{W}|\mathbf{Z}, \mathbf{\Phi}) = \prod_d \prod_n p(w_{dn}|z_{dn}, \mathbf{\Phi})$ according to the generating process.

An alternative way to understand Bayesian inference is that the posterior distribution by Bayes' rule is equivalent to the solution of the optimization problem

$$\min_{q(\mathbf{\Theta},\mathbf{Z},\mathbf{\Phi})} \mathrm{KL}[q(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}) \| p_0(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi})] - \mathbb{E}_q[\log p(\mathbf{W}|\mathbf{Z}, \mathbf{\Phi})]$$
$$\text{s.t.}: q(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}) \in \mathcal{P}, \tag{1}$$

where $\mathrm{KL}(q||p)$ is the Kullback-Leibler divergence and $\mathcal{P}$ is the space of probability distributions. In fact, if we add the constant $\log p(\mathbf{W})$ to the objective, the problem is the minimization of KL-divergence $\mathrm{KL}(q(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}) \| p(\mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}|\mathbf{W}))$, whose solution is naturally the desired posterior distribution by the Bayes' rule.

One advantage of this variational formulation of Bayesian inference is that it can be naturally extended to include some regularization terms on the desired post-data posterior distribution q. This insight has been taken to develop regularized Bayesian inference (RegBayes) [29], a computational framework of doing Bayesian inference with posterior regularization. As shown in [12], MedLDA is one example of RegBayes model. Moreover, our Gibbs max-margin topic models follow this similar idea too.

Gibbs Classifier: In learning theory, one approach to building classifiers with a posterior distribution of models is to minimize an expected loss, under the framework known as Gibbs classifiers (or stochastic classifiers) [14, 6, 10] with nice theoretical properties. For our case of inferring the distribution of latent topic assignments $\mathbf{Z}$ and the classification model $\boldsymbol{\eta}$, the expected margin loss is defined as follows. If we have drawn a sample of the topic assignments $\mathbf{Z}$ and the prediction model $\boldsymbol{\eta}$ from a posterior distribution $q(\boldsymbol{\eta}, \mathbf{Z})$, we can define the linear discriminant function

$$f(\boldsymbol{\eta}, \mathbf{z}; \mathbf{w}) = \boldsymbol{\eta}^\top \bar{\mathbf{z}}, \tag{2}$$

where $\bar{\mathbf{z}}$ is a the average topic assignment vector with each element being $\bar{z}_k = \frac{1}{N} \sum_{n=1}^N \mathbb{I}(z_n^k = 1)$, and make predictions using the latent Gibbs rule

$$\hat{y} = \mathrm{sign} f(\boldsymbol{\eta}, \mathbf{z}; \mathbf{w}). \tag{3}$$

Let $\zeta_d = \ell - y_d \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d$, where ℓ is a positive cost parameter. Then, the hinge loss of the classifier is $\mathcal{R}(\boldsymbol{\eta}, \mathbf{Z}) = \sum_d \max(0, \zeta_d)$, a function of the latent variables $(\boldsymbol{\eta}, \mathbf{Z})$, and the expected hinge loss is

$$\mathcal{R}(q) = \mathbb{E}_q[\mathcal{R}(\boldsymbol{\eta}, \mathbf{Z})] = \sum_d \mathbb{E}_q[\max(0, \zeta_d)],$$

[4] A K-dimension binary vector with only one nonzero entry.

a function of the posterior distribution $q(\boldsymbol{\eta}, \mathbf{Z})$. Since for any $(\boldsymbol{\eta}, \mathbf{Z})$, the hinge loss $\mathcal{R}(\boldsymbol{\eta}, \mathbf{Z})$ is an upper bound of the training error of the latent Gibbs classifier (3), i.e., $\mathcal{R}(\boldsymbol{\eta}, \mathbf{Z}) \geq \sum_d \ell \mathbb{I}(y_d \neq \hat{y}_d)$, we have

$$\mathcal{R}(q) \geq \sum_d \mathbb{E}_p[\ell \mathbb{I}(y_d \neq \hat{y}_d)].$$

In other words, $\mathcal{R}(q)$ is an upper bound of the expected training error of the Gibbs classifier (3). Thus, it is a good surrogate loss for learning a posterior distribution which could lead to a low training error in expectation.

Regularized Bayesian Inference: To integrate the above two components for hybrid learning, Gibbs MedLDA regularizes the properties of the topic representations by solving the regularized Bayesian inference (RegBayes) problem

$$\min_{q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) \in \mathcal{P}} \mathcal{L}(q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})) + 2c\mathcal{R}(q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})), \quad (4)$$

where c is a regularization parameter and $\mathcal{L}(q)$ is the objective of problem (1). Due to the strong coupling between the Gibbs classifier and the LDA model, we can expect to learn a posterior distribution $q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})$ that on one hand describes the observed data and on the other hand predicts as well as possible on training data.

In [28], extensive comparison with MedLDA was provided. Basically, MedLDA is also a RegBayes model, but it uses a different posterior regularization which is derived from an expected classifier. The effective algorithms for MedLDA are the two-stage approaches [12] that apply Monte Carlo methods as the inner inference engine and iteratively solve latent SVMs to learn the classifier distribution.

2.2 Formulation with Data Augmentation

If we directly solve problem (4), the expected hinge loss $\mathcal{R}$ is hard to deal with because of the non-differentiable max function. Fortunately, a simple collapsed Gibbs sampling algorithm can be developed with analytical forms of local conditional distributions, based on a data augmentation formulation of the expected hinge-loss.

