Linear Time Samplers for Supervised Topic Models using Compositional Proposals

Xun Zheng

15 March 2016

Abstract

Background. Topic models are effective probabilistic tools for processing large collections of unstructured data. With the exponential growth of modern industrial scale data and the ambition to capture more refined hidden information through exploring bigger models, topic models face the significant scalability challenge which lots of previous work has been devoted to addressing, culminating in the recent fast Markov chain Monte Carlo sampling algorithms for the *unsupervised* latent Dirichlet allocation (LDA) formulations.

Aim. In this work we aim to extend the recent sampling advances for unsupervised LDA models to *supervised* tasks. We focus on the MedLDA model that is able to simultaneously discover latent structures and make accurate predictions. We compare this against existing algorithms and show the improved speed of the new algorithm.

Data. We use the 20 newsgroups data set, which is a collection of nearly 20,000 newsgroup documents, divided across 20 different newsgroups, such as computer hardware, sports, politics, and religion. We also use a multi-labeled Wikipedia data set where labels represent categories of each Wikipedia article.

Methods. Through innovative combinations of a set of sampling techniques such as *mixtures of MCMC kernels*, we are able to reduce the complexity to linear in the number of topics and the data size.

Results. In the experiments we observe an order of magnitude speedups over the current state-of-the-art implementation, while achieving similar prediction performances.

Conclusions. To our best knowledge, this is the first linear time sampling algorithm for supervised topic models. Through the experiments, we verify that the new algorithm helps MedLDA to learn discriminative latent structures in the text more efficiently.

Keywords: Inference, MCMC, Topic Models, Scale Mixtures

1 Introduction

Bayesian methods have been extremely influential in the past twenty years, thanks to modern Markov chain Monte Carlo sampling advances that free Bayesians from making strong conjugacy assumptions and render posterior distributions amenable to efficient analysis. One prominent example is the topic models, such as the latent Dirichlet analysis (Blei et al., 2003, LDA). Through explicitly modeling the latent probabilistic relations among observed variables, topic models can effectively reduce large unstructured categorical data into semantically meaningful and interpretable low dimensional representations. Partly due to its ability in resolving the polysemy problem (*i.e.*, the same word can have different meanings under different context), topic models have been widely used in many practical applications, for instance genetics Pritchard et al. (2000), image analysis Li and Perona (2005), text mining Griffiths and Steyvers (2004); Blei et al. (2003), collaborative filtering Chen et al. (2008), prediction tasks Zhu et al. (2012); Blei and McAuliffe (2007), and many more.

One significant challenge topic models face is their scalability. On one hand, technology innovations have made it possible to collect very large amount of industrial scale data in an unprecedented rate. On the other hand, the need to capture more subtle information hidden in the data requires bigger, more sophisticated mathematical models, leading eventually to more unknown parameters. Consequently, there has been a lot of recent work aiming at scaling up various topic models to very large datasets and very big models. Sampling algorithms, in particular, Markov chain Monte Carlo (MCMC) methods, have played an eminent role in this direction.

Since in this work we mostly focus on the LDA model (and related), due to space limits, we can only mention a few inspiring contributions in this regime. While the original LDA model relied on the variational inference method Blei et al. (2003), soon Griffiths and Steyvers (2004) proposed the first efficient collapsed Gibbs sampling algorithm that scales much better. Exploiting the observation that only few topics appear in a certain document and few words are assigned to a certain topic, the SparseLDA Yao et al. (2009) further reduces the sampling cost of Griffiths and Steyvers (2004). Another significant contribution is the AliasLDA Li et al. (2014), which made explicit the crucial insight that model parameters only change slowly during sampling. By a masterful combination of the independent Metropolis-Hastings algorithm Metropolis et al. (1953); Hastings (1970) with Walker's alias method Walker (1977) for sampling discrete proposals in amortized constant time, AliasLDA was able to enjoy even better efficiency. Lastly, built on the insight of AliasLDA, the very recent LightLDA Yuan et al. (2015) accomplished the first linear time sampling algorithm for LDA. Impressive as they are, the aforementioned works suffer one common drawback: they all work on the *unsupervised* LDA model, hence are not able to exploit any supervised information (*e.g.* labels, tags, annotations, *etc.*).

The Gibbs MedLDA Zhu et al. (2014), on the other hand, is a hybrid model that combines the unsupervised LDA representation and the supervised large-margin classifier under the maximum entropy principle. Compared with other supervised LDA formulations (*e.g.* Blei and McAuliffe, 2007; Zhu et al., 2012), Gibbs MedLDA often leads to better performance, and more importantly, it can directly exploit modern MCMC techniques without positing any restrictive assumption on the posterior distribution. The state-of-the-art implementation of Gibbs MedLDA, in Zhu et al. (2014, 2013), costs $O(K^3 + DK^2 + D\bar{N}K)$ when there are K topics and D documents with average length $\bar{N}$. Although still faster than existing supervised LDA alternatives, the superlinear dependence on K prevents Gibbs MedLDA from scaling up to very large datasets or even moderately large models (*e.g.* K around a few tens to hundreds).

The main goal of this work is to extend the recent fast sampling algorithms Li et al. (2014); Yuan et al. (2015) from the unsupervised LDA to supervised tasks, in order to analyze the latent structure of labeled text data more efficiently. The mathematical formulation of Gibbs MedLDA imposes a fair amount of computational challenges, thus makes up an ideal model for demonstrating the techniques we have developed. More precisely, we make the following contributions: 1). For the unsupervised latent representation part, we extend the factorized proposal in LightLDA Yuan et al. (2015) to regularized posterior distributions. This requires building a local proposal per document and we show that its complexity can still be amortized to $O(1)$. 2). For the supervised classifier part, we propose to apply the Gibbs sampler to draw the large-margin classifier, completely bypassing the costly need of forming and inverting the precision matrix. 3). By carefully combing a set of sampling techniques we are able to significantly reduce the complexity of Gibbs MedLDA from $O(K^3 + DK^2 + D\bar{N}K)$ to $O(DK + D\bar{N})$, without altering the posterior distribution at all. To our best knowledge, this is the first linear time sampling algorithm for supervised LDA models. 4). Through extensive experiments we verify that our proposed linear time sampling algorithm converges an order of magnitude faster than the current state-of-the-art implementation while achieving similar prediction accuracy. The improvement is expected to be even larger when the model size grows.

Problem and approach: One significant challenge topic models face is the scalability. There have been some works on more efficient algorithms for unsupervised LDA, however supervised topic models remain inefficient due to their complex structure. The goal of this work is to extend the recent fast sampling algorithms from the unsupervised LDA to supervised tasks, in order to analyze the latent structure of labeled text data more efficiently. To achieve this, we combine several classic MCMC techniques such as mixture of kernels.

Outline: We first collect some background material on MCMC sampling in §2.1 for later use. Then in §2.2 we briefly recall the Gibbs MedLDA model and in §2.3 we review some recent sampling advances for LDA that inspired this work. We present in §3 our main result, a linear time sampling algorithm for supervised LDA models. Extensive experiments are reported in §4, and we conclude in §5.

