Sparse Online Topic Models

Aonan Zhang, Jun Zhu, Bo Zhang
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory of Information Science and Technology
Department of Computer Science and Technology
Tsinghua University, Beijing 100084, China
{zan12; dcszj; dcszb}@mail.tsinghua.edu.cn

ABSTRACT

Topic models have shown great promise in discovering latent semantic structures from complex data corpora, ranging from text documents and web news articles to images, videos, and even biological data. In order to deal with massive data collections and dynamic text streams, probabilistic online topic models such as online latent Dirichlet allocation (OLDA) have recently been developed. However, due to normalization constraints, OLDA can be ineffective in controlling the sparsity of discovered representations, a desirable property for learning interpretable semantic patterns, especially when the total number of topics is large. In contrast, sparse topical coding (STC) has been successfully introduced as a non-probabilistic topic model for effectively discovering sparse latent patterns by using sparsity-inducing regularization. But, unfortunately STC cannot scale to very large datasets or deal with online text streams, partly due to its batch learning procedure. In this paper, we present a sparse online topic model, which directly controls the sparsity of latent semantic patterns by imposing sparsity-inducing regularization and learns the topical dictionary by an online algorithm. The online algorithm is efficient and guaranteed to converge. Extensive empirical results of the sparse online topic model as well as its collapsed and supervised extensions on a large-scale Wikipedia dataset and the medium-sized 20Newsgroups dataset demonstrate appealing performance.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical

General Terms

Algorithms, Experimentation

Keywords

Large-scale data, Online learning, Topic models, Sparse latent representations

1. INTRODUCTION

Probabilistic topic models, such as probabilistic latent semantic indexing [17] and its fully Bayesian generalization of latent Dirichlet allocation (LDA) [5], have been widely applied to discover latent semantic structures from collections of data, which can be text documents [5, 3, 6, 4, 9, 28], images [13, 34, 12, 31, 22, 37], and even biological data [1]. Since exact posterior inference is intractable, both variational [5] and Monte Carlo [15] methods have been widely developed for approximate inference, which can normally deal with medium-sized datasets. In order to deal with large-scale data analysis problems, which are not uncommon in many application areas, various techniques have been developed to speed up the inference algorithms, such as the parallel inference algorithms on multiple CPU or GPU cores and multiple machines (please see [40] for a nice summary of existing techniques). Another nice advance is the development of online inference algorithms, which can not only deal with massive data corpora but also can deal with dynamic text streams, where data samples are incoming one-by-one or in small batches. One representative work is the online variational inference method for latent Dirichlet allocation (OLDA) [16]. OLDA and its later extensions, including the online collapsed Gibbs sampling [20] and the hybrid online variational-Gibbs [27] methods have shown a success to scale to corpora containing millions of articles.

However, the above online probabilistic topic models can be ineffective in controlling the sparsity of the discovered representations, partly due to their normalization constraints on the admixing proportions [42]. Sparsity of the representations in a semantic space is a desirable property in text modeling [33] and human vision [29]. For example, we will expect not every topic or sense, but only a few of them that make a non-zero contribution for each document or each word [33]; this is especially important in practice for large scale text mining endeavors such as those undertaken in industry, where it is not uncommon to learn hundreds if not thousands of topics for millions or billions of documents. Without an explicit sparcification procedure, it would be extremely challenging, if not impossible, to nail down the semantic meanings of a document or word.

In this paper, we present an approach to learning sparse online topic models, both to improve time efficiency and to deal with streaming data. Our approach is based on our recent work of sparse topical coding (STC) [42], a hierarchical non-negative matrix factorization (NMF) [23] model using word codes and document codes to represent an article at the individual word level and the whole document level, respectively. By using unnormalized code vectors, STC offers an extra freedom to reconstruct word counts in text using a log-Poisson loss, and it can effectively control the sparsity of latent representations to find compact topical representations by imposing appropriate sparsity-inducing regulariza-

WWW 2013, May 13–17, 2013, Rio de Janeiro, Brazil.
ACM 978-1-4503-2035-1/13/05.

tion. Such effectiveness has been further demonstrated in the context of learning compact descriptors for images and videos [21, 14, 25]. However, the existing batch dictionary learning algorithm takes a full scan of the corpus at each gradient descent step, which is demanding in terms of both memory and computation; also, the batch algorithm cannot explore the redundancy of large-scale datasets for more effective training. Thus, in the current batch form, STC does not scale up to large-scale datasets and cannot deal with dynamic text streams.

