Linear Time Samplers for Supervised Topic Models using Compositional Proposals

Xun Zheng
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, USA
xunzheng@cs.cmu.edu

Yaoliang Yu
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, USA
yaoliang@cs.cmu.edu

Eric P. Xing
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA, USA
epxing@cs.cmu.edu

ABSTRACT

Topic models are effective probabilistic tools for processing large collections of unstructured data. With the exponential growth of modern industrial data, and consequentially also with our ambition to explore much bigger models, there is a real pressing need to significantly scale up topic modeling algorithms, which has been taken up in lots of previous works, culminating in the recent fast Markov chain Monte Carlo sampling algorithms in [10, 22] for the *unsupervised* latent Dirichlet allocation (LDA) formulations.

In this work we extend the recent sampling advances for unsupervised LDA models to *supervised* tasks. We focus on the Gibbs MedLDA model [26] that is able to simultaneously discover latent structures and make accurate predictions. By combining a set of sampling techniques we are able to reduce the $O(K^3 + DK^2 + D\bar{N}K)$ complexity in [26] to $O(DK + D\bar{N})$ when there are K topics and D documents with average length $\bar{N}$. To our best knowledge, this is the first linear time sampling algorithm for supervised topic models. Our algorithm requires minimal modifications to incorporate most loss functions in a variety of supervised tasks, and we observe in our experiments an order of magnitude speedup over the current state-of-the-art implementation, while achieving similar prediction performances.

The open-source C++ implementation of the proposed algorithm is available at https://github.com/xunzheng/light_medlda.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation, Performance

Keywords

Inference; MCMC; Topic Models; Large Margin Classification; Regression; Scale Mixtures

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
KDD'15, August 11-14, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783371.

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 [4, 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 [17], image analysis [11], text mining [4, 8], collaborative filtering [5], prediction tasks [3, 23], and many more.

One significant challenge we face when applying topic models on real industrial data is the scalability issue. On one hand, technology innovations have made it possible to collect very large amount of industrial scale data at 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), and 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 [4], soon [8] 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 [21] further reduces the sampling cost of [8]. Another significant contribution is the AliasLDA [10], which made explicit the crucial insight that model parameters only change slowly during sampling. By a masterful combination of the independent Metropolis-Hastings algorithm [9, 12] with Walker's alias method [20] 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 [22] 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 [24], 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.* 3, 23], 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 [24, 26], 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 [10, 22] from the unsupervised LDA to *supervised* tasks. Throughout we use the Gibbs MedLDA as a running example to demonstrate the main ideas, for a). Gibbs MedLDA has already been shown to be very effective in various supervised tasks [24]; b). Gibbs MedLDA is very flexible, allowing a unified treatment of binary classification, multi-task learning, multi-label classification, regression, *etc.*; c). 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 [22] 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 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). Building on the classical result on scale-mixtures, we present a unified treatment of all common loss functions in supervised tasks. With minimal modifications this results in the fastest sampling algorithm for a variety of losses in classification, multi-task learning, regression, *etc.* 5). 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.

Paper Outline.

We first collect some background material on MCMC sampling in §2.1 for later use. Then in §2.2 we 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. Extensions and experiments are given in §4 and §5, respectively. Finally, we conclude in §6.

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 [9, 12] 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

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

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)

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

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.* [19]). *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 [7]: 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 [18]: 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 [24] 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 extend to the multi-class and regression setting in §4.

Gibbs MedLDA consists of two components: a latent Dirichlet allocation (LDA) [4] 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 [4] posits each document as an admixture of K topics, where each topic $\mathbf{\Phi}_k, k = 1, \ldots, K$, 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 $w_{di} \sim \mathrm{Mult}(\mathbf{\Phi}_{z_{di}})$;

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 $\mathbf{\Phi}_{z_{di}}$ 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, $\mathbf{\Phi}_k \sim \mathrm{Dir}(\boldsymbol{\beta})$. Throughout we denote $\bar{N} = \frac{1}{D}\sum_{d=1}^D N_d$ as the average number of tokens appearing in documents.