2 Preliminaries

We begin by briefly reviewing some MCMC background, in particular the composition principle for transition kernels. Next, we recall the Gibbs MedLDA model and review some recent fast sampling advances for LDA.

2.1 MH and MCMC Sampling

For probability density functions $p(\cdot)$ that are (computationally) hard to sample directly, the Metropolis-Hastings (MH) algorithm Metropolis et al. (1953); Hastings (1970) offers a very convenient alternative. It repeats the following steps using a proposal density $q(\cdot|\cdot)$:

- Draw $Y \sim q(\cdot|X)$;
- Set $X = Y$ with probability

$$A = A(X, Y) = \min\left\{\frac{p(Y)q(X|Y)}{p(X)q(Y|X)}, 1\right\}. \tag{1}$$

Of course, MH is efficient only when it is easy to draw from the proposal density $q(\cdot|X)$ and when the acceptance probability A is large. The two, as defined, are at odds with each other: the proposal that provides the largest acceptance ratio (*i.e.* $A \equiv 1$) is the density $q(\cdot|X) = q(\cdot) = p(\cdot)$, which is what we avoid to draw directly in the first place. Nevertheless, by carefully balancing the heaviness of drawing from the proposal and the probability of accepting the proposed sample Y, we can achieve great flexibility and efficiency. In this work we choose the *independent* Metropolis algorithm, *i.e.* the proposal $q(Y|X) = q(Y)$ does not depend on X. Since the target density $p(\cdot)$ appears in ratio in the acceptance probability (1), we need only know it up to a (multiplicative) universal constant, which can be very convenient for Bayesian posterior analysis.

Underlying the MH algorithm is a Markov chain with a specific transition kernel (*e.g.* transition probability matrix)

$$K(x, y) = A(x, y) \cdot q(y|x) + (1 - r(x)) \cdot \delta_x(y), \tag{2}$$

where $r(x) = \int A(x, y)q(y|x)\mathrm{d}y$ and δ_x denotes the Dirac point mass at x. By simulating the Markov chain the sample X will eventually follow the (unique) stationary distribution $\pi(\cdot) = p(\cdot)$, under mild regularity conditions. More generally, under the name Markov chain Monte Carlo (MCMC), one can simulate *any* Markov chain (not necessarily constructed as in the MH algorithm) as long as its stationary density $\pi(\cdot)$ coincides (uniquely) with the target density $p(\cdot)$. Here again we face the tradeoff between the convenience of drawing from the chain $K(\cdot, \cdot)$ and its mixing rate of convergence to the target density $p(\cdot)$. The modern success of Bayesian inference, including this work, heavily relies on carefully balancing this tradeoff.

One great flexibility of MCMC is that we can *compose* different transition kernels, to achieve better performance. The underlying idea is extremely simple:

Theorem 1 (Composition Principle, *e.g.* Tierney (1994)) *If both transition kernels K_1 and K_2 have $p(\cdot)$ as stationary density, so do $K_1 \circ K_2$ and $\gamma K_1 + (1 - \gamma)K_2$ for any $\gamma \in [0, 1]$.*

The former kernel $K_1 \circ K_2$ corresponds to drawing *cyclically* from K_1 and K_2 while the latter kernel $\gamma K_1 + (1 - \gamma)K_2$ corresponds to drawing from K_1 with probability γ or K_2 otherwise. The point is that whenever it is *convenient* to construct multiple good transition kernels for our problem, we do not have to make a choice among them: we use them *all* through composition. This can dramatically improve the mixing property of the underlying Markov chain (without complicating the sampling procedure much). It is clear that the composition principle extends immediately to more than two kernels, more precisely three in our case.

Perhaps the most famous example of the composition principle is the Gibbs sampler Geman and Geman (1984): In order to draw from the joint density $Z = (Z_1, Z_2) \sim p(\cdot)$ we sample *cyclically* from the conditional kernel

$$K(f(X), Z) = \Pr\big(Z \mid f(Z) = f(X)\big), \tag{3}$$

using two different functions $f_1(x_1, x_2) = x_1, f_2(x_1, x_2) = x_2$, upon which we have the familiar rule:

$$X_2 \sim K_1(X_1, X_2) = p(X_2|X_1), X_1 \sim K_2(X_2, X_1) = p(X_1|X_2).$$

Note that neither kernel K_1 or K_2 has the target density $p(\cdot)$ as the *unique* stationary density, but after composition the uniqueness is often automatic. The Gibbs sampler also opens the possibility for data augmentation Tanner and Wong (1987): Suppose we want to sample from $p(X)$, which is

computationally intensive. By augmenting with "virtual" data W we can sample the joint density $p(X, W)$ using the Gibbs sampler, provided that the conditional densities $p(X|W)$ and $p(W|X)$ are easy to sample from. Dropping W we get the desired sample X that follows $p(\cdot)$ after burn-in.

Our main goal is to carefully combine the above MCMC sampling techniques: (independent) MH, composition principle, Gibbs sampler, and data augmentation, so as to significantly speed up *supervised* topic model training on very large datasets and very big models.

2.2 Gibbs MedLDA

Gibbs MedLDA Zhu et al. (2014) is a hybrid generative/discriminative model that jointly learns the latent topic representations (unsupervised) and large-margin classifiers for enhanced prediction (supervised). To set up the model, let $\mathcal{V} = \{1, \ldots, V\}$ index the V words in our vocabulary, and $\mathcal{D} = \{(\mathbf{w}_d, y_d)\}_{d=1}^D$ be the labeled training set, where $\mathbf{w}_d = \{w_{di}\}_{i=1}^{N_d}$ is the set of tokens appearing in document d, *i.e.*, each $w_{di} \in \mathcal{V}$. For ease of presentation, $y_d \in \mathcal{Y} = \{-1, +1\}$ indicates the (binary) label of document d. We will consider the multi-class setting later.

Gibbs MedLDA consists of two components: a latent Dirichlet allocation (LDA) Blei et al. (2003) likelihood model that describes the input documents $\mathbf{W} = \{\mathbf{w}_d\}_{d=1}^D$, and a stochastic classifier that takes supervising signal $\mathbf{y} = \{y_d\}_{d=1}^D$ into account. Specifically, LDA Blei et al. (2003) posits each document as an admixture of K topics, where each topic $\boldsymbol{\Phi}_k, k = 1, \ldots, K$, represents a multinomial distribution over the V words. The generative process of the d-th document proceeds as:

1. Draw topic mixing coefficients $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$;
2. For each position $i = 1, \ldots, N_d$, in the document:
   (a) Draw topic assignment $z_{di} \sim \mathrm{Mult}(\boldsymbol{\theta}_d)$;
   (b) Draw token $w_{di} \sim \mathrm{Mult}(\boldsymbol{\Phi}_{z_{di}})$;

where $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution with the hyperparameter $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ controlling its shape, $\mathrm{Mult}(\cdot)$ is the single-trial multinomial distribution, and $\boldsymbol{\Phi}_{z_{di}}$ denotes the topic indexed by the current topic assignment z_{di}. In a fully Bayesian treatment, topics themselves are considered as random variables and assumed to be generated from the conjugate prior, *i.e.*, for all k, $\boldsymbol{\Phi}_k \sim \mathrm{Dir}(\boldsymbol{\beta})$. Throughout we denote $\bar{N} = \frac{1}{D}\sum_{d=1}^D N_d$ as the average number of tokens appearing in documents.

Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_d\}_{d=1}^D$ be the set of topic proportions and $\mathbf{Z} = \{\mathbf{z}_d\}_{d=1}^D$ be the set of topic assignments, where $\mathbf{z}_d = \{z_{di}\}_{i=1}^{N_d}$ represents the topic assignments in document d. Since only the tokens $\mathbf{W}$ are observed, LDA infers the posterior distribution for other *unobserved* latent variables:

$$p(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}|\mathbf{W}) \propto p_0(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) p(\mathbf{W}|\mathbf{Z}, \boldsymbol{\Phi}), \tag{4}$$

where $p_0(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) = p_0(\mathbf{Z}|\boldsymbol{\Theta}) p_0(\boldsymbol{\Theta}|\boldsymbol{\alpha}) p_0(\boldsymbol{\Phi}|\boldsymbol{\beta})$ is the prior distribution and $p(\mathbf{W}|\mathbf{Z}, \boldsymbol{\Phi})$ stands for the multinomial likelihood described above. Clearly, the posterior distribution is the unique solution of the following variational problem:

$$\underset{q}{\text{minimize}} \quad \mathrm{KL}\left[q(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) \| p(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}|\mathbf{W})\right], \tag{5}$$

where, and in the following, the minimization is performed w.r.t. all probability densities, and $\mathrm{KL}[p\|q]$ measures the Kullback-Leibler divergence between density p and q. Trivial as it is, the

variational form of Bayesian inference makes it possible to add regularizations to the posterior. Using this idea, Gibbs MedLDA incorporates a stochastic classifier, represented as the random variable $\boldsymbol{\eta}$, to the objective (5):

$$\underset{q}{\text{minimize}} \quad \mathcal{L}\big(q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})\big) + 2\lambda \cdot \mathcal{R}\big(q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi})\big), \tag{6}$$

where

$$\mathcal{L}(q) = \mathrm{KL}\left[q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) \| p(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi} | \mathbf{W})\right]$$

is the term in (5), and $\mathcal{R}(q) = \sum_{d=1}^{D} \mathbb{E}_q\left[(1 - y_d f(\boldsymbol{\eta}, \mathbf{z}_d))_+\right]$ is the expected hinge loss induced by the *stochastic* linear discriminant function[1] $f(\boldsymbol{\eta}, \mathbf{z}_d) = \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d$ built on normalized topic counts $\bar{\mathbf{z}}_d = \frac{1}{N_d} \sum_{i=1}^{N_d} z_{di}$. Lastly, λ is the regularization constant that balances the two objectives in (6).

The regularizer $\mathcal{R}$ in (6) couples the (unsupervised) latent representation $\mathbf{Z}$ with the (supervised) classifier $\boldsymbol{\eta}$, leading to more pronounced prediction power. Importantly, since $\mathcal{R}$ is simply a linear functional of the posterior distribution, we can still derive a closed-form solution from (6):

$$q(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}) \propto p(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi} | \mathbf{W}) \cdot \boldsymbol{\phi}(\mathbf{y} | \mathbf{Z}, \boldsymbol{\eta}), \tag{7}$$

where $p(\cdot)$ is the usual posterior in (4), and

$$\boldsymbol{\phi}(\mathbf{y} | \mathbf{Z}, \boldsymbol{\eta}) \propto \prod_{d=1}^{D} \exp\big(-2\lambda \cdot \max\{\zeta_d, 0\}\big) \tag{8}$$

$$\zeta_d = 1 - y_d \cdot f(\boldsymbol{\eta}, \mathbf{z}_d) = 1 - y_d \cdot \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d \tag{9}$$

is the extra psudo-likelihood term induced by the regularizer $\mathcal{R}$. The inference of the latent variables $\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi}$ consists of repeatedly drawing samples from the (regularized) posterior density (7). The key insight, originated from Griffiths and Steyvers (2004) for LDA, is that the usual posterior $p(\boldsymbol{\eta}, \boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{\Phi} | \mathbf{W})$ can be efficiently sampled using the Gibbs sampler mentioned in §2.1. For Gibbs MedLDA, the extra term $\boldsymbol{\phi}(\cdot)$ in (8) creates additional difficulty: neither itself or its conditional given all other variables can be easily sampled. Fortunately, as shown in Polson and Scott (2011), using data augmentation with an extra scale random variable $\boldsymbol{\xi}$, the pseudo-likelihood can be written as the marginal of the scale-mixture of normal densities:

$$\boldsymbol{\phi}(\mathbf{y}, \boldsymbol{\xi} | \mathbf{Z}, \boldsymbol{\eta}) \propto \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\xi_d^3}} \exp\left(-\frac{(1 + \lambda\zeta_d\xi_d)^2}{2\xi_d}\right), \tag{10}$$

where recall that ζ_d is defined in (9). The conditionals of the augmented density can then be easily sampled (more details below). Overall, the state-of-the-art implementation in Zhu et al. (2014, 2013) costs $O(K^3 + DK^2 + D\bar{N}K)$ for an entire cycle of Gibbs sampling. We will significantly bring down this complexity to $O(DK + D\bar{N})$, based on a careful combination of MCMC techniques and recent fast sampling algorithms for LDA, which we briefly review next.

[1] In contrast, the original MedLDA in Zhu et al. (2012) considered the *expected* linear discriminant function $\sum_{d=1}^{D}(1 - y_d \mathbb{E}_q[\bar{\mathbf{z}}_d^\top \boldsymbol{\eta}])_+$, which, unfortunately, is computationally more challenging. By Jensen's inequality, it is clear that the objective of Gibbs MedLDA upper bounds that of MedLDA.

2.3 Previous Work on Fast Sampling for LDA

The original LDA formulation Blei et al. (2003) was solved using variational inference, under restrictive mean field assumptions. Later on Griffiths and Steyvers (2004) provided the first efficient sampling method, which largely boosts the interest in topic models. More precisely, Griffiths and Steyvers (2004) noted that the latent variables $\mathbf{\Theta}$ and $\mathbf{\Phi}$, due to conjugacy, can be analytically integrated out, leaving only the topic assignment $\mathbf{Z}$. Then the (collapsed) Gibbs sampler can be efficiently applied, leading to the (conditional) multinomial distribution[2]:

$$p(z_{di} = k|\text{rest}) \propto (n_{kd}^{-di} + \alpha_k) \cdot \frac{\mathsf{n}_{kw}^{-di} + \beta_w}{(\mathsf{n}_k^{-di} + \bar{\beta}V)}, \tag{11}$$

where n_{kd} counts the number of tokens in document d that are assigned to topic k, n_{kw} counts the number of word w assigned to topic k, and n_k counts the number of total words assigned to topic k. The superscript $^{-di}$ means excluding the current token from the respective counts, and $\bar{\beta} = \frac{1}{V}\sum_{w=1}^{V} \beta_w$ is the average. Directly drawing from the multinomial (11) costs $O(K)$. This is costly when K is large and a lot of recent work has tried to improve it.