To address the above weakness of STC, we propose a novel sparse online topic model, which is essentially an online algorithm to learn the topical dictionary in STC. Our algorithm, based on the recent success of online stochastic optimization [8, 32], can scale to large data corpora (e.g., the entire Wikipedia corpus containing 6.6M articles) and can cope with dynamic text streams. Our main contributions can be summarised as follows:

- We introduce online sparse topical coding (OSTC), which is efficient for learning online sparse topical representations.
- We provide a theoretical analysis that when using a general setting for the learning rate, our online learning algorithm converges to a stationary point under reasonable conditions.
- We present the collapsed sparse topical coding model as well as its online learning algorithm, and the online learning algorithm for the supervised max-margin sparse topical coding (MedSTC) [42].
- Our empirical results on the medium-sized 20Newsgroups dataset and a large-scale Wikipedia dataset show that 1) online learning algorithms can improve time efficiency, while not sacrificing prediction performance or the perplexity performance of held-out data; 2) online sparse topical coding achieves lower perplexity and higher word code sparsity than probabilistic online LDA.

The rest of the paper is structured as follows. Section 2 summarizes related works. Section 3 briefly overviews STC and its batch learning algorithm. Section 4 presents the online sparse topical coding algorithm, analyzes its convergence, and discusses two extensions for learning collapsed STC and supervised STC. Section 5 presents empirical results on Wikipedia and 20Newsgroups data. Finally, Section 6 concludes.

2. RELATED WORK

Various works have been developed for modeling independent dynamic text streams [39, 20] and dealing with large data corpora using topic models [16, 27, 38]. Online topic models combine these two targets into one objective. It has been shown that these models can easily scale up to a corpus containing a few millions of articles [16, 27] by using proper inference methods.

Another thing we care about is the sparsity of latent representations for the data [23]. Suppose we have an article, we can expect only a few topical meanings in it. In the language of topic models, the latent representations of the article and its words tend to be sparse. Sparsity is also important for large scale text mining endeavors, where it is common to cut down the semantic meaning of a document or word from its topical descriptors learned from millions of articles for storage. Several models aim at faster and more efficient inference procedures [15, 36, 2]. However, the inferred latent representations for these models are very dense. STC is a sparse topic model which relaxes the normalization constraints of the latent representations and explicitly put a sparse-inducing regularization on them. This method has been proved to be more successful to learn a sparse topical representation and its MAP inference is even significantly faster than some probabilistic topic models [42].

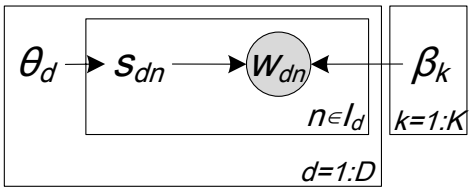


Figure 1: Graphical structure of STC [42].

Our online model is based on STC. We aim at building a topic model that can scale to a large corpus and can deal with dynamic text streams while simultaneously preserving sparse coding.

3. SPARSE TOPICAL CODING

In this section, we briefly overview sparse topical coding and its existing batch learning algorithm. We also provide a new interpretation for the sparse topic model from the projection point of view.

Let V be a vocabulary with N terms. In a bag-of-words model, a document d is represented as a vector $\mathbf{w}_d = (w_{d1}, \cdots, w_{d|I_d|})^\top$, where I_d is the index set of words that appear and w_{dn} $(n \in I_d)$ is the number of appearances of word n in document d. Sparse topical coding is a technique that projects the input $\mathbf{w}_d$ into a linear latent space spanned by a set of automatically learned bases (a basis set is also called a *dictionary*). The combination weights denote a representation of document d in the latent space. STC is a hierarchical non-negative matrix factorization [29], with two-layers of latent representations for words and the entire documents, respectively. For the ease of understanding, it is helpful to start with a probabilistic generating process, which also provides an explicit comparison with LDA.