Let $\phi(y_d|\mathbf{z}_d, \boldsymbol{\eta}) = \exp\{-2c \max(0, \zeta_d)\}$ be the unnormalized pseudo-likelihood of the response variable for document d. Then, problem (4) can be written as

$$\min_{q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) \in \mathcal{P}} \mathcal{L}(q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})) - \mathbb{E}_q[\log \phi(\mathbf{y}|\mathbf{Z}, \boldsymbol{\eta})], \quad (5)$$

where $\phi(\mathbf{y}|\mathbf{Z}, \boldsymbol{\eta}) = \prod_d \phi(y_d|\boldsymbol{\eta}, \mathbf{z}_d)$. Solving problem (5), we can get the normalized posterior distribution

$$q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) = \frac{p_0(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) p(\mathbf{W}|\mathbf{Z}, \boldsymbol{\Phi}) \phi(\mathbf{y}|\mathbf{Z}, \boldsymbol{\eta})}{\psi(\mathbf{y}, \mathbf{W})},$$

where $\psi(\mathbf{y}, \mathbf{W})$ is the normalization constant. Using the ideas of data augmentation [20, 17], the unnormalized pseudo-likelihood can be expressed as

$$\phi(y_d|\mathbf{z}_d, \boldsymbol{\eta}) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_d}} \exp\big(- \frac{(\lambda_d + c\zeta_d)^2}{2\lambda_d}\big) d\lambda_d$$

This result indicates that the posterior distribution of Gibbs MedLDA, $q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})$, can be expressed as the marginal of a higher dimensional distribution that includes the augmented variables $\boldsymbol{\lambda} = \{\lambda_d\}_{d=1}^D$. The complete posterior distribution is

$$q(\boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) = \frac{p_0(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) p(\mathbf{W}|\mathbf{Z}, \boldsymbol{\Phi}) \phi(\mathbf{y}, \boldsymbol{\lambda}|\mathbf{Z}, \boldsymbol{\eta})}{\psi(\mathbf{y}, \mathbf{W})},$$

where the pseudo-joint distribution of $\mathbf{y}$ and $\boldsymbol{\lambda}$ is $\phi(\mathbf{y}, \boldsymbol{\lambda}|\mathbf{Z}, \boldsymbol{\eta}) = \prod_d \frac{1}{\sqrt{2\pi\lambda_d}} \exp\big(- \frac{(\lambda_d + c\zeta_d)^2}{2\lambda_d}\big)$.

2.3 Inference with Collapsed Gibbs Sampling

Although with the data augmentation formulation we can do Gibbs sampling to infer the complete posterior distribution $q(\boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})$ and thus $q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})$ by ignoring $\boldsymbol{\lambda}$, the mixing would be slow due to the large sample space of all latent variables. One way to effectively accelerate the mixing is to integrate out the intermediate Dirichlet variables $(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ and build a Markov chain whose equilibrium distribution is the resulting marginal distribution $q(\boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{Z})$. This idea has been successfully used in LDA [11] and was taken in [28] to develop a collapsed Gibbs sampler for Gibbs MedLDA. With the data augmentation representation, this leads to an "augment-and-collapse" sampling algorithm for Gibbs MedLDA, as summarized below.

By integrating out the Dirichlet variables $(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ in the complete posterior distribution, we get the collapsed posterior distribution

$$q(\boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{Z}) \propto p_0(\boldsymbol{\eta}) \Big[\prod_{d=1}^D \frac{\delta(\mathbf{C}_d + \boldsymbol{\alpha})}{\delta(\boldsymbol{\alpha})} \Big] \prod_{k=1}^K \frac{\delta(\mathbf{C}_k + \boldsymbol{\beta})}{\delta(\boldsymbol{\beta})}$$
$$\prod_{d=1}^D \frac{1}{\sqrt{2\pi\lambda_d}} \exp\Big(- \frac{(\lambda_d + c\zeta_d)^2}{2\lambda_d}\Big),$$

where $\delta(\mathbf{x}) = \frac{\prod_{i=1}^{\dim(\mathbf{x})} \Gamma(x_i)}{\Gamma(\sum_{i=1}^{\dim(\mathbf{x})} x_i)}$; $\Gamma(\cdot)$ is the Gamma function; C_k^t is the number of times the term t being assigned to topic k over the whole corpus; $\mathbf{C}_k = \{C_k^t\}_{t=1}^V$ is the set of word counts associated with topic k; C_d^k is the number of times that terms being associated with topic k within the d-th document; and $\mathbf{C}_d = \{C_d^k\}_{k=1}^K$ is the set of topic counts for document d. Then, the conditional distributions used in collapsed Gibbs sampling are as follows.

For $\boldsymbol{\eta}$: For the commonly used isotropic Gaussian distribution $p_0(\boldsymbol{\eta}) = \prod_k \mathcal{N}(\eta_k; 0, \nu^2)$, where ν is a non-zero parameter, the conditional distribution of $\boldsymbol{\eta}$ given the other variables is also Gaussian:

$$q(\boldsymbol{\eta}|\mathbf{Z}, \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\eta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad (6)$$

where the posterior mean and the covariance matrix are $\boldsymbol{\Sigma} = \big(\frac{1}{\nu^2} I + c^2 \sum_d \frac{\bar{\mathbf{z}}_d \bar{\mathbf{z}}_d^\top}{\lambda_d}\big)^{-1}$ and $\boldsymbol{\mu} = \boldsymbol{\Sigma}\big(c \sum_d y_d \frac{\lambda_d + c\ell}{\lambda_d} \bar{\mathbf{z}}_d\big)$. We can easily draw a sample from this K-dimensional multivariate Gaussian distribution. The inverse can be robustly done using Cholesky decomposition, an $O(K^3)$ procedure. Since K is normally not large, the inversion can be done efficiently, especially in the applications where the number of documents is much larger than the number of topics.

For $\mathbf{Z}$: By canceling common factors, the conditional distribution of one variable z_{dn} given others $\mathbf{Z}_\neg$ is

$$q(z_{dn}^k = 1|\mathbf{Z}_\neg, \boldsymbol{\eta}, \boldsymbol{\lambda}, w_{dn} = t) \propto \frac{(C_{k,\neg n}^t + \beta_t)(C_{d,\neg n}^k + \alpha_k)}{\sum_t C_{k,\neg n}^t + \sum_{t=1}^V \beta_t}$$
$$\exp\Big(\frac{\gamma y_d (c\ell + \lambda_d)\eta_k}{\lambda_d} - c^2 \frac{\gamma^2 \eta_k^2 + 2\gamma(1-\gamma)\eta_k \Lambda_{dn}^k}{2\lambda_d}\Big), \quad (7)$$

where $C_{\cdot,\neg n}^{\cdot}$ indicates that term n is excluded from the corresponding document or topic; $\gamma = \frac{1}{N_d}$; and $\Lambda_{dn}^k = \frac{1}{N_d - 1} \sum_{k'} \eta_{k'} C_{d,\neg n}^{k'}$ is the discriminant function value without word n. We can see that the first term on the right hand is from the LDA model for observed word counts and it is the same as that in the collapsed Gibbs sampling method for LDA [11]; while the second term is from the supervised signal $\mathbf{y}$ which comes into play through the expected loss in problem (4).