The SparseLDA Yao et al. (2009) decomposes the multinomial in (11) into three parts in order to exploit the sparsity in the topic counts n_{kd} and n_{kw}, *i.e.*, only few topics appear in a certain document and only few words appear in a certain topic. A significant step is taken in AliasLDA Li et al. (2014) towards a constant sampling cost. It used the independent MH algorithm with a proposal consisting of two parts: the first part, essentially $p_w(k)$ in (12) below, can be constructed using the alias table Walker (1977) in $O(K)$ time, and the second part exploits sparsity in n_{kd} hence has smaller complexity than $O(K)$. Since the first part, the word proposal $p_w(k)$, is shared by all documents, it can be re-used K times, leading to the amortized $O(1)$ complexity. Overall the complexity is dominated by the average number of topics appearing in any document: smaller than $O(K)$ but still bigger than $O(1)$. Finally, the recent LightLDA Yuan et al. (2015) was able to achieve $O(1)$ complexity, based on the composition principle mentioned in §2.1. In words, it considered the following factorized proposal in MH:

$$q(z_{di} = k|\text{rest}) \propto \underbrace{(n_{kd} + \alpha_k)}_{=p_d(k)} \times \underbrace{\frac{(\mathsf{n}_{kw} + \beta_w)}{(\mathsf{n}_k + \bar{\beta}V)}}_{=p_w(k)}. \tag{12}$$

As in AliasLDA Li et al. (2014), the word proposal $p_w(k)$ is shared by all documents hence can be sampled using alias table in amortized $O(1)$ time. The document proposal $p_d(k)$ is local to each document, but can be sampled almost for free: simply picking a random token in document d and using its topic assignment takes care of the n_{kd} term. The constant term α_k has little influence on the effect and efficiency of the sampling procedure. Using the composition principle (*c.f.* Theorem 1), LightLDA *cyclically* samples from the word proposal and the document proposal, and achieves amortized $O(1)$ complexity.

We mention that another line of work tries to scale up LDA by parallelization, see *e.g.* Newman et al. (2009); Ahmed et al. (2012); Yuan et al. (2015). Conceivably our sampling algorithm for Gibbs MedLDA below can also be parallelized, and will be investigated in our future work.

[2]The counts n_{kd} for topic-document pairs and n_{kw} for topic-word pairs are different objects, hence we use slightly different fonts for them to reduce confusion.

3 Lightweight Gibbs MedLDA

As mentioned above, the state-of-the-art implementation of Gibbs MedLDA in Zhu et al. (2014) costs $O(K^3+DK^2+D\bar{N}K)$ in a full cycle. The superlinear dependence on K, the number of topics, prevents Gibbs MedLDA from scaling to large text corpus where a moderately large K is needed to catch the long tail behavior. Moreover, a larger K, resulting in more latent features, may also be beneficial for training the large-margin classifier. Considering the excellent discriminative power of Gibbs MedLDA and the recent impressive advances for LDA, it is thus very desirable to develop a fast *linear time* sampling algorithm for the former as well. We provide such an algorithm in this section, effectively reducing the complexity to $O(DK + D\bar{N})$, which is clearly the best possible. As we demonstrate in the experiments (§4), this improvement is already significant for K around a few tens to hundreds.

3.1 The Regularized Posterior: Recalled

For ease of reference, let us first recall the regularized posterior density in Gibbs MedLDA (*c.f.* §2.2):

$$q(\boldsymbol{\eta}, \boldsymbol{\xi}, \mathbf{Z}|\mathbf{W}) \propto p_0(\boldsymbol{\eta}) \left[\prod_{d=1}^{D} \frac{\mathsf{B}(n_{\cdot d} + \boldsymbol{\alpha})}{\mathsf{B}(\boldsymbol{\alpha})}\right] \prod_{k=1}^{K} \frac{\mathsf{B}(\mathsf{n}_{k\cdot} + \boldsymbol{\beta})}{\mathsf{B}(\boldsymbol{\beta})} \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\xi_d^3}} \exp\left(-\frac{(1+\lambda\zeta_d\xi_d)^2}{2\xi_d}\right), \quad (13)$$

where $\mathsf{B}(\cdot)$ is the multivariate Beta function, n_{kd} is the number of tokens in document d assigned to topic k, $n_{\cdot d} = \{n_{kd}\}_{k=1}^K$ is the topic counts of document d, n_{kw} is the number of word w assigned to topic k, and $\mathsf{n}_{k\cdot} = \{\mathsf{n}_{kw}\}_{w=1}^V$ is the word counts of topic k. Following Griffiths and Steyvers (2004) we have used conjugacy to analytically integrate out the topic mixing coefficients $\boldsymbol{\Theta}$ and topic distribution $\boldsymbol{\Phi}$. The augmented variable $\boldsymbol{\xi}$ is introduced to help sampling the conditional density of $\boldsymbol{\eta}$, the large-margin classifier. Recall from (9) that $\zeta_d = 1 - y_d\boldsymbol{\eta}^\top \bar{\mathbf{z}}_d$ represents the margin of the classifier on document d. Lastly, we impose the normal distribution prior $p_0(\boldsymbol{\eta}) = \prod_{k=1}^K \mathcal{N}(\eta_k; 0, \nu^{-1})$ on the classifier $\boldsymbol{\eta}$.

As in Zhu et al. (2014), we will use the Gibbs sampler mentioned in §2.1 to sample from the posterior $q(\cdot)$. The individual sampling steps for each conditional density, with substantial improvements upon Zhu et al. (2014), are detailed in the next three subsections.

3.2 Sampling the Augmented Variable $\boldsymbol{\xi}$

Recall that $\boldsymbol{\xi}$ is augmented, as "virtual data", to help sampling the classifier $\boldsymbol{\eta}$ below. Its conditional density, given both $\mathbf{Z}$ and $\boldsymbol{\eta}$, factorizes among documents with each coordinate following the inverse Gaussian distribution:

$$p(\xi_d|\mathbf{Z}, \boldsymbol{\eta}) \propto \frac{1}{\sqrt{2\pi\xi_d^3}} \exp\left(-\frac{(1+\lambda|\zeta_d|\cdot\xi_d)^2}{2\xi_d}\right), \quad (14)$$

whose mean and shape parameters are respectively $\frac{1}{\lambda|\zeta_d|}$ and 1. Using the root splitting technique in Michael et al. (1976) we can draw from the inverse Gaussian distribution in $O(1)$ time. This step is the same as in Zhu et al. (2014), and costs in total $O(D)$ time.