Let $\mathbf{\Theta} = \{\boldsymbol{\theta}_d\}_{d=1}^D$ 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:

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

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

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

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):

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

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

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

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 $\boldsymbol{\eta}, \mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}$ consists of repeatedly drawing samples from the (regularized) posterior density (7). The key insight, originated from [8] for LDA, is that the usual posterior $p(\boldsymbol{\eta}, \mathbf{\Theta}, \mathbf{Z}, \mathbf{\Phi}|\mathbf{W})$ 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 [15], 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

[1] In contrast, the original MedLDA in [23] 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.

in [24, 26] 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.

2.3 Previous Work on Fast Sampling for LDA

The original LDA formulation [4] was solved using variational inference, under restrictive mean field assumptions. Later on [8] provided the first efficient sampling method, which largely boosts the interest in topic models. More precisely, [8] noted that the latent variables $\boldsymbol{\Theta}$ and $\boldsymbol{\Phi}$, 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 [21] 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 [10] 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 [20] 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 [22] 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 [10], 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.

[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.

We mention that another line of work tries to scale up LDA by parallelization, see *e.g.* [1, 14, 22]. Conceivably our sampling algorithm for Gibbs MedLDA below can also be parallelized, and will be investigated in our future work.

3. LIGHTWEIGHT GIBBS MEDLDA

As mentioned above, the state-of-the-art implementation of Gibbs MedLDA in [24] 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 (§5), 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):

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

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 [8] 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 [24], 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 [24], are detailed in the next three subsections.

3.2 Sampling the Augmented Variable ξ

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:

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

whose mean and shape parameters are respectively $\frac{1}{\lambda|\zeta_d|}$ and 1. Using the root splitting technique in [13] we can draw from the inverse Gaussian distribution in $O(1)$ time. This step is the same as in [24], and costs in total $O(D)$ time.

3.3 Sampling the Topic Assignment z

This part differs substantially from [24] and consists of the first key component towards the claimed linear time sampling algorithm. Writing out the conditional density again:

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

where the following definitions are adopted throughout:

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

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

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 [24], costs $O(K)$. Instead, inspired by the recent LightLDA [22], we turn to the independent MH (§2.1) with the ideal *factorized* proposal:

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

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 [10]. In details, we use Walker's method [20] 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 [22] 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 [20] 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

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

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 [20]. 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

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

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

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

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 η

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 [24], which is fine at the time since sampling the topic assignment in [24] 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 [24]. 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 [24]. 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: $p(z_i = k|\text{rest}) \propto \hat{\boldsymbol{\Phi}}_{kw_i}(n_k^{-i} + \alpha_k)$. 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. SOME EXTENSIONS

In this section we discuss how to extend the basic binary classification setting in §3 to multi-class and regression tasks.

4.1 Multi-class/Multi-task Classification

Our algorithm easily extends to the multi-class setting where the label space $\mathcal{Y} = \{1, \ldots, C\}$. There are at least three ways. The first one is extremely simple: we use the one-vs-all (or the one-vs-one) strategy and train a separate classifier for each class while treating all other classes as negative. This results in C separate classifiers and for prediction we follow the maximally confident one:

$$\hat{y} = \underset{c=1,\ldots,C}{\text{argmax}} \; (\bar{\mathbf{z}}^c)^\top \boldsymbol{\eta}^c. \tag{25}$$

The nice part of this approach is that the C classifiers can be trained in parallel.

The second strategy is to recast the multi-class problem as an instance of multi-task learning. More specifically, we train C classifiers $\boldsymbol{\eta}^c, c = 1, \ldots, C$ on the *shared* latent topic assignment $\mathbf{Z}$. Define the regularizer for each class c:

$$\mathcal{R}^c(q) = \textstyle\sum_{d=1}^D \mathbb{E}_q((1 - y_d^c \bar{\mathbf{z}}_d^\top \boldsymbol{\eta}^c)_+), \tag{26}$$

where y_d^c is the *binary* label indicating whether the d-th document belongs to the c-th class. Note that the same topic assignment $\bar{\mathbf{z}}_d$ is shared among all classes. As before we can derive the regularized posterior in closed-form:

$$q(\boldsymbol{\eta}, \boldsymbol{\xi}, \mathbf{Z}|\mathbf{W}) \propto p_0(\boldsymbol{\eta}) \left[\prod_{d=1}^D \frac{\mathsf{B}(n_{\cdot d} + \boldsymbol{\alpha})}{\mathsf{B}(\boldsymbol{\alpha})}\right] \prod_{k=1}^K \frac{\mathsf{B}(\mathsf{n}_{k\cdot} + \boldsymbol{\beta})}{\mathsf{B}(\boldsymbol{\beta})}$$
$$\prod_{c=1}^C \prod_{d=1}^D \frac{1}{\sqrt{2\pi(\xi_d^c)^3}} \exp\left(-\frac{(1 + \lambda\zeta_d^c\xi_d^c)^2}{2\xi_d^c}\right), \tag{27}$$

where we have again integrated out the topic mixing coefficient $\boldsymbol{\Theta}$ and topic distribution $\boldsymbol{\Phi}$, and introduced the augment variable $\boldsymbol{\xi}^c$ for each class to ease sampling its classifier $\boldsymbol{\eta}^c$. Using the product prior $p_0(\boldsymbol{\eta}) = \prod_{c=1}^C \prod_{k=1}^K \mathcal{N}(\eta_k^c; 0, 1/\nu^c)$ we can derive the fast sampling algorithm similarly as the binary setting. We omit the straightforward details but mention one important implementation trick: When drawing the topic assignment $\mathbf{Z}$ we need to evaluate the acceptance probabilities which involves the following:

$$\sum_{c=1}^C g_d^c(\eta_k^c) = \sum_{c=1}^C \frac{\lambda y_d^c(1 + \lambda\xi_d^c)\eta_k^c}{N_d} - \lambda^2\xi_d^c \frac{(\eta_k^c)^2 + 2\eta_k^c \sum_{\kappa=1}^K \eta_\kappa^c n_{\kappa d}^{-di}}{2N_d^2}.$$

Naively evaluating the above costs $O(D\bar{N}L)$ time in total while through careful bookkeeping we can reduce it to $O(DLK + DK^2)$ time, which can be advantageous for long documents. The overall sampling time is $O(DLK + DK^2 + D\bar{N})$, significantly faster than the state-of-the-art: $O(LK^3 + DLK^2 + D\bar{N}K)$ in [24]. For prediction on the test set, we use the following rule:

$$\hat{y} = \underset{c=1,\ldots,C}{\text{argmax}} \; \bar{\mathbf{z}}^\top \boldsymbol{\eta}^c. \tag{28}$$

Comparing with the one-vs-all prediction rule (25), the latent topic assignments $\bar{\mathbf{z}}$ are shared here among all classes.

The third strategy is to use a genuine multi-class formulation, *e.g.* [6]. The resulting sampling algorithm differs substantially hence we do not discuss it here.

4.2 Extension to Other Losses

In the above we have mainly focused on the hinge loss $\ell_d(\boldsymbol{\eta}) := (1 - y_d \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d)_+$, but our linear time sampling algorithm can be easily extended to other losses, thanks to the classical result on scale-mixtures of the normal density:

THEOREM 2 (SCALE-MIXTURE OF NORMAL DENSITY, [2]). *Let the loss function $\ell : \mathbb{R}_+ \to \mathbb{R}_+$ be continuous. Then for all $\lambda > 0$ we have the scale-mixture representation of $\exp(-\lambda \ell(t))$ w.r.t. some density function $f_\lambda : \mathbb{R}_+ \to \mathbb{R}_+$,* i.e.

$$\exp(-\lambda \ell(t)) = \int_0^\infty \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{t^2}{2\sigma^2}\right) f_\lambda(\sigma^2) \mathrm{d}\sigma^2 \quad (29)$$

$$= \int_0^\infty \frac{1}{\sqrt{2\pi\xi^3}} \exp\left(-\frac{1}{2}\xi t^2\right) f_\lambda(1/\xi) \mathrm{d}\xi, \quad (30)$$

if and only if the function $\ell(\sqrt{t})$ is infinitely differentiable on $\mathbb{R}_{++}$ with derivatives satisfying $(-\frac{\mathrm{d}}{\mathrm{d}t})^n \ell(\sqrt{t}) \leq 0$ for all $n \geq 1, t > 0$.