For $\boldsymbol{\lambda}$: Finally, the conditional distribution of the augmented variables $\boldsymbol{\lambda}$ given the other variables is a generalized inverse Gaussian distribution [9]:

$$\begin{aligned} q(\lambda_d|\mathbf{Z},\boldsymbol{\eta}) &\propto \frac{1}{\sqrt{2\pi\lambda_d}} \exp\big(-\frac{(\lambda_d + c\zeta_d)^2}{2\lambda_d}\big) \\ &= \mathcal{GIG}\big(\lambda_d; \frac{1}{2}, 1, c^2\zeta_d^2\big), \end{aligned}$$

where $\mathcal{GIG}(x;p,a,b) = C(p,a,b)x^{p-1}\exp(-\frac{1}{2}(\frac{b}{x}+ax))$ and $C(p,a,b)$ is a normalization constant. Alternatively, λ_d^{-1} follows an inverse Gaussian distribution

$$q(\lambda_d^{-1}|\mathbf{Z},\boldsymbol{\eta}) = \mathcal{IG}\Big(\lambda_d^{-1}; \frac{1}{c|\zeta_d|}, 1\Big), \quad (8)$$

where $\mathcal{IG}(x;a,b) = \sqrt{\frac{b}{2\pi x^3}}\exp(-\frac{b(x-a)^2}{2a^2x})$ for $a,b>0$.

With the above conditional distributions, we can construct a Markov chain which iteratively draws samples of the classifier weights $\boldsymbol{\eta}$ using Eq. (6), the topic assignments $\mathbf{Z}$ using Eq. (7) and the augmented variables $\boldsymbol{\lambda}$ using Eq. (8), with an initial condition. To sample from an inverse Gaussian distribution, we apply the efficient transformation method with multiple roots [15]. In our experiments, we initially set $\boldsymbol{\lambda} = 1$ and randomly draw $\mathbf{Z}$ from a uniform distribution. In training, we run this Markov chain to finish the burn-in stage with T iterations. Then, we draw a sample $\hat{\boldsymbol{\eta}}$ as the Gibbs classifier to make predictions on testing data.

2.4 Prediction

To apply the Gibbs classifier $\hat{\boldsymbol{\eta}}$, we need to infer the topic assignments for testing document, denoted by $\mathbf{w}$. A fully Bayesian treatment needs to compute an integral in order to get the posterior distribution of the topic assignment given training data $\mathcal{D}$ and the testing document content $\mathbf{w}$:

$$p(\mathbf{z}|\mathbf{w},\mathcal{D}) \propto \int p(\mathbf{z},\mathbf{w},\Phi|\mathcal{D})d\Phi = \int p(\mathbf{z},\mathbf{w}|\Phi)p(\Phi|\mathcal{D})d\Phi,$$

where the second equality holds due to the conditional independence assumption of the documents given the topics. Various approximation methods can be applied to compute the integral. Here, we take the approach applied in [27, 12], which uses a point estimate of topics $\boldsymbol{\Phi}$ from training data and makes prediction based on them. Specifically, we use the MAP estimate $\hat{\boldsymbol{\Phi}}$ (a Dirac measure) to approximate the probability distribution $p(\boldsymbol{\Phi}|\mathcal{D})$. For the collapsed Gibbs sampler, an estimate of $\hat{\boldsymbol{\Phi}}$ using the samples is

$$\hat{\phi}_{kt} \propto C_k^t + \beta_t.$$

Then, given a testing document $\mathbf{w}$, we infer its latent components $\mathbf{z}$ using $\hat{\boldsymbol{\Phi}}$ as

$$p(z_n^k = 1|\mathbf{z}_{\neg n},\mathbf{w},\mathcal{D}) \propto \hat{\phi}_{kw_n}(C_{\neg n}^k + \alpha_k), \quad (9)$$

where $C_{\neg n}^k$ is the times that the terms in this document $\mathbf{w}$ are assigned to topic k with the n-th term excluded.

3. MULTI-TASK GIBBS MEDLDA

Multi-task learning is a scenario where multiple potentially related tasks are learned jointly with the hope that their performance can be boosted by sharing some statistic strengths among these tasks, and it has attracted a lot of research attention. In particular, learning a common representation shared by all the related tasks has proven to be an effective way to capture task relationships [2, 3, 29]. Here, we take the similar approach to learning multiple predictive models which share the common topic representations. As having been demonstrated in previous work [29] and our own experiments later, one successful application of the multi-task model is to do the single-label multi-class or multi-label [21] classification, where each task corresponds to a binary classifier to determine whether a data point belongs to a particular category.

3.1 The Model with Data Augmentation

We consider the L binary classification tasks and each task i is associated with a classifier with weights $\boldsymbol{\eta}_i$. We assume that all tasks work on the same set of input data $\mathbf{W} = \{\mathbf{w}_d\}_{d=1}^D$, but each data d has different binary labels $\{y_d^i\}_{i=1}^L$ in different tasks. When we have the classifier weights and the topic assignments $\mathbf{Z}$, drawn from a posterior distribution $q(\boldsymbol{\eta},\mathbf{Z})$, we follow the same principle as in Gibbs MedLDA and define the latent Gibbs rule for each task as

$$\forall i = 1,\dots L: \quad \hat{y}^i = \mathrm{sign}F(\boldsymbol{\eta}_i,\mathbf{z};\mathbf{w}) = \mathrm{sign}(\boldsymbol{\eta}_i^\top\mathbf{z}). \quad (10)$$

Let $\zeta_d^i = \ell - y_d^i\boldsymbol{\eta}_i^\top\bar{\mathbf{z}}_d$. The hinge loss of the classifier i is

$$\mathcal{R}_i(\boldsymbol{\eta}_i,\mathbf{Z}) = \sum_d \max(0,\zeta_d^i)$$

and the expected hinge loss is

$$\mathcal{R}_i(q) = \mathbb{E}_q[\mathcal{R}_i(\boldsymbol{\eta}_i,\mathbf{Z})] = \sum_d \mathbb{E}_q[\max(0,\zeta_d^i)].$$

For each task i, we can follow the argument as in Gibbs MedLDA to show that the expected loss $\mathcal{R}_i(q)$ is an upper bound of the expected training error $\sum_d \mathbb{E}_q[\ell\mathbb{I}(y_d^i \neq \hat{y}_d^i)]$ of the Gibbs classifier (10). Thus, it is a good surrogate loss for learning a posterior distribution which could lead to a low training error in expectation.