3.3 Sampling the Topic Assignment Z

This part differs substantially from Zhu et al. (2014) and consists of the first key component towards the claimed linear time sampling algorithm. Writing out the conditional density again:

$$p(z_{di} = k|\text{rest}) \propto (n_{kd}^{-di} + \alpha_k) \cdot \frac{\mathsf{n}_{kw}^{-di} + \beta_w}{\mathsf{n}_k^{-di} + \bar{\beta} V} \cdot \exp\left(g_d(\eta_k)\right), \tag{15}$$

where the following definitions are adopted throughout:

$$g_d(\eta_k) = \lambda \frac{y_d(1 + \lambda \xi_d)\eta_k}{N_d} - \lambda^2 \xi_d \frac{\eta_k^2 + 2\eta_k m_d^{-di}}{2N_d^2} \tag{16}$$

$$m_d^{-di} = \sum_{k=1}^{K} \eta_k n_{kd}^{-di}. \tag{17}$$

Like other counts, the classifier re-weighted count m_d^{-di} can be incrementally updated in $O(1)$ time within each document d. Directly sampling the above multinomial, as is done in Zhu et al. (2014), costs $O(K)$. Instead, inspired by the recent LightLDA Yuan et al. (2015), we turn to the independent MH (§2.1) with the ideal *factorized* proposal:

$$f(z_{di} = k|\text{rest}) \propto \underbrace{(\tilde{n}_{kd} + \alpha_k)}_{p_d(k)} \cdot \underbrace{\frac{\tilde{\mathsf{n}}_{kw} + \beta_w}{\tilde{\mathsf{n}}_k + \bar{\beta} V}}_{p_w(k)} \cdot \underbrace{\exp\left\{\tilde{g}_d(\eta_k)\right\}}_{p_e(k)}. \tag{18}$$

Note the similarity with the true conditional (15). However, directly drawing from the ideal proposal $f(\cdot)$ is still costly. The key is to "freeze" the proposal (explaining our tilde notation) so that we can amortize computation Li et al. (2014). In details, we use Walker's method Walker (1977) to build an alias table for the proposal $f(\cdot)$. This takes $O(K)$ time, but subsequent drawing from the alias table costs only $O(1)$. Thus if we recycle the alias table for $O(K)$ times the total complexity can be amortized to $O(1)$. The *independent* MH is then employed to account for the "frozen" hence obsolete proposal, leaving the stationary density unchanged. After recycling the alias table for $O(K)$ times, we rebuild it using the fresh counts.

The alias method we described above suffers from one drawback though. It involves all three counts: the topic-document pair $\tilde{n}_{kd}$, the topic-word pair $\tilde{\mathsf{n}}_{kw}$, and the classifier re-weighted count $\tilde{m}_d$. Thus during sampling whenever we switch to a different document or word, using the obsolete proposal in (18) will result in low acceptance. To address this issue, we follow the approach in LightLDA Yuan et al. (2015) to split the proposal into three parts: the doc-proposal $p_d(k)$, the word-proposal $p_w(k)$, and the exp-proposal $p_e(k)$. We build an (independent) MH Markov chain for each proposal, and use the composition principle (Theorem 1) to combine them.

Doc-proposal: The doc-proposal $p_d(k)$ can be sampled in $O(1)$ time as follows. We further split it into two parts, the $\tilde{n}_{kd}$ term and the constant α_k term. Since the sum $\tilde{n}_d := \sum_k \tilde{n}_{kd}$ can be incrementally maintained in $O(1)$ time, we first flip a coin (with bias $\tilde{n}_d / \sum_k \alpha_k$) to decide which part to sample from. For moderately large $\boldsymbol{\alpha}$, most time we will be sampling the $\tilde{n}_{kd}$ part, which is extremely simple: given the topic assignments $\mathbf{z}_d$, we only need to draw a random token in document d and use its topic assignment. For the constant term α_k, we use the alias method Walker (1977) that builds the alias table in $O(K)$ time but repeated re-cycling of the table amortizes the

complexity down to $O(1)$. Note that this alias table can even be shared between documents. The acceptance probability required in the independent MH (*c.f.* (1)), say transitioning from state s to state t, is given by

$$A_d = \min\left\{\frac{(n_{td}^{-di}+\alpha_t)(\mathsf{n}_{tw}^{-di}+\beta_w)(\mathsf{n}_s^{-di}+\bar{\beta}V)\exp(g_d(\eta_t))}{(n_{sd}^{-di}+\alpha_s)(\mathsf{n}_{sw}^{-di}+\beta_w)(\mathsf{n}_t^{-di}+\bar{\beta}V)\exp(g_d(\eta_s))}\times\frac{\tilde{n}_{sd}+\alpha_s}{\tilde{n}_{td}+\alpha_t},\quad 1\right\}, \tag{19}$$

which is easily evaluated in $O(1)$ time after incrementally bookkeeping the counts and the classifier re-weighted count m_d (*c.f.* (16)). Thus the overall sampling time for the doc-proposal is amortized to $O(1)$ per token.

Word-proposal: The word proposal $p_w(k)$ is handled using the alias method Walker (1977). We construct its alias table in $O(K)$ time but can re-use the table for drawing K samples in $O(1)$ time each. Note that the word-proposal is shared among all documents, thus even in the very unlucky case where a certain word only appears say once in document d, its alias table can still be re-used in other documents. Therefore the $O(K)$ time spent on building the table is amortized again to $O(1)$ per token. The acceptance probability

$$A_w = \min\left\{\frac{(n_{td}^{-di}+\alpha_t)(\mathsf{n}_{tw}^{-di}+\beta_w)(\mathsf{n}_s^{-di}+\bar{\beta}V)\exp(g_d(\eta_t))}{(n_{sd}^{-di}+\alpha_s)(\mathsf{n}_{sw}^{-di}+\beta_w)(\mathsf{n}_t^{-di}+\bar{\beta}V)\exp(g_d(\eta_s))}\times\frac{(\tilde{\mathsf{n}}_{sw}+\beta_s)(\tilde{\mathsf{n}}_t+\bar{\beta}V)}{(\tilde{\mathsf{n}}_{tw}+\beta_t)(\tilde{\mathsf{n}}_s+\bar{\beta}V)},\quad 1\right\} \tag{20}$$

is evaluated in $O(1)$ time similarly as that of the doc-proposal.

Exp-proposal: We use again the alias method for the exp-proposal $p_e(k)$. The key observations here are: 1). The classifier re-weighted count m_d^{-di} can be easily evaluated in $O(1)$ time after bookkeeping m_d; 2). The alias table of the exp-proposal, while local to each document, can be re-used for other tokens in the same document, therefore the $O(K)$ time spent in building the table is amortized to $O(1)$. The acceptance probability

$$A_e = \min\left\{\frac{(n_{td}^{-di}+\alpha_t)(\mathsf{n}_{tw}^{-di}+\beta_w)(\mathsf{n}_s^{-di}+\bar{\beta}V)\exp(g_d(\eta_t))}{(n_{sd}^{-di}+\alpha_s)(\mathsf{n}_{sw}^{-di}+\beta_w)(\mathsf{n}_t^{-di}+\bar{\beta}V)\exp(g_d(\eta_s))}\times\frac{\exp(\tilde{g}_d(\eta_s))}{\exp(\tilde{g}_d(\eta_t))},\quad 1\right\} \tag{21}$$

again is easily evaluated in $O(1)$ time. Note that the exponential terms above do not cancel out because the tilde terms rely on the slightly obsolete count m_d^{-di}.

Proposal composition: After having the three proposals described above constructed, we use the composition principle (*c.f.* Theorem 1) to combine them. In the experiments, we will compare the cyclic combination and the mixture combination (with equal odds). For each token, we can even iterate the composed transition kernel for a small number of times (say 3). The overall time is $O(DK + D\bar{N})$, where the first factor comes from building the alias table in each document and the second factor is simply the number of tokens we must process in each full cycle. We remind that although each proposal only takes care of a part of the full conditional, its acceptance probability in MH restores stationarity, that is, we never alter the stationary density. Thus after burn-in we are still sampling the true (regularized) posterior. This well illustrates the flexibility of MCMC and is the key to achieve linear time sampling here.