When the derivative condition in Theorem 2 holds, we can find the density function $f_\lambda(\cdot)$ explicitly by computing the inverse Laplace transform of the function $\exp(-\lambda \ell(\sqrt{t}))$. Note that any scale-mixture must be a symmetric function (*c.f.* right-hand side of (29)), although extension to skewed scale-mixtures that break symmetry is straightforward, *i.e.*, allowing the normal density in (29) to have nonzero mean that may depend on t. The hinge loss we thoroughly discussed above is skewed: $\exp(-2\lambda \max\{t, 0\}) = \exp(-\lambda|t|)\exp(-\lambda t)$, where $\exp(-\lambda|t|)$ is a usual scale-mixture.

The scale-mixture of normal density is extremely useful when the prior distribution is also normal. Indeed, the regularized posterior distribution $q(\cdot)$ derived from (6) have the following more general form under any loss ℓ that admits the scale-mixture representation in (29):

$$q(\boldsymbol{\eta}, \boldsymbol{\xi}, \mathbf{Z}|\mathbf{W}) \propto \left[\prod_{d=1}^D \frac{\mathsf{B}(n_{\cdot d} + \boldsymbol{\alpha})}{\mathsf{B}(\boldsymbol{\alpha})}\right] \prod_{k=1}^K \frac{\mathsf{B}(\mathsf{n}_{k\cdot} + \boldsymbol{\beta})}{\mathsf{B}(\boldsymbol{\beta})}$$

$$\times p_0(\boldsymbol{\eta}) \prod_{d=1}^D \exp\left(-\frac{1}{2}\xi_d \zeta_d^2 - a\zeta_d\right) \tilde{f}_\lambda(\xi_d), \quad (31)$$

where ζ_d is a bi-affine function of the classifier weight $\boldsymbol{\eta}$ and the normalized latent topic assignment $\bar{\mathbf{z}}_d$, *e.g.*, $\zeta_d = 1 - y_d \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d$ for binary y_d or $\zeta_d = y_d - \boldsymbol{\eta}^\top \bar{\mathbf{z}}_d$ for real-valued y_d. If we impose the normal prior $p_0(\boldsymbol{\eta}) = \prod_{k=1}^K \mathcal{N}(\eta_k; 0, \nu^{-1})$, then $\boldsymbol{\eta}$ in the posterior (31), when conditioned on all other variables, follows again the normal distribution hence can be efficiently sampled.

The fast sampling algorithm we developed so far extends immediately to the more general regularized posterior (31):

- Sampling the augmented variable $\boldsymbol{\xi}$ reduces to repeatedly sample the univariate density $\exp(-\frac{1}{2}\xi_d \zeta_d^2)\tilde{f}_\lambda(\xi_d)$. The efficiency of this step of course depends on the density $\tilde{f}_\lambda$. We have extensively discussed the hinge loss that leads to the inverse Gaussian density. We mention two more examples. For the ϵ-insensitive loss $\ell(\zeta_d) = (|\zeta_d| - \epsilon)_+$ used in regression [24], we are sampling again the inverse Gaussian, which can be done in $O(1)$ time [13]. For the logistic loss $\ell(\zeta_d) = -y_d\zeta_d + \log(1 + \exp(\zeta_d))$, we need to sample the Pólya-Gamma distribution, which again can be done efficiently [16].
- Sampling the topic assignment $\mathbf{Z}$ is almost the same as in §3.3, except some minor change on the acceptance probability and the exp-proposal: we need to compute $-\gamma_d \eta_k^2 + \delta_d \eta_k - \psi_d m_d^{-di} \eta_k$, where the coefficients $\gamma_d, \delta_d, \psi_d$ depend on the exact form of ζ_d. Importantly, the classifier reweighted count m_d^{-di} can be incrementally maintained in $O(1)$ time as before.
- Sampling the classifier $\boldsymbol{\eta}$ needs to be done using again the Gibbs sampler as in §3.4. As mentioned above, the conditional posterior of $\boldsymbol{\eta}$ for any scale-mixture loss ℓ is again normal, but its precision matrix is computationally expensive to form. Instead, we use the Gibbs sampler and through careful bookkeeping we can draw the K classifiers in all documents in total time $O(KD)$.

Cycling through the above three steps for M burn-in steps we get samples from the true (regularized) posterior. The overall time remains $O(DK + D\bar{N})$.

5. 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 [24], referred as GibbsMedLDA. As shown previously in [24], GibbsMedLDA outperforms most existing supervised topic models, and we refer the interested readers to [24] for detailed comparisons.