Then, following the similar procedure of defining the binary Gibbs MedLDA classifier, we can define the multi-task Gibbs MedLDA model as solving the following regularized Bayesian inference problem

$$\min_{q(\boldsymbol{\eta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\Phi})\in\mathcal{P}} \mathcal{L}(q(\boldsymbol{\eta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\Phi})) + 2c\mathcal{R}_{MT}(q(\boldsymbol{\eta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\Phi})), (11)$$

where the multi-task expected hinge loss is defined as a summation of the expected hinge loss of all the tasks

$$\mathcal{R}_{MT}(q(\boldsymbol{\eta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\Phi})) = \sum_{i=1}^L \mathcal{R}_i(q(\boldsymbol{\eta},\boldsymbol{\Theta},\mathbf{Z},\boldsymbol{\Phi})). \quad (12)$$

Due to the separability of the multi-task expected hinge loss, we can apply the same method as in the binary model to reformulate each task-specific expected hinge loss $\mathcal{R}_i$ as a scale mixture by introducing a set of augmented variables $\{\lambda_d^i\}_{d=1}^D$. More specifically, let

$$\phi_i(y_d^i|\mathbf{z}_d,\boldsymbol{\eta}) = \exp\{-2c\max(0,\zeta_d^i)\}$$

be the unnormalized pseudo-likelihood of the response variable for document d in task i. Then, we have

$$\phi_i(y_d^i|\mathbf{z}_d,\boldsymbol{\eta}) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_d^i}}\exp\big(-\frac{(\lambda_d^i + c\zeta_d^i)^2}{2\lambda_d^i}\big)d\lambda_d^i.$$

Algorithm 1 for Multi-task Gibbs MedLDA

```
Initialization: set $\boldsymbol{\lambda} = 1$ and randomly draw $z_{dk}$ from a uniform distribution.
for $m = 1$ to $T$ do
  for $i = 1$ to $L$ do
    draw a classifier $\boldsymbol{\eta}_i$ from the normal distribution (13)
  end for
  for $d = 1$ to $D$ do
    for each word $n$ in document $d$ do
      draw a topic using distribution (14)
    end for
    for $i = 1$ to $L$ do
      draw $(\lambda_d^i)^{-1}$ (and thus $\lambda_d^i$) from distribution (15).
    end for
  end for
end for
```

Obviously, when $L = 1$, the multi-task model reduces to the binary Gibbs MedLDA.

3.2 A Collapsed Gibbs Sampling Algorithm

Similar as in the binary Gibbs MedLDA, we can derive the collapsed Gibbs sampling algorithm, as outlined in Algorithm 1. Specifically, let

$$\phi_i(\mathbf{y}^i, \boldsymbol{\lambda}_i|\mathbf{Z}, \boldsymbol{\eta}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\lambda_d^i}} \exp\big(-\frac{(\lambda_d^i + c\zeta_d^i)^2}{2\lambda_d^i}\big)$$

be the joint pseudo-likelihood of the class labels $\mathbf{y}^i = \{y_d^i\}_{d=1}^D$ and the augmentation variables $\boldsymbol{\lambda}_i = \{\lambda_d^i\}_{d=1}^D$. Then, for the multi-task Gibbs MedLDA, we can integrate out the Dirichlet variables (Θ, Φ) and get the collapsed posterior distribution

$$q(\boldsymbol{\eta}, \boldsymbol{\lambda}, \mathbf{Z}) \propto p_0(\boldsymbol{\eta}) \Big[\prod_{d=1}^{D} \frac{\delta(\mathbf{C}_d + \boldsymbol{\alpha})}{\delta(\boldsymbol{\alpha})}\Big] \prod_{k=1}^{K} \frac{\delta(\mathbf{C}_k + \boldsymbol{\beta})}{\delta(\boldsymbol{\beta})}$$
$$\prod_{i=1}^{L}\prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\lambda_d^i}} \exp\Big(-\frac{(\lambda_d^i + c\zeta_d^i)^2}{2\lambda_d^i}\Big).$$

Then, we can derive the conditional distributions used in collapsed Gibbs sampling are as follows.

For $\boldsymbol{\eta}$: we also assume its prior is an isotropic Gaussian distribution $p_0(\boldsymbol{\eta}) = \prod_i \prod_k \mathcal{N}(\eta_{ik}; 0, \nu^2)$. Then, we have $q(\boldsymbol{\eta}|\mathbf{Z}, \boldsymbol{\lambda}) = \prod_{i=1}^L q(\boldsymbol{\eta}_i|\mathbf{Z}, \boldsymbol{\lambda})$, and for each task

$$q(\boldsymbol{\eta}_i|\mathbf{Z}, \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\eta}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \qquad (13)$$

where the posterior mean and the covariance matrix are $\boldsymbol{\Sigma}_i = \big(\frac{1}{\nu^2}I + c^2 \sum_d \frac{\bar{\mathbf{z}}_d \bar{\mathbf{z}}_d^\top}{\lambda_d^i}\big)^{-1}$ and $\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i\big(c\sum_d y_d^i \frac{\lambda_d^i + c\ell}{\lambda_d^i}\bar{\mathbf{z}}_d\big)$. Similarly, the inverse can be robustly and efficiently done using Cholesky decomposition, an $O(K^3)$ procedure.

For $\mathbf{Z}$: The conditional distribution of $\mathbf{Z}$ is

$$q(\mathbf{Z}|\boldsymbol{\eta}, \boldsymbol{\lambda}) \propto \prod_{d=1}^{D} \frac{\delta(\mathbf{C}_d + \boldsymbol{\alpha})}{\delta(\boldsymbol{\alpha})} \Big[\prod_{i=1}^{L} \exp\big(-\frac{(\lambda_d^i + c\zeta_d^i)^2}{2\lambda_d^i}\big)\Big]$$
$$\prod_{k=1}^{K} \frac{\delta(\mathbf{C}_k + \boldsymbol{\beta})}{\delta(\boldsymbol{\beta})}.$$

By canceling common factors, we can derive the conditional distribution of one variable z_{dn} given others $\mathbf{Z}_\neg$ as:

$$q(z_{dn}^k = 1|\mathbf{Z}_\neg, \boldsymbol{\eta}, \boldsymbol{\lambda}, w_{dn} = t) \propto \frac{(C_{k,\neg n}^t + \beta_t)(C_{d,\neg n}^k + \alpha_k)}{\sum_t C_{k,\neg n}^t + \sum_{t=1}^V \beta_t}$$
$$\prod_{i=1}^{L} \exp\Big(\frac{\gamma y_d^i (c\ell + \lambda_d^i)\eta_{ik}}{\lambda_d^i} - c^2 \frac{\gamma^2 \eta_{ik}^2 + 2\gamma(1-\gamma)\eta_{ik}\Lambda_{dn}^i}{2\lambda_d^i}\Big), \qquad (14)$$

where $\Lambda_{dn}^i = \frac{1}{N_d - 1}\sum_{k'} \eta_{ik'} C_{d,\neg n}^{k'}$ is the discriminant function value without word n. We can see that the first term is from the LDA model for observed word counts and the second term is from the supervised signal $\{y_d^i\}$ from all the multiple tasks.