3.1 A Probabilistic Generating Process

Let $\boldsymbol{\beta}$ denote a dictionary with K bases, of which each row $\boldsymbol{\beta}_k$ is a N dimensional basis. For text documents, $\boldsymbol{\beta}_k$ is a topic, i.e., a unigram distribution over the terms in V. This statement leads to the constraint that $\boldsymbol{\beta}_k \in \mathcal{P}$, where $\mathcal{P}$ is a $(N-1)$-simplex. We will use $\boldsymbol{\beta}_{\cdot n} \in \mathbb{R}^K$ to denote the nth column of $\boldsymbol{\beta}$. Graphically, STC is a hierarchical latent variable model, as shown in Fig. 1, where $\boldsymbol{\theta}_d \in \mathbb{R}^K$ is the *document code* (i.e., the latent representation of a document d) while each $\mathbf{s}_{dn} \in \mathbb{R}^K$ is a *word code* (i.e., latent representation of the individual word n in document d).

Formally, STC assumes that for each document d the word codes $\mathbf{s}_{dn}$ are conditionally independent given its document code $\boldsymbol{\theta}_d$ and the observed word counts w_{dn} are independent given their latent representations $\mathbf{s}_{dn}$. The generative process for each document d is:

1. draw a document code $\boldsymbol{\theta}_d \sim p(\boldsymbol{\theta})$;
2. for each word $n \in I_d$:
   (a) draw a word code $\mathbf{s}_{dn} \sim p(\mathbf{s}|\boldsymbol{\theta}_d)$;
   (b) draw a word count $w_{dn} \sim p(w|\mathbf{s}_{dn}, \boldsymbol{\beta})$.

For the last step of generating word counts, we require the distribution to satisfy the constraint $\mathbb{E}_p[w] = \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n} + \epsilon$, where ϵ is a small positive number for avoiding degenerated distributions. One nice choice, as used in STC, is the Poisson distribution

$$p(w_{dn}|\mathbf{s}_{dn}, \boldsymbol{\beta}) = Poisson(w_{dn}; \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n} + \epsilon), \quad (1)$$

where the linear combination $\mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n}$ has been used as the *mean* parameter of a Poisson distribution $Poisson(x; \nu) = \frac{\nu^x e^{-\nu}}{x!}$. This idea of using the linear combination $\mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n}$ as mean parameters can be generalized to the broad class of exponential family distributions for modeling various types of data. We refer the readers to [42] for more details. But we emphasize one advantage of such a mean parametrization, that is, using the linear combination as mean parameter makes it natural and convenient to constrain the feasible domains (e.g., non-negative domain for modeling word counts) of the word codes in order to have a good interpretation, while it would be reluctant to do so when using the linear combination as natural parameters[1]. As shown in [23], imposing appropriate constraints such as non-negativity constraints could result in significantly sparser and more interpretable patterns.

3.2 STC as a MAP Estimation

The generating procedure defines a joint distribution

$$p(\boldsymbol{\theta}_d, \mathbf{s}_d, \mathbf{w}_d|\boldsymbol{\beta}) = p(\boldsymbol{\theta}_d) \prod_{n \in I_d} p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d) p(w_{dn}|\mathbf{s}_{dn}, \boldsymbol{\beta}), \quad (2)$$

where $\mathbf{s}_d = \{\mathbf{s}_{dn}, n \in I_d\}$. To infer sparse word codes, STC defines $p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d)$ as a product of two component distributions

$$p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d) \propto p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d, \gamma) p(\mathbf{s}_{dn}|\rho) \quad (3)$$