Datasets: We conduct experiments on three benchmark datasets[3]: the 20Newsgroups for binary and multi-task classification, a hotel review dataset for regression, and a Wikipedia dataset 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.

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

	# train	# test	# word	type
20NG	11,269	7,505	61,188	multi-class
Hotel	2,500	2,500	12,000	regression
Wiki	1,100,000	5,000	917,683	multi-label

[3]Available at: http://qwone.com/~jason/20Newsgroups, http://bigml.cs.tsinghua.edu.cn/~ningchen/data.htm, http://lshtc.iit.demokritos.gr/, respectively.

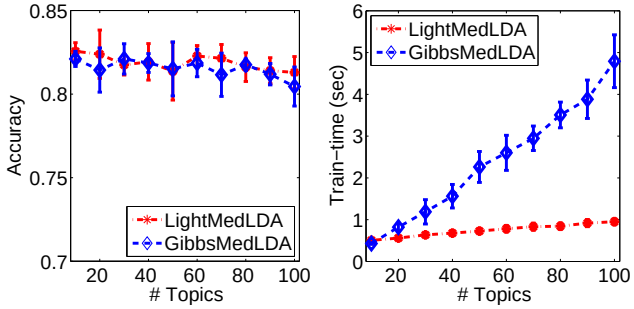

Figure 1: Classification accuracy (left) and training time (right) of LightMedLDA and GibbsMedLDA on the binary 20Newsgroups sub-dataset.

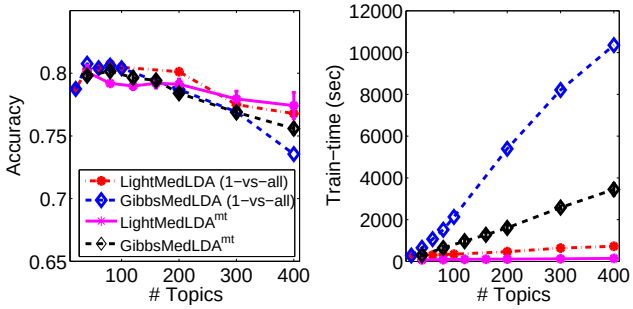

Figure 2: Classification accuracy and training time of LightMedLDA, GibbsMedLDA, multi-task LightMedLDA and multi-task GibbsMedLDA on the full 20Newsgroups dataset. One-vs-all strategy is used for binary classifiers.

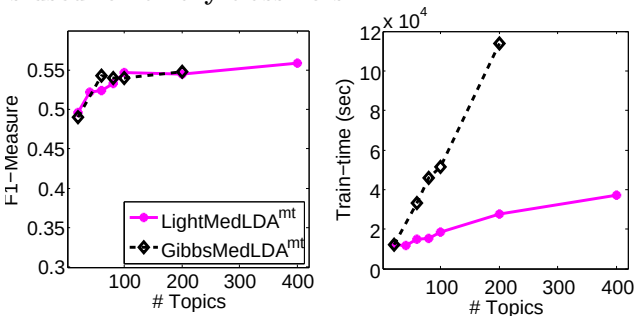

Figure 3: F1-measure and training time of LightMedLDA^{mt} and GibbsMedLDA^{mt} on the multi-labeled Wikipedia dataset.

5.1 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 [24], we set $\lambda = 262.4$ and the number of burn-in steps to $M = 10$. As shown in Figure 1 (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 1 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 mentioned in §4.1 to train 20 separate binary classifiers and used the prediction rule in (25). The results are shown in Figure 2, 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 2 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).

5.2 Multi-task Classification

In this subsection we test the multi-task formulation described in §4.1, again on the full 20Newsgroups dataset. We train all 20 classifiers simultaneously on the *same* latent representation, and we use the prediction rule in (28) to combine the classifiers. The results are shown in Figure 2, 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 3, 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.

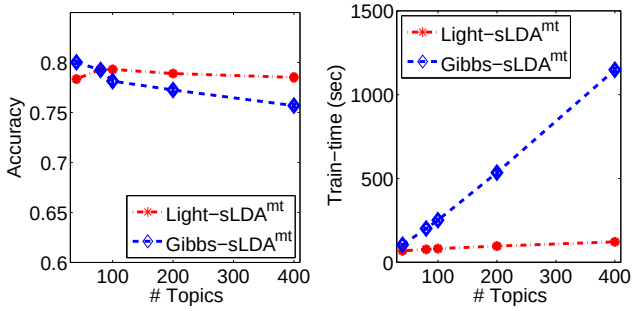

Figure 4: Classification accuracy and training time of Light-sLDA^{mt} and Gibbs-sLDA^{mt} on the full 20Newsgroups dataset.