For $\boldsymbol{\lambda}$: Finally, one can derive that the conditional distribution of the augmented variables $\boldsymbol{\lambda}$ is fully factorized, $q(\boldsymbol{\lambda}|\mathbf{Z}, \boldsymbol{\eta}) = \prod_i \prod_d q(\lambda_d^i|\mathbf{Z}, \boldsymbol{\eta})$, and each variable follows a generalized inverse Gaussian distribution

$$q(\lambda_d^i|\mathbf{Z}, \boldsymbol{\eta}) = \mathcal{GIG}\big(\lambda_d^i; \frac{1}{2}, 1, c^2(\zeta_d^i)^2\big).$$

Therefore, $(\lambda_d^i)^{-1}$ follows an inverse Gaussian distribution

$$q((\lambda_d^i)^{-1}|\mathbf{Z}, \boldsymbol{\eta}) = \mathcal{IG}\Big((\lambda_d^i)^{-1}; \frac{1}{c|\zeta_d^i|}, 1\Big). \qquad (15)$$

4. PARALLEL IMPLEMENTATION

One nice property of the above Gibbs sampling algorithm for the multi-task Gibbs MedLDA[5] is that it can be easily parallelized, due to the following observations.

- The augmented variables $\boldsymbol{\lambda}$ are "locally" associated with each document. Therefore, we can easily parallelize the step of sampling λ_d (or λ_d^i for multi-task Gibbs MedLDA) to multiple cores and multiple machines.

- The data points contribute to the global variables $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ through the simple summation operator, and for different tasks the global variables $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ are conditionally independent given $(\mathbf{Z}, \boldsymbol{\lambda})$. Due to the separability of summation, we can easily partition the data into different machines for performing local summation, followed by a global aggregation. This suggests a MapReduce architecture for parallelizing the step of updating $\boldsymbol{\eta}$.

The true difficulty is on parallelizing the sampling step of topic assignments, z_{dn}, which depend on the global variables C_k^t (i.e., the topic-word count table) that need to be updated/synchronized during the local sampling process. Fortunately, our sampling algorithm is highly modular—once the classifier weights $\boldsymbol{\eta}$ (and the augmented variables $\boldsymbol{\lambda}$) are given, the sampling of each topic assignment is almost the same as that in the standard LDA, except some additional computation which is carried out locally to each document, as show in Eq. (14). Therefore, we can leverage the recent advances in parallel topic models [1, 19] to solve this problem. Given a multi-core and multi-machine LDA sampler, we can develop our parallel sampler using a simple procedure as detailed below.

[5] The binary Gibbs MedLDA model is a special case of the multi-task model with $L = 1$.

Table 1: The amount of time (seconds) taken by the step of sampling $\boldsymbol{\eta}$ and network communication on the Wiki data set. K–the number of topics; M–the number of machines; communication includes both reduce and broadcast time.

	sample $\boldsymbol{\eta}$	communication	total
K=500, M=10	71.26 (2.63%)	88.28 (3.25%)	2712.64
K=500, M=20	71.29 (5.01%)	52.90 (3.72%)	1423.53
K=1000, M=10	1447.07 (16.99%)	226.47 (2.66%)	8517.97
K=1000, M=20	1449.51 (28.73%)	135.44 (2.68%)	5044.61

Let M be the total number of processes (or machines) and let $\mathcal{D}_m$ be the data in process m. Then each process m performs the following computations

1. **draw topic assignments**: use the given LDA sampler to draw the topic assignments with the updated equation to compute local probability (14);

2. **draw scale parameters**: draw the scale parameter for each document using the distribution (15);

3. **compute local statistics**: compute the following statistics

$$\boldsymbol{\mu}_i^m = \sum_{d=1}^{D_m} y_d^i \frac{\lambda_d^i + c\ell}{\lambda_d^i} \bar{\mathbf{z}}_d, \ \boldsymbol{\Sigma}_i^m = \sum_{d=1}^{D_m} \frac{\bar{\mathbf{z}}_d \bar{\mathbf{z}}_d^\top}{\lambda_d^i}, \qquad (16)$$

   where D_m is the number of data samples in $\mathcal{D}_m$.

Since $\boldsymbol{\Sigma}_i^m$ is symmetric, it suffices to compute only the upper or lower triangle. After process m has finished the local computation, it passes the local statistics $\boldsymbol{\mu}_i^m$ and $\boldsymbol{\Sigma}_i^m$ to the master process, which performs the following operations

1. **compute $\boldsymbol{\Sigma}$**: by collecting the message from slaves, it computes $\boldsymbol{\Sigma}_i = \left(\frac{1}{\nu^2} I + c^2 \sum_m \boldsymbol{\Sigma}_i^m\right)^{-1}$.

2. **compute $\boldsymbol{\mu}$**: after obtaining the new $\boldsymbol{\Sigma}_i$, it updates $\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i \big(c \sum_m \boldsymbol{\mu}_i^m\big)$.

3. **draw classifier weights $\boldsymbol{\eta}$**: when all local statistics are reduced to master process, it samples $\boldsymbol{\eta}$ using distribution computed by (13).

4. **synchronize classifier weights $\boldsymbol{\eta}$**: after sampling $\boldsymbol{\eta}$, it broadcasts the new classifier weights to the slaves.

There are indeed more sophisticated synchronization strategies that could be applied, however as shown in Table 1, both communication and sampling of $\boldsymbol{\eta}$ take little time compared to the main algorithm. Therefore this treatment is sufficient to achieve high performance. For the LDA sampler, we use the current state-of-the-art method [1].