where $p(\mathbf{s}_{dn}|\boldsymbol{\theta}_d, \gamma)$ is an isotropic Gaussian distribution $\mathcal{N}(\boldsymbol{\theta}_d, \gamma^{-1})$ and $p(\mathbf{s}_{dn}|\rho) = \text{Laplace}(0, \rho^{-1})$ is a Laplace distribution. This composite distribution is super-Gaussian [19] and the Laplace term will bias towards finding sparse word codes. For $p(\boldsymbol{\theta}_d)$, both the normal prior $p(\boldsymbol{\theta}_d) = \mathcal{N}(0, \lambda^{-1})$ and the Laplace prior $p(\boldsymbol{\theta}_d) = \text{Laplace}(0, \lambda^{-1})$ were discussed in [42].

Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_d\}, \mathbf{S} = \{\mathbf{s}_d\}$ and $\mathbf{W} = \{\mathbf{w}_d\}$ to denote all the latent document codes, latent word codes and observed word counts in the whole corpus. When $p(\boldsymbol{\theta})$ is normal, STC solves the constrained problem

$$\min_{\boldsymbol{\Theta}, \mathbf{S}, \boldsymbol{\beta}} \ \ell(\mathbf{S}, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\Theta}\|_2^2 + \frac{\gamma}{2} \sum_{d, n \in I_d} \|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2 + \rho \|\mathbf{S}\|_1$$
$$\text{s.t.}: \ \boldsymbol{\Theta} \geq 0; \ \mathbf{S} \geq 0; \ \ \boldsymbol{\beta}_k \in \mathcal{P}, \ \forall k, \quad (4)$$

where the objective function is the negative logarithm of the posterior $p(\boldsymbol{\Theta}, \mathbf{S}, \boldsymbol{\beta}|\mathbf{W})$ with a constant omitted; $\|\boldsymbol{\Theta}\|_2^2 = \sum_d \|\boldsymbol{\theta}_d\|_2^2$ and $\|\mathbf{S}\|_1 = \sum_{d,n} \|\mathbf{s}_{dn}\|_1$. For text, the log-Poisson loss is $\ell(\mathbf{S}, \boldsymbol{\beta}) = \sum_d \ell(\mathbf{s}_d, \boldsymbol{\beta})$, where

$$\ell(\mathbf{s}_d, \boldsymbol{\beta}) = \sum_{n \in I_d} \ell(w_{dn}, \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n}) \quad (5)$$

is the log-loss contributed by document d and

$$\ell(w_{dn}, \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n}) = -\log Poisson(w_{dn}; \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n} + \epsilon) \quad (6)$$

is the loss contributed by the individual word n. Since word counts are non-negative, a negative $\boldsymbol{\theta}$ or $\mathbf{s}$ will lose interpretability. Therefore, STC constrains the code parameters to be non-negative, as in [18]. A non-negative code $\boldsymbol{\theta}$ or $\mathbf{s}$ can be interpreted as representing the relative importance of topics. The parameters (λ, γ, ρ) are non-negative constants and they can be selected via cross-validation.

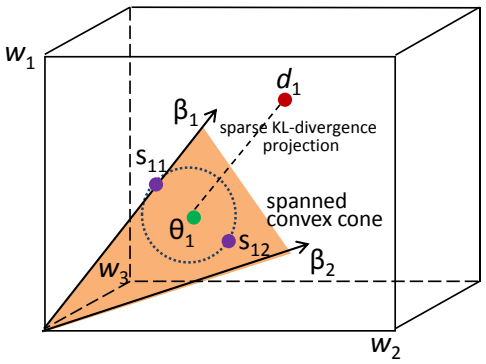


Figure 2: A new projection view of STC with two topical bases over a vocabulary with three terms.

To help understand the above definition of STC, we also provide a new projection interpretation of STC as illustrated in Fig. 2. Suppose we have two topical bases $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ over a vocabulary with three terms $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$. The document d_1 has two terms, each being projected to a point in the spanned convex cone[2] under a KL-divergence measure[3], and the document code $\boldsymbol{\theta}_1$ is an aggregation of the two word codes $\mathbf{s}_{11}$ and $\mathbf{s}_{12}$. By using appropriate regularization, the projection could be sparse. In this figure we illustrate both sparse and non-sparse cases. For example, the word code $\mathbf{s}_{11}$ is sparse (i.e., on the boundary) while $\mathbf{s}_{12}$ is not.