3.4 Sampling the Classifier $\boldsymbol{\eta}$

Lastly we show how to sample the classifier weight $\boldsymbol{\eta}$ again in linear time. Since we assume isotropic Gaussian prior $p_0(\boldsymbol{\eta}) = \mathcal{N}\left(\boldsymbol{\eta}; \mathbf{0}, \nu^{-1}\mathbf{I}\right)$, the conditional density of $\boldsymbol{\eta}$, given all other latent variables,

is again Gaussian:

$$\boldsymbol{\eta}|\mathbf{Z}, \boldsymbol{\xi} \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}, \boldsymbol{\Xi}^{-1}\right), \tag{22}$$

where the posterior mean $\tilde{\boldsymbol{\mu}} = \boldsymbol{\Xi}^{-1}\overline{\mathbf{Z}}\mathbf{u}$ and the precision matrix $\boldsymbol{\Xi} = \nu\mathbf{I} + \lambda^2 \sum_d \xi_d \bar{\mathbf{z}}_d \bar{\mathbf{z}}_d^\top$, where $\mathbf{u} \in \mathbb{R}^D$ with the d-th entry $u_d = \lambda y_d(1 + \lambda\xi_d)$. Note that forming the precision matrix $\boldsymbol{\Xi}$ costs $O(K^2D)$; inverting it to get the covariance matrix costs $O(K^3)$; and sampling the Gaussian with the covariance matrix costs $O(K^3)$. This is the approach used in Zhu et al. (2014), which is fine at the time since sampling the topic assignment in Zhu et al. (2014) costs already $O(D\bar{N}K)$, dominating the overall cost. Since we have successfully reduced the latter complexity to $O(DK + D\bar{N})$ in §3.3, the brute-force $O(DK^2 + K^3)$ cost for sampling the classifier can no longer be neglected. In fact, we verified in our experiments that this step starts to dominate the training time even for moderately large K. Therefore, we need a faster sampling algorithm for the classifier part.

The idea is to use the Gibbs sampler. Indeed, we have the following (univariate) conditional normal density:

$$\eta_k|\text{rest} \;\sim\; \mathcal{N}(\tau_k^{-1}\mu_k, \tau_k^{-1}), \tag{23}$$

where the (unnormalized) mean

$$\mu_k = \sum_d \bar{\mathbf{z}}_{dk}\Big(u_d - \lambda^2\xi_d \sum_{j\neq k} \bar{\mathbf{z}}_{dj}\eta_j\Big) \tag{24}$$

and the precision $\tau_k = \nu + \lambda^2 \sum_d \xi_d \bar{\mathbf{z}}_{dk}^2$. The key observation here is that both the mean *vector* $\boldsymbol{\mu}$ and the precision *vector* $\boldsymbol{\tau}$ can be computed in $O(KD)$ time by proper bookkeeping. Note also that we completely bypass the need of forming the precision matrix $\boldsymbol{\Xi}$. After having the mean and precision, drawing each univariate η_k costs $O(1)$ time. Therefore, a full iteration of all K weights costs $O(KD)$. We point out that there is no need to iterate the Gibbs sampler here many times: even a single iteration would still preserve the stationary density. This very flexibility of MCMC is the key to obtain our linear time sampling algorithm, without altering the target posterior density at all.

3.5 Collecting the Pieces

We now have all ingredients for our linear time sampling algorithm for the Gibbs MedLDA model defined in §2.2: We cycle through the three components presented in the above subsections: sampling augmented variable $\boldsymbol{\xi}$ in §3.2, sampling topic assignment variable $\mathbf{Z}$ in §3.3, and sampling classifier variable $\boldsymbol{\eta}$ in §3.4. We repeat the procedure M times for burn-in, after which new samples $\mathbf{Z}$ and $\boldsymbol{\eta}$ can be regarded as true samples from the regularized posterior. The augmented variable $\boldsymbol{\xi}$ is simply discarded. As promised, the overall time is $O(DK + D\bar{N})$, a significant improvement over the current state-of-the-art Zhu et al. (2014). We point out that our improvement on sampling complexity is obtained by potentially slowing down the mixing rate of the underlying Markov chain—the point, nevertheless, is that through a more delicate balance between the two costs we can achieve greater efficiency.

Testing: For inference on the *test* data, we follow the same procedure as in Zhu et al. (2014). First, we infer the topic distribution using the point estimate: $\hat{\boldsymbol{\Phi}}_{kw} \propto \mathsf{n}_{kw} + \beta_w$. Then, given a test document $\mathbf{w}$, we infer its latent topic assignment $\mathbf{z}$ by drawing samples from the conditional density:

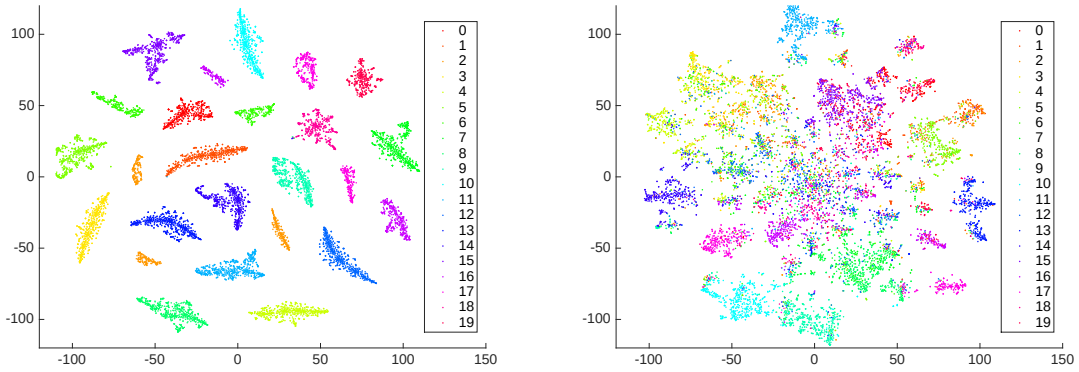

Figure 1: The 2D embedding of LightMedLDA (left) and unsupervised LDA (right) with $K = 100$ at iteration 50. Each point is the latent representation of a document and each color corresponds to a label. We can observe that supervised model is much better in learning discriminative latent structures than unsupervised model.

$p(z_i = k|\text{rest}) \propto \hat{\mathbf{\Phi}}_{kw_i}(n_k^{-i} + \alpha_k)$. This Gibbs sampling procedure is repeated until some convergence criteria is met (*e.g.* the relative change of the data likelihood falls below some threshold). Finally we apply the classifier $\boldsymbol{\eta}$ (sampled during training period) on the averaged topic assignment $\bar{\mathbf{z}}_d$ to make prediction: $\hat{y} = \text{sign}(\boldsymbol{\eta}^\top \bar{\mathbf{z}}_d)$. In practice, we keep a few samples of $\boldsymbol{\eta}$ and use their average to predict. This usually leads to more robust performance.