5. EXPERIMENTS

We run our experiments on a cluster with 20 nodes, where each node is equipped with two 6-core CPUs (2.93GHz).

5.1 Data Sets

We present experiments on several public text categorization data sets, whose statistics are shown in Table 2. The 20Newsgroups (20NG) data set consists of about 20K postings within 20 groups and each document has a single categorical label, ranging from 1 to 20; we follow the same setting as in [27] to build train/test partition and the vocabulary. The Wiki data set is built from the large Wikipedia set used in the PASCAL LSHC challenge 2012, and each document has multiple labels. The original data set[6] is extremely imbalanced. We built our data set by selecting the 20 categories that have the largest numbers of documents and keeping all the documents that is labeled by one of the 20 categories. The third data set is the Reuter's Corpus Volume (RCV1-v2) [13], another standard benchmark[7] of which each document has multiple labels. To test the scalability of our method, we have partitioned the data set into training and testing sets with a ratio of 7 : 1.

Table 2: Statistics of the data sets. N–the number of documents; V–the number of terms; and L–the number of categories.

data set	N-train	N-test	V	L	type
20NG	11,269	7,505	61,188	20	single-label
Wiki	1,100,000	5,000	917,683	20	multi-label
RCV	703,863	100,551	288,062	103	multi-label

5.2 Single-label Classification

We first present some empirical results on the singly labeled 20Newsgroups data set. For the binary Gibbs MedLDA, one-vs-all is an effective strategy to do multi-class classification [18]. To make the multi-task Gibbs MedLDA (MT-GibbsMedLDA) applicable to the singly labeled data set, we need to transform the true label to get the label for each binary task. Let the label space be $\mathcal{Y} = \{1, \ldots, L\}$. We define one binary classification task for each category i and the task is to distinguish whether a data example belongs to the class i (with binary label +1) or not (with binary label −1). All the binary tasks share the same topic representations. To apply the model as we have presented in Section 3, we need to determine the true binary label of each document in a task. Given the multi-class label y_d of document d, this can be easily done by defining

$$\forall i = 1, \ldots, L: \ y_d^i = \begin{cases} +1 \text{ if } y_d = i \\ -1 \text{ otherwise} \end{cases}.$$

Figure 1 shows the accuracy and training time of the multi-task Gibbs MedLDA, the one-vs-all binary Gibbs MedLDA [28], the multi-class MedLDA using Gibbs sampling [12] built with an expected classifier, and the two-stage approach of first using Gibbs LDA (gLDA) [11] to learn latent topic features and then building a SVM classifier[8]. We can see that the multi-task formulation of Gibbs MedLDA produces comparable performance as the one-vs-all method; while the two Gibbs MedLDA models slightly outperform MedLDA. Furthermore, the multi-task model is computationally more efficient than the one-vs-all approach due to the less number of topics. A naive parallelization of the one-vs-all approach is to learn the 20 binary classifiers in parallel, which improves the efficiency. However, the one-vs-all approach may not be a good choice if we want to get a holistic view of

[6] Available at: http://lshtc.iit.demokritos.gr/

[7] Available at: http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

[8] The SVM classifier built on raw bag-of-words as well other variants of supervised topic models were outperformed by MedLDA. See [27] for an extensive comparison.

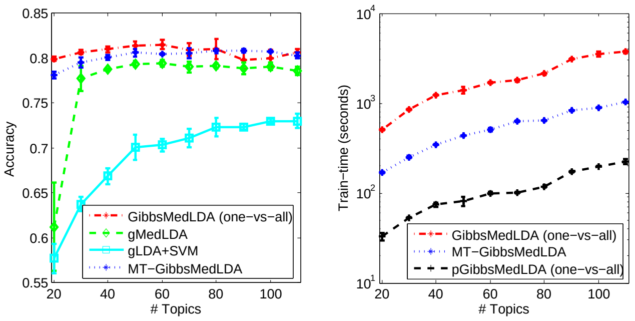

Figure 1: Classification accuracy and training time of multi-task GibbsMedLDA, GibbsMedLDA with one-vs-all strategy, and the multi-class MedLDA with stage-wise Gibbs sampling.

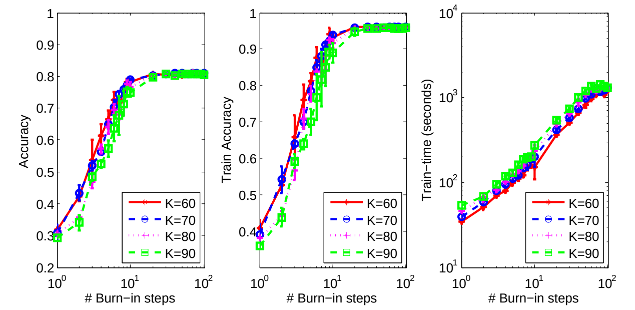

Figure 2: The classification accuracy, training accuracy and training time of the multi-task Gibbs MedLDA with different burn-in steps.

the topic structures of the entire corpus, because it learns 20 independent sets of topics which are not easy to be merged. From Figure 2, we can see that the sampling algorithm converges quickly to stable performance within 40 iterations. These results demonstrate the effectiveness of the multi-task Gibbs MedLDA. In all the experiments, we fixed $\boldsymbol{\alpha} = 6.4\mathbf{e}$, $\boldsymbol{\beta} = 0.01\mathbf{e}$, $\ell = 1$ and $c = 16$. As in [28], Gibbs MedLDA is insensitive to these parameters in wide ranges. We omitted the sensitivity analysis for saving space.

Figure 3 shows the accuracy and training time of the multi-task Gibbs MedLDA with different numbers of machines and different numbers of CPU cores. We can see that the single-machine-multi-core implementation is about 1 order of magnitude faster than the single-core version; while using multiple machines can further improve the efficiency dramatically. Meanwhile, the classification accuracy does not sacrifice much in a distributed environment.

5.3 Multi-label Classification

We now present the experiments of multi-task Gibbs MedLDA on the two multi-label data sets, where each task is a binary classifier to identify whether a document belongs to a particular category. We use the F-measure, a harmonic mean of precision and recall, to evaluate the performance.