3.3 Existing Batch Learning Algorithm

Problem (4) is biconvex, i.e., convex over $\boldsymbol{\beta}$ or $(\boldsymbol{\Theta}, \mathbf{S})$ when the other is fixed, but not joint convex over $(\boldsymbol{\Theta}, \mathbf{S}, \boldsymbol{\beta})$. A natural algorithm to solve this biconvex problem for a local optimum is coordinate descent, as used in [42] and sparse coding methods [24]. The algorithm alternately performs the following two steps.

Hierarchical sparse coding: optimizing over $\mathbf{S}$ and $\boldsymbol{\Theta}$. Since documents are i.i.d, we can perform the hierarchical sparse coding for each document separately. For document

[1]For example, the natural parameter of the Poission distribution $Poisson(x; \nu)$ is $\log \nu$. If we use the natural parametrization and let $\log \nu = \mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n} + \epsilon$, we will have $\nu = \exp(\mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n} + \epsilon)$. The exponential transformation will make the resulting problem of STC hard to solve.

[2]The combination weight is a word code.

[3]Minimizing the log-Poisson loss in Eq. (6) is equivalent to minimizing the unnormalized KL-divergence between observed word counts w_{dn} and their reconstructions $\mathbf{s}_{dn}^\top \boldsymbol{\beta}_{\cdot n}$ [35].

d, we solve the constrained optimization problem

$$\min_{\boldsymbol{\theta}_d, \mathbf{s}_d} \ \ell(\mathbf{s}_d, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\theta}_d\|_2^2 + \frac{\gamma}{2} \sum_{n \in I_d} \|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2 + \rho \|\mathbf{s}_d\|_1$$
$$\text{s.t.} : \boldsymbol{\theta}_d \geq 0; \ \mathbf{s}_{dn} \geq 0, \forall n \in I_d. \quad (7)$$

As shown in [42], a coordinate descent procedure can be developed with iterative closed-form updates for word codes and document codes. Moreover, this algorithm has the same structure as the variational inference algorithm of the counterpart LDA [5] model. To compare with online LDA [16], which uses variational inference, we adopt the coordinate descent strategy to solve problem (7) in our online sparse topical coding. More formally, the algorithm alternatively solves

Optimize over $\mathbf{s}_d$: when $\boldsymbol{\theta}_d$ is fixed, $\mathbf{s}_{dn}$ are not coupled. For each $\mathbf{s}_{dn}$, the solution is $s_{dn}^k = \max(0, \nu_{dn}^k)$, where ν_{dn}^k is the larger solution of the equation

$$\gamma \beta_{kn} (\nu_{dn}^k)^2 + (\gamma \mu + \beta_{kn} \eta) \nu_{dn}^k + \mu \eta - w_{dn} \beta_{kn} = 0,$$

where $\mu = \sum_{j \neq k} s_{dn}^j \beta_{jn} + \epsilon$ and $\eta = \beta_{kn} + \rho - \gamma \theta_d^k$. This one dimensional problem can be solved in closed-form.

Optimize over $\boldsymbol{\theta}_d$: when $\mathbf{s}_d$ is fixed, the closed-form solution is

$$\forall k, \ \theta_d^k = \frac{\gamma}{\lambda / |I_d| + \gamma} \bar{s}_d^k, \quad (8)$$

where $\bar{s}_d^k = \frac{1}{|I_d|} \sum_{n \in I_d} s_{dn}^k$. If $\lambda \ll \gamma$, the document code $\boldsymbol{\theta}_d$ is close to the *averaging* aggregation of its individual word codes. Another choice is to set $\lambda = \gamma$, and we have $\theta_d^k = \frac{|I_d|}{1 + |I_d|} \bar{s}_d^k$, which is again close to the average if $|I_d|$ is large. Following [42], we set $\lambda = \gamma$ since it reduces one parameter to tune. Moreover, if the Laplace prior $p(\boldsymbol{\theta}_d) = \text{Laplace}(0, \lambda^{-1})$ is used, a closed-form solution also exists,

$$\forall k, \ \theta_d^k = \max(0, \bar{s}_d^k - \frac{\lambda}{\gamma |I_d|}), \quad (9)$$

which is a *truncated averaging* strategy for aggregating individual word codes to obtain $\boldsymbol{\theta}_d$.