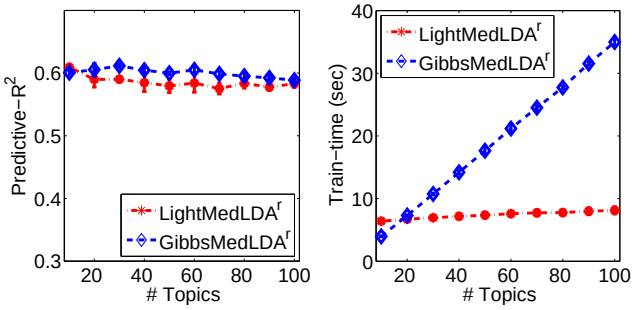

Figure 5: Predictive-R^2 and training time of LightMedLDA^r and GibbsMedLDA^r on the hotel review dataset.

5.3 Extension to Other Losses

All previous experiments were performed under the hinge loss. In this part we explore other loss functions for classification and regression. We tried the logistic loss for multi-task classification on 20Newsgroups and the ϵ-insensitive loss for regression on the hotel review dataset. See §4.2 for the algorithmic modifications needed for these losses. For the logistic loss, we set $\alpha_k \equiv 5.6/K$, $\lambda = 204.8$, and $M = 40$ for both our method, denoted as Light-sLDA, and the competitor Gibbs-sLDA in [25]. We used the more efficient multi-task formulation. For the ϵ-insensitive loss used for regression, following [3, 24] we used the predictive-R^2 as the performance measure (the larger the better). We set $\epsilon = 1e-3, \lambda = 262.4$, and the number of burn-in steps $M = 15$ for LightMedLDA and $M = 10$ for GibbsMedLDA. The results are shown in Figure 4 and Figure 5 respectively. Again, we observe that both algorithms achieved comparable performance while LightMedLDA consumed significantly less training time than GibbsMedLDA over the entire range of topic numbers.

5.4 Sensitivity Analysis

In this last part of experiments we conduct a thorough sensitivity analysis of the proposed LightMedLDA algorithm w.r.t. the different proposal compositions, the number of MH steps, and the number of Gibbs sub-iterations. We record both the classification accuracy and the training time on the full 20Newsgroups dataset.

Proposal compositions: Recall from §3 that the regularized posterior is factorized into three parts, and based on each we constructed the doc-proposal, the word-proposal, and the exp-proposal. Note that each proposal, equipped

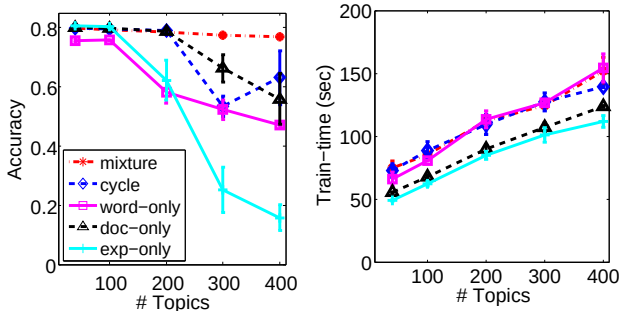

Figure 6: Classification accuracy and training time of LightMedLDA^{mt} on the full 20Newsgroups dataset with mixture of proposals, cycle of proposals, word-proposal only, doc-proposal only, and exp-proposal only.