Figure 4 shows the classification F-measure and training time of multi-task Gibbs MedLDA, comparing with the linear SVM classifier built on raw bag-of-words features and the two stage approach, LDA+SVM, which first fits an LDA model using all the documents and then learn a linear SVM classifier. For Gibbs MedLDA, we report the performance in the single-machine-multi-core setting as well as the setting with all the 20 machines. For LDA+SVM, we use the public Yahoo-LDA on 20 machines (240 CPU cores)[9]. Note that for fair comparison, we use the standard collapsed Gibbs sampling for both LDA and Gibbs MedLDA, although Yahoo-LDA has the option to perform fast Gibbs sampling [26]. Developing a fast Gibbs sampling algorithm for Gibbs MedLDA is one of our future work. To learn the SVM classifiers, we use the liblinear package[10] with the one-vs-all strategy and train each binary classifier on one of the 20 machines. We can see that Gibbs MedLDA dramatically improves the classification performance over the two-stage approach of LDA+SVM. Furthermore, we found that the SVM classifier on the raw features doesn't work well, mainly due to the sparsity issue of the feature space. For training time, the amount of time required by the supervised Gibbs MedLDA is comparable to that by the unsupervised LDA. These results are impressive since Gibbs MedLDA performs two jobs of topic discovery and classifier learning jointly, while LDA performs topic discovery only.

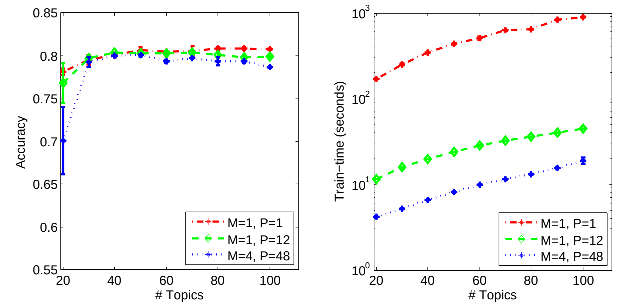

Figure 3: The classification accuracy and training time of the multi-task Gibbs MedLDA with different numbers of machines (M) and CPU cores (P).

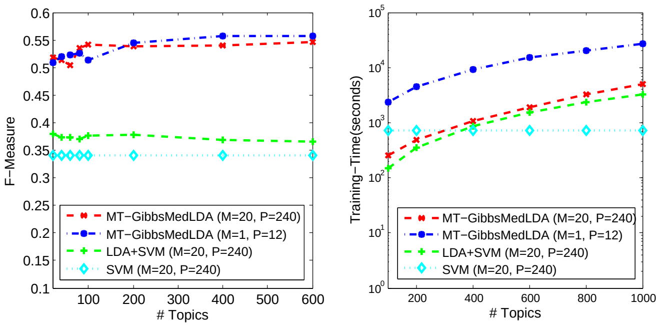

Figure 4: F-measure and training time of various methods on the Wiki data set.

Figure 5 presents how the classification performance and training time of the distributed MT-GibbsMedLDA ($M = 20$ and $P = 240$) change with respect to T (i.e., the number of burn-in steps). We can observe that with a number of burn-in steps (e.g., 40, 60 or 80), we can get quite stable prediction performance, which 20 is not sufficiently large; and using a large T generally increases the training time about linearly. We set $T = 40$ in the experiments.

5.4 Scalability

[9] Available at: https://github.com/shravanmn/Yahoo_LDA.

[10] Available at: http://www.csie.ntu.edu.tw/~cjlin/liblinear/

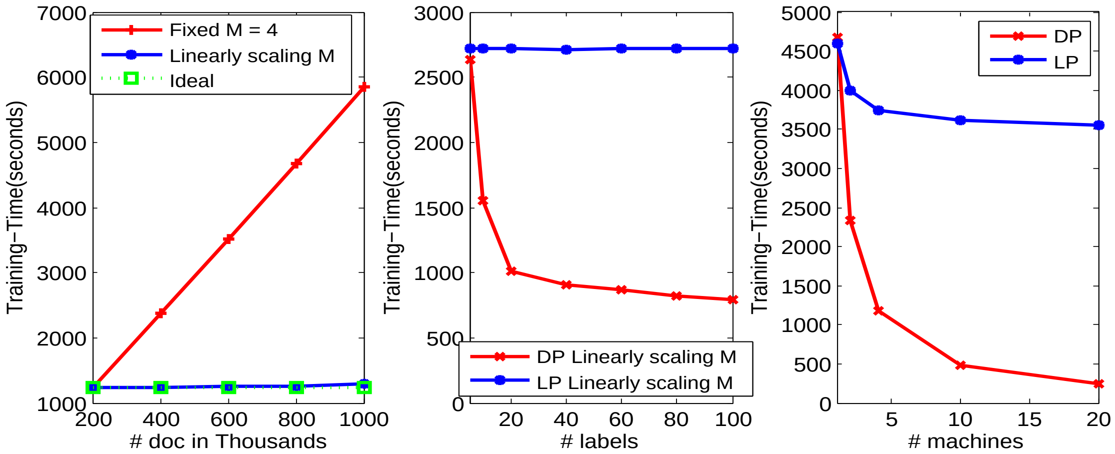

Figure 6: Scalability analysis. (Left) When fixing M (i.e., the number of machines) we observe a linear dependence between training time and the amount of data; when the data to machine ratio is kept constant, the training time remains about constant. Here, we use $\{4, 8, 12, 16, 20\}$ machines such that each machine receives 50K documents. (Middle) when fixing the number of data points, we observe a sublinear decrease of training time for the DP strategy as the number of machines increases; while the time of the LP strategy remains constant when the label to machine ratio is fixed. Here, we use $\{1, 2, 4, 8, 12, 16, 20\}$ machines such that each machine in the LP strategy receives 5 categories. (Right) When both the number of labels and the number of documents are fixed, increasing the number of machines leads to a linear decrease (see text for explanation) of the running time for DP, while LP is slower. Here, we use $\{1, 2, 4, 10, 20\}$ machines.

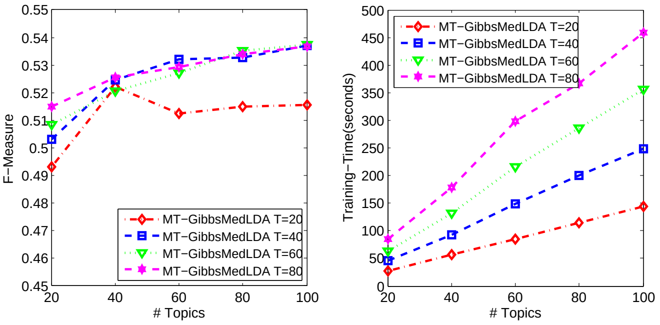

Figure 5: F-measure and training time of MT-GibbsMedLDA ($M = 20$ and $P = 240$) with different numbers of burn-in steps on the Wiki data set.