4 Experiments

We now present empirical results to verify the efficiency of the proposed linear time sampling algorithm, referred as LightMedLDA. Its C++ implementation is available at https://github.com/xunzheng/light_medlda. We will focus on comparing against the current state-of-the-art implementation in Zhu et al. (2014), referred as GibbsMedLDA. As shown previously in Zhu et al. (2014), GibbsMedLDA outperforms most existing supervised topic models, and we refer the interested readers to Zhu et al. (2014) for detailed comparisons.

Datasets: We conduct experiments on three benchmark datasets: the 20Newsgroups[3] for binary and multi-task classification and a Wikipedia dataset[4] with 1.1 million documents for multi-label classification. See Table 1 for their summary statistics.

Setup: For all experiments, if not mentioned explicitly, the tuning parameters are set as follows: the regularization constant $\lambda = 102.4$; the number of inner MH steps for sampling the topic assignment (§3.3) $S_{mh} = 6$ (*i.e.*, applying each proposal twice); the number of Gibbs sampling sub-iterations for sampling the classifier (§3.4) $S_{gibbs} = 2$. We use symmetric Dirichlet priors with hyper-parameter $\alpha_k \equiv 6.4/K, \beta_w \equiv 0.01$. Experimental results are averaged over multiple runs with the standard deviation provided. Except for the large Wikipedia dataset, performance is measured on a standard desktop with a 3.30 GHz CPU.

[3] Available at: http://qwone.com/~jason/20Newsgroups
[4] Available at: http://lshtc.iit.demokritos.gr/

	# train	# test	# word	type
20NG	11,269	7,505	61,188	multi-class
Wiki	1,100,000	5,000	917,683	multi-label

Table 1: Summary statistics for the benchmark datasets. Both 20NG and Wiki have 20 classes.

4.1 Qualitative Evaluation

We run LightMedLDA and unsupervised LDA on the full 20Newsgroups data set with 100 topics. Figure 1 shows the 2D embedding of the latent representations at the 50th iteration. Each data point in the figure is corresponds to a topical representation, *i.e.*, a 100 dimensional vector of a document, while each color corresponds to one of the class label out of a total of 20. The embedding was computed using an open source toolkit t-SNE (Van der Maaten and Hinton, 2008). If the colors are well-separated, the learned topics are more discriminative, which is desirable both for interpretation and for downstream tasks such as classification. By comparing two figures, we can easily observe that with supervised method one can learn a more discriminative latent structure of the text data.

4.2 Binary Classification

We pick two subgroups *alt.atheism* and *talk.religion.misc* from the 20Newsgroups dataset to form a binary classification task. This sub-dataset consists of 856 documents for training and 569 documents for testing. Similar to the settings in Zhu et al. (2014), we set $\lambda = 262.4$ and the number of burn-in steps to $M = 10$. As shown in Figure 2 (left), when we vary the number of latent topics K from 10 to 100, both LightMedLDA and GibbsMedLDA achieved consistent prediction accuracies around 80% on the test set, with slightly better performance from LightMedLDA. This is expected since both algorithms are solving the same problem (whereas the slight difference may be caused by different convergence speed). Shown on the right panel of Figure 2 are the training times of LightMedLDA and GibbsMedLDA. We observe that even on this small sub-dataset, the training time of GibbsMedLDA increased sharply w.r.t. the number of latent topics (x-axis). This confirms the superlinear dependence of GibbsMedLDA's complexity on the model size. In contrast, LightMedLDA converged much faster, and kept the training time under 1s even for 100 topics (the largest we tried on this small dataset).

We then tried classifying all 20 classes on the full 20Newsgroups dataset. We used the one-vs-all strategy to train 20 separate binary classifiers. The results are shown in Figure 3, along with the multi-task results described in the next subsection. On the left we see that again LightMedLDA and GibbsMedLDA achieved similar accuracies, with slightly better performance for LightMedLDA (due possibly to its faster convergence). We observe that the classification accuracy starts to decrease once the number of topics exceeds 150. This can be explained by: First, for larger K both algorithms need more iterations to converge while we capped the number of iterations to $M = 25$ and $M = 20$ for LightMedLDA and GibbsMedLDA, respectively; Second, the algorithms may start to overfit when K is large. On the right of Figure 3 we observe similar behavior of the training time when we vary the number of topics: GibbsMedLDA increases sharply due to its superlinear dependence on K while LightMedLDA is only slightly affected even when $K = 400$ (the largest we tried on the full dataset).

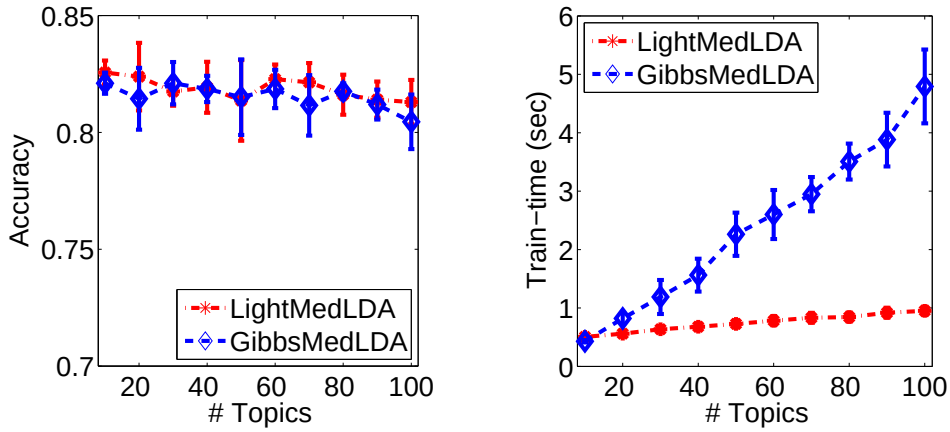

Figure 2: Classification accuracy (left) and training time (right) of LightMedLDA and GibbsMedLDA on the binary 20Newsgroups sub-dataset. We can observe that although the classification accuracy remains similar, our new algorithm scales nicely with increasing model size.

4.3 Multi-task Classification

In this subsection we test the multi-task formulation again on the full 20Newsgroups dataset. We train all 20 classifiers simultaneously on the *same* latent representation. The results are shown in Figure 3, along with the previous one-vs-all results for comparison. The conclusions are similar to the one-vs-all setting: Both LightMedLDA and GibbsMedLDA achieved similar accuracies on the test set, but LightMedLDA is much faster in terms of training time, in particular when the number of topics is large. From the right panel it is also clear that the multi-task formulation took significantly less time than the one-vs-all strategy. This is not surprising as the one-vs-all essentially repeats the computation 20 times (one for each separate classifier). Note that we did not explore parallelization in this work.