Dictionary learning: this step involves solving

$$\min_{\boldsymbol{\beta}} \ \ell(\mathbf{S}, \boldsymbol{\beta}), \quad \text{s.t.} : \boldsymbol{\beta}_k \in P, \forall k. \quad (10)$$

STC uses a projected gradient descent method to update $\boldsymbol{\beta}$, where the projection to the ℓ_1-ball can be done efficiently in $O(N)$ time [11]. We will use the public implementation of the batch algorithm as our baseline[4].

4. ONLINE SPARSE TOPICAL CODING

The above algorithm empirically converges faster than the variational inference algorithm of probabilistic LDA by avoiding calls to digamma function [42]. However, it requires a full pass through the corpus at each gradient descent step of learning dictionary. A full pass of a very large dataset would be very expensive in terms of both memory and efficiency. Furthermore, the batch gradient descent for dictionary learning can be inefficient in utilizing the redundance information of a large dataset. To overcome such inefficiency, we propose the online sparse topical coding (OSTC), which uses an online learning algorithm to learn the dictionary $\boldsymbol{\beta}$. Our online algorithm is nearly as simple as the batch coordinate descent algorithm for STC, but converges much faster for large datasets, as we shall see.

[4]http://www.ml-thu.net/∼jun/stc.html

Algorithm 1 Online Sparse Topical Coding

1: Initialize $\boldsymbol{\beta}^0, \boldsymbol{\theta}_0, s_0$
2: **for** t= 0,1,2,... **do**
3: read document d^t
4: $(\boldsymbol{\theta}_t, \mathbf{s}_t) = HierarchicalSparseCoding(d^t)$
5: let $g^t = \nabla \ell(\boldsymbol{\beta}^t)$ and $\alpha^t = \tau_0 / (t + \tau)$
6: $\boldsymbol{\beta}^{t+1} \leftarrow \Pi_P(\boldsymbol{\beta}^t - \alpha^t g^t)$
7: **end for**

The online learning algorithm for STC is described in algorithm 1. At each iteration t, we randomly sample a data point $\mathbf{w}_t$ and perform the hierarchical sparse coding step to find the optimal codes $\boldsymbol{\theta}_t$ and $\mathbf{s}_t$, holding the dictionary fixed. Then, we update the dictionary using the information collected from the data $\mathbf{w}_t$ by using the first-order update rule

$$\boldsymbol{\beta}^{t+1} = \Pi_P(\boldsymbol{\beta}^t - \alpha^t \mathbf{g}(\boldsymbol{\beta}^t; \mathbf{w}_t)) \quad (11)$$

where the gradient

$$\mathbf{g}(\boldsymbol{\beta}_t; \mathbf{w}_t) = \nabla \ell(\mathbf{s}_t, \boldsymbol{\beta})|_{\boldsymbol{\beta}^t}$$

and α^t denotes the learning rate. The update rule is in fact the solution of the subproblem

$$\min_{\boldsymbol{\beta}} \ \ell(\mathbf{s}_t, \boldsymbol{\beta}^t) - \alpha^t \langle \mathbf{g}(\boldsymbol{\beta}^t; \mathbf{w}_t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \rangle + \frac{1}{2} \|\boldsymbol{\beta} - \boldsymbol{\beta}^t\|_2^2$$

under a projection to ensure $\boldsymbol{\beta}$ be a topical dictionary. We have denoted the projection to the simplex P by Π_P.