Figure 6 (Left) shows the scalability analysis for MT-GibbsMedLDA on the Wiki data set. We can draw the following conclusions. First, when the computational resources are kept fixed, the amount of time required to process data scales linearly with the amount of data. Second, when the data to machine ratio is kept constant, the amount of time required to process the data is about constant, very close to the ideal line. The tiny performance loss with more machines is mainly due to the latency in network communication. But in general, these observations suggest that the parallel implementation of our sampling algorithms can scale nicely to massive data sets. For example, with 20 machines (240 CPU cores), we can finish the training on 2.8M documents with 20 categories within an hour.

Figure 6 (Middle) shows the scalability analysis on the RCV data set, when changing the number of labels while the total number of data samples is unchanged. We consider two parallelization strategies:

- **Document partition (DP)**: build a single multi-task model and split the data equally to multiple machines;
- **Label partition (LP)**: split the total number of categories equally and build independent multi-task models, one on each machine;

For label partition, it confirms that the amount of time remains constant with respect to the number of labels when the label to machine ratio is kept constant, e.g., 5 in our case. While for the document partition strategy, since increasing the number of labels doesn't change the amount of data, the running time decreases as more machines are used; furthermore, since more classifier parameters are indeed introduced the running time decreases sub-linearly when the label to machine ratio is fixed.

Finally, Figure 6 (Right) shows the amount of time required by the distributed MT-GibbsMedLDA when the number of machines increases, while the number of labels and the number of documents are fixed, on the Wiki data set. We can observed that for the strategy of DP, the amount of time decreases about linearly, i.e., when the number of machines is doubled, the running time decreases to about a half; while LP is slower because each machine in LP needs to process more data and the total number of topics is larger. Also, note that the most right point of LP is in fact the one-vs-all approach with binary GibbsMedLDA, which is much slower than MT-GibbsMedLDA with the DP strategy; this demonstrates the advantages of the multi-task formulation.

6. CONCLUSIONS AND DISCUSSIONS

We have presented a highly scalable approach to building max-margin supervised topic models for large-scale multi-class and multi-label text categorization. Our Gibbs sampling algorithm builds on a novel formulation of multi-task Gibbs max-margin topic models as well as a data augmentation formulation. The algorithm is modular and can take advantages of recent advances in scalable inference for unsupervised topic models. Extensive results on large scale data sets demonstrate that Gibbs max-margin topic models can significantly improve the classification performance while require comparable time as the unsupervised topic models.

Due to the restriction of computational resources, our experiments have been carried out on a relatively small cluster with tens of machines. In the future, we plan to carry out careful investigations on large clusters (e.g., with thousands of machines) with massive corpora consisting of tens of thousands of categories and millions of data points, as commonly encountered in PASCAL and ImageNet challenges. Finally, the data augmentation techniques are general and can be applied to improve the inference accuracy of other topic models or latent variable models in general, such as relational topic models [8] for network analysis and matrix factorization [24] for collaborative filtering.

7. ACKNOWLEDGMENTS

This work is supported by National Key Foundation R&D Projects (No.s 2013CB329403, 2012CB316301), Tsinghua Initiative Scientific Research Program No.20121088071, and the 221 Basic Research Plan for Young Faculties at Tsinghua University.

8. REFERENCES

[1] A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. Smola. Scalable inference in latent variable models. In *International Conference on Web Search and Data Mining (WSDM)*, 2012.

[2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research (JMLR)*, (6):1817–1853, 2005.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[4] D. Blei and J. McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 121–128, 2007.

[5] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[6] O. Catoni. PAC-Bayesian supervised classification: The thermodynamics of statistical learning. *Monograph series of the Institute of Mathematical Statistics*, 2007.

[7] J. Chang and D. Blei. Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[8] N. Chen, J. Zhu, F. Xia, and B. Zhang. Generalized relational topic models with data augmentation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.

[9] L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, 1986.

[10] P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning (ICML)*, pages 353–360, 2009.

[11] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of National Academy of Science (PNAS)*, pages 5228–5235, 2004.

[12] Q. Jiang, J. Zhu, M. Sun, and E. Xing. Monte Carlo methods for maximum margin supervised topic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[13] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research (JMLR)*, 5:361–397, Dec. 2004.

[14] D. McAllester. PAC-Bayesian stochastic model selection. *Machine Learning*, 51:5–21, 2003.

[15] J. Michael, W. Schucany, and R. Haas. Generating random variates using transformations with multiple roots. *The American Statistician*, 30(2):88–90, 1976.

[16] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *Journal of Machine Learning Research (JMLR)*, (10):1801–1828, 2009.

[17] N. Polson and S. Scott. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1–24, 2011.

[18] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research (JMLR)*, (5):101–141, 2004.

[19] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Very Large Data Base (VLDB)*, 3(1-2):703–710, 2010.

[20] M. Tanner and W.-H. Wong. The calculation of posterior distributions by data augmentation. *Journal of the Americal Statistical Association (JASA)*, 82(398):528–540, 1987.

[21] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pages 667–685, 2010.

[22] D. van Dyk and X. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics (JCGS)*, 10(1):1–50, 2001.

[23] Y. Wang and G. Mori. Max-margin latent Dirichlet allocation for image classification and annotation. In *British Machine Vision Conference (BMVC)*, 2011.

[24] M. Xu, J. Zhu, and B. Zhang. Fast max-margin matrix factorization with data augmentation. In *International Conference on Machine Learning (ICML)*, 2013.

[25] S. Yang, J. Bian, and H. Zha. Hybrid generative/discriminative learning for automatic image annotation. In *Uncertainty in Artificial Intelligence (UAI)*, 2010.

[26] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *ACM SIGKDD*, pages 937–946, 2009.

[27] J. Zhu, A. Ahmed, and E. Xing. MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research (JMLR)*, (13):2237–2278, 2012.

[28] J. Zhu, N. Chen, H. Perkins, and B. Zhang. Gibbs max-margin topic models with fast sampling algorithms. In *International Conference on Machine Learning (ICML)*, 2013.

[29] J. Zhu, N. Chen, and E. Xing. Infinite latent SVM for classification and multi-task learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1620–1628, 2011.