We further performed a multi-labeled prediction task on the massive Wikipedia dataset that has 1.1 million documents. Due to the multi-label nature we used the F1-measure (the harmonic mean of the precision and recall) to evaluate the performance. We only considered the multi-task formulation for this dataset as the one-vs-all strategy would take too long. We set the number of inner Gibbs steps $S_{gibbs} = 4$ (for drawing classifiers, see §3.4) and the number of burn-in steps $M = 80$ for LightMedLDA and $M = 40$ for GibbsMedLDA. The larger number of burn-in steps is caused by the large size of this dataset. The results are shown in Figure 4, from which we conclude that both algorithms consistently achieved the F1-measure around 0.55 while LightMedLDA converges substantially faster. With 200 topics, GibbsMedLDA already took 33 hours on this large dataset while LightMedLDA, with 400 topics, took only 11 hours. GibbsMedLDA with 400 topics was too slow to converge.

5 Conclusion

Topic models such as LDA are excellent tools in processing large collections of unstructured data, and a lot of recent work has devoted to scaling them to large industrial data and big models. Building on these recent sampling advances for *unsupervised* LDA formulations, we have presented

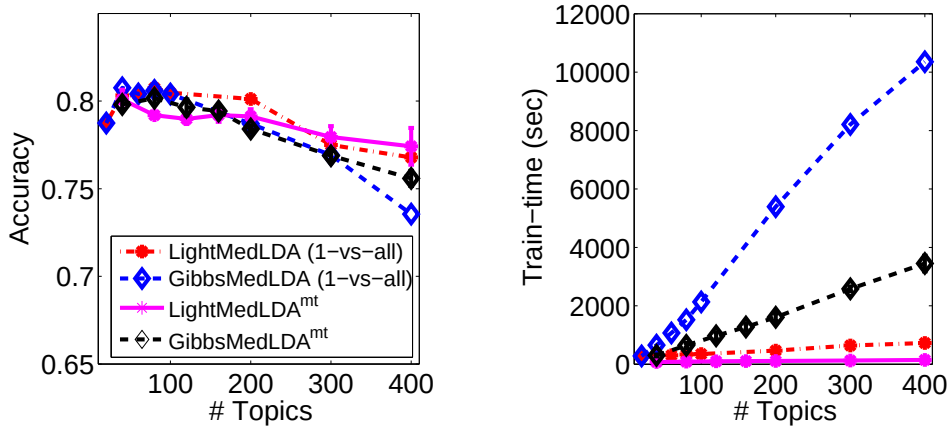

Figure 3: Classification accuracy and training time of LightMedLDA, GibbsMedLDA, multi-task LightMedLDA and multi-task GibbsMedLDA on the full 20Newsgroups dataset. One-vs-all strategy is used for binary classifiers. We can again observe that while the new algorithm has similar accuracy as exact Gibbs sampling counterparts, it is significantly more efficient, especially for large models.

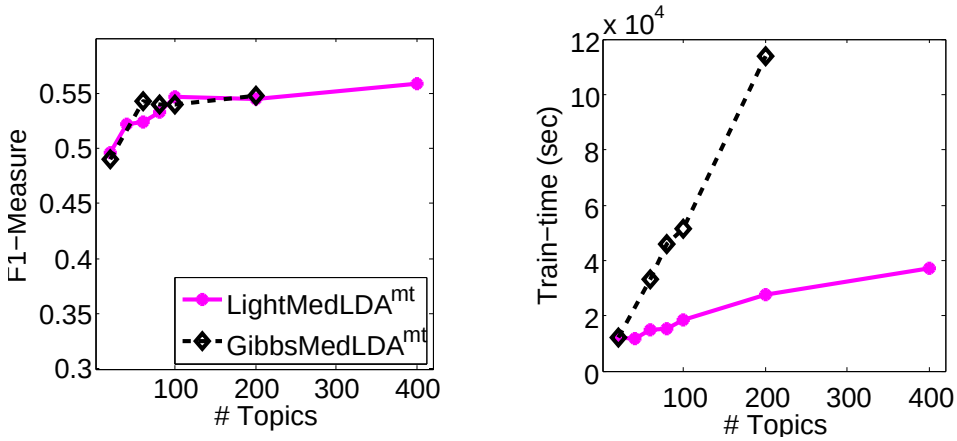

Figure 4: F1-measure and training time of LightMedLDA^{mt} and GibbsMedLDA^{mt} on the multi-labeled Wikipedia dataset. They both achieve about 0.55 F-1 score, however GibbsMedLDA already becomes impractical for $K = 200$.

the first linear time sampling algorithm for the *supervised* topic model, Gibbs MedLDA, that can exploit the large supervision information to achieve better predictions. Our algorithm easily extends to a variety of losses in binary classification, multi-task learning, multi-label classification and regression, and we observed in our experiments an order of magnitude speedup over the current state-of-the-art implementation, while obtaining comparable accuracy. For future work we plan to explore nonparametric extensions and parallel implementations.

6 Acknowledgements

This work was supported by NIH Grants R01GM087694 and P30DA035778, and NSF Grant IIS1447676. This is a joint work with Yaoliang Yu, who provided valuable insights and enormous help throughout the study.

References

Ahmed, A., Aly, M., Gonzalez, J., Narayanamurthy, S., and Smola, A. J. (2012). Scalable inference in latent variable models. In *WSDM*, pages 123–132. 7

Blei, D. M. and McAuliffe, J. D. (2007). Supervised topic models. In *NIPS*, pages 121–128. 2

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022. 2, 5, 7

Chen, W.-Y., Zhang, D., and Chang, E. Y. (2008). Combinational collaborative filtering for personalized community recommendation. In *KDD*, pages 115–123. 2

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741. 4

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of National Academy of Science*, 101:5228–5235. 2, 6, 7, 8

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109. 2, 3

Li, A. Q., Ahmed, A., Ravi, S., and Smola, A. J. (2014). Reducing the sampling complexity of topic models. In *KDD*, pages 891–900. 2, 3, 7, 9

Li, F.-F. and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, pages 524–531. 2

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092. 2, 3

Michael, J. R., Schucany, W. R., and Haas, R. W. (1976). Generating random variates using transformations with multiple roots. *The American Statistician*, 30:88–90. 8

Newman, D., Asuncion, A., Smyth, P., and Welling, M. (2009). Sampling-based approaches to calculating marginal densities. *Journal of Machine Learning Research*, 10:1801–1828. 7

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

Pritchard, J. K., Stephens, M., and Donnelly, P. (2000). Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959. 2

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation (with discussions). *Journal of the American Statistical Association*, 82(398):528–550. 4

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussions). *Annals of Statistics*, 22:1701–1728. 4

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85. 13

Walker, A. J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3:253–256. 2, 7, 9, 10

Yao, L., Mimno, D., and McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. In *KDD*. 2, 7

Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E. P., Liu, T.-Y., and Ma, W.-Y. (2015). LightLDA: Big topic models on modest compute clusters. In *WWW*. 2, 3, 7, 9

Zhu, J., Ahmed, A., and Xing, E. P. (2012). MedLDA: maximum margin supervised topic models. *Journal of Machine Learning Research*, 13:2237–2278. 2, 6

Zhu, J., Chen, N., Perkins, H., and Zhang, B. (2014). Gibbs max-margin topic models with data augmentation. *Journal of Machine Learning Research*, 15:1073–1110. 2, 5, 6, 8, 9, 11, 12, 13

Zhu, J., Zheng, X., Zhou, L., and Zhang, B. (2013). Scalable inference in max-margin topic models. In *KDD*. 2, 6