Mini-batches: A useful technique to reduce noise in stochastic learning is to consider multiple observations per iteration. Suppose we have M data at each iteration. After fitting the sparse codes for each document, the online update rule is

$$\boldsymbol{\beta}^{t+1} = \Pi_P(\boldsymbol{\beta}^t - \alpha^t \frac{1}{M} \sum_{d=1}^{M} \mathbf{g}(\boldsymbol{\beta}_t; \mathbf{w}_t^d)), \quad (12)$$

where $\mathbf{w}_t^d$ is the dth document in mini-batch t. Note that when $M = D$, we recover the batch STC. To provide some intuitive ideas, an illustration of the online learning procedure is shown in Fig. 3, whose detail description will be presented at the end of this section, after we have presented the convergence analysis and extensions.

Comparison with online LDA: Recently efficient online learning algorithms have been proposed for LDA to scale up to large datasets and to deal with dynamic text streams [16, 20, 27]. Our algorithm closely resembles the online variational Bayesian algorithm for LDA [16]. This similarity makes it convenient to compare the two variants of online topic models, including time efficiency and sparsity of word codes, as reported in the experiments.

4.1 Analysis of Convergence

The deterministic formulation of STC allows us to analyze the convergence behavior of the online algorithm. First, we analyze the regularity of the objective function in dictionary learning.

Lemma 1. *The cost function $\ell(\mathbf{s}_t, \boldsymbol{\beta}; \mathbf{w}_t)$ is convex over $\boldsymbol{\beta}$ and bounded from below; and its gradient and Hessian matrix are bounded.*