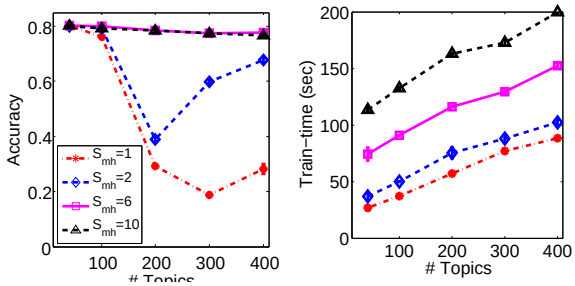

Figure 7: Classification accuracy and training time of LightMedLDA^{mt} on the full 20Newsgroups dataset with different number of MH steps.

with its correct acceptance probability, *e.g.* (19), (20), and (21), respectively, are all *bona fide* MCMC algorithms and should lead to the same stationary density eventually. However, their rate of convergence to the target posterior may be different. As verified in the left panel of Figure 6, using the mixture of all three proposals (*i.e.* uniformly randomly choosing one of them for $S_{mh} = 6$ times) leads to the best test accuracy, consistently on all topic numbers we tried. Using the exp-proposal alone leads to the poorest result, mostly because the exp-proposal is derived from the classifier part and contains the least information for drawing the topic assignment. Using the word-proposal alone is better than the exp-proposal but worse than the doc-proposal alone, possibly because the word-proposal is shared among documents hence may incur a longer lag while the doc-proposal refreshes more frequently (every time we switch documents). Interestingly, cyclically sampling the three proposals performs substantially worse than the mixture of proposals, even though in each iteration they used each of the proposals twice (on average). Another observation we draw from Figure 6 is that when the number of topics is small, all combinations of proposals seem to perform equally well. Moreovoer, we observe from the right panel of Figure 6 that the exp-proposal alone and the doc-proposal alone leads to substantially less training time.

Number of MH steps: Figure 7 shows the influence of the number of MH steps in drawing the topic assignments (§3.3). For space limits, we only present the result for the mixture of proposals (for its best accuracy verified above). It is clear that using more MH steps improves the accuracy but also increases the training time. In practice we found

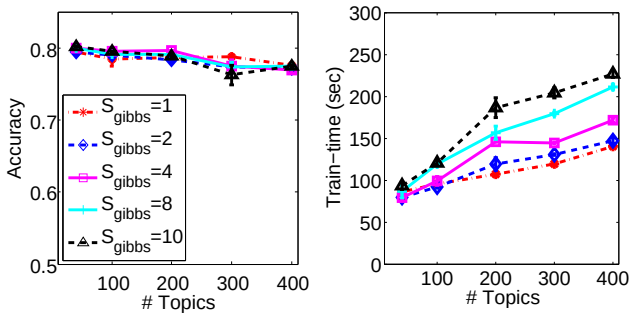

Figure 8: Classification accuracy and training time of LightMedLDA^{mt} on the full 20Newsgroups dataset with different number of Gibbs sub-iterations.

that using six MH steps (two for each proposal) seems to lead to the best tradeoff.

Number of Gibbs sub-iterations: Figure 8 shows the influence of the number of Gibbs sub-iterations in drawing the classifier (§3.4). Again we only present the result for the mixture of proposals. We observe that the prediction accuracy stays relatively constant while more Gibbs sub-iterations clearly increases the training time. In practice it seems 1 or 2 Gibbs sub-iterations would suffice.

6. 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 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.

Acknowledgment

We thank the reviewers for their valuable comments. This work was supported by NIH Grants R01GM087694 and P30DA035778, and NSF Grant IIS1447676.

References

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

[2] D. F. Andrews and C. L. Mallow. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society: Series B*, 36(1):99–102, 1974.

[3] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[4] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

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

[6] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

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

[8] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of National Academy of Science*, 101:5228–5235, 2004.

[9] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[10] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola. Reducing the sampling complexity of topic models. In *KDD*, 2014.

[11] F.-F. Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

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

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

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

[16] N. G. Polson, S. L. Scott, and J. Windle. Bayesian inference for logistic models using Pólya-Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

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

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

[19] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1728, 1994.

[20] A. J. Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3:253–256, 1977.

[21] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

[22] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma. LightLDA: Big topic models on modest compute clusters. In *WWW*, 2015.

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

[24] J. Zhu, N. Chen, H. Perkins, and B. Zhang. Gibbs max-margin topic models with data augmentation. *Journal of Machine Learning Research*, 15:1073–1110, 2014.

[25] J. Zhu, X. Zheng, and B. Zhang. Improved bayesian logistic supervised topic models with data augmentation. In *ACL*, 2013.

[26] J. Zhu, X. Zheng, L. Zhou, and B. Zhang. Scalable inference in max-margin topic models. In *KDD*, 2013.