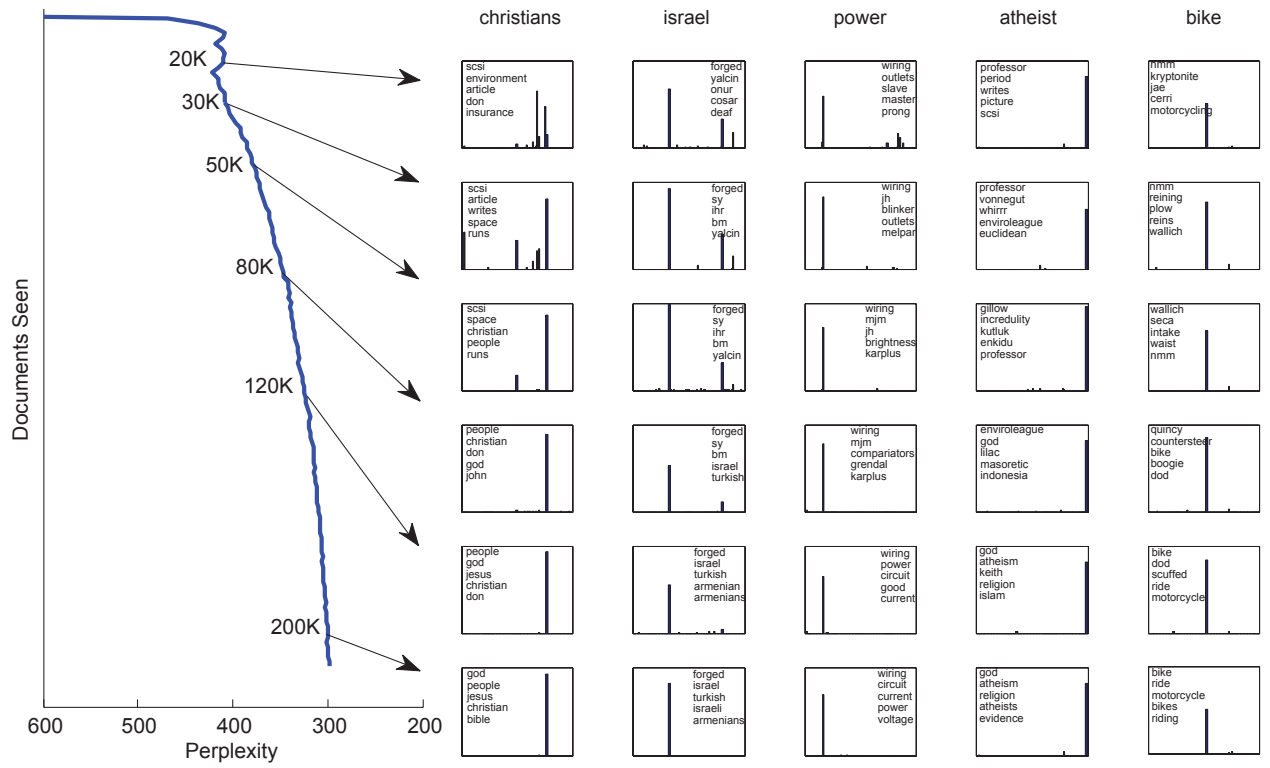


Figure 3: The change of perplexity and average word codes on test documents during the training process of OMedSTC (See section 4.2.2), as the online algorithm scans more articles (see the numbers near the blue curve). From top to bottom, we can see that the held-out perplexity drops down (in the left figure); the average word codes grow sparser (the right five columns); and the semantic meaning of the most salient topics representing the 5 selected words becomes clearer (for each topic, we present the 5 top-ranked terms inside the boxes).

Proof: The first part is obvious for the log-Poisson loss, since we have avoided the degenerated cases by introducing the parameter ϵ and the maximum word count is bounded in real cases. The gradient $\nabla_{\boldsymbol{\beta}_{.n}}\ell(\mathbf{s}_t,\boldsymbol{\beta};\mathbf{w}_t)=\mathbb{I}(n\in I_t)(1-\frac{w_{tn}}{\mathbf{s}_{tn}^\top\boldsymbol{\beta}_{.n}+\epsilon})\mathbf{s}_{tn}$ is also bounded for the same reason. For the last part, we directly prove the largest eigenvalue of Hessian matrix is bounded:

$$\begin{aligned}\lambda_{max} &= \sup_{\boldsymbol{\beta}\geq 0}\sup_{\|z\|_2\leq 1} z^\top\nabla^2_{\boldsymbol{\beta}_{.n}}\ell(\mathbf{s}_t,\boldsymbol{\beta};\mathbf{w}_t)z\\ &= \sup_{\boldsymbol{\beta}\geq 0}\sup_{\|z\|_2\leq 1} z^\top\Big(\sum_{n\in I_t}\frac{\mathbf{s}_{tn}w_{tn}\mathbf{s}_{tn}^\top}{(\mathbf{s}_{tn}^\top\boldsymbol{\beta}_{.n}+\epsilon)^2}\Big)z\\ &= \sup_{\|z\|_2\leq 1} z^\top\Big(\sum_{n\in I_t}\frac{\mathbf{s}_{tn}w_{tn}\mathbf{s}_{tn}^\top}{\epsilon^2}\Big)z=\frac{\|\mathbf{s}_t\mathrm{diag}(\mathbf{w}_t)\mathbf{s}_t^\top\|_2}{\epsilon^2}\\ &\leq \frac{1}{\epsilon^2}\|\mathbf{s}_t\|_2^2\|\mathrm{diag}(\mathbf{w}_t)\|_2\leq\frac{w_{t_{max}}}{\epsilon^2}\|\mathbf{s}_t\|_1\|\mathbf{s}_t\|_\infty.\end{aligned}$$

Since $\mathbf{s}_{tn}$ and $\boldsymbol{\beta}_{.n}$ are non-negative, the first supremum is achieved when $\mathbf{s}_{tn}^\top\boldsymbol{\beta}_{.n}=0$. Then we use the definition of the induced matrix 2-norm to get a more compact expression. Finally, using inequalities of matrix norm and the maximum word count $w_{t_{max}}$, we get the last inequality. Note that $\|\mathbf{s}_t\|_1=\max_k\Sigma_n\mathbf{s}_{tn}^k$ was bounded by the number of different words exist in a mini-batch and $\|\mathbf{s}_t\|_\infty=\max_n\Sigma_k\mathbf{s}_{tn}^k$ relates to the scale of $\mathbf{s}_{tn}$ and was controlled by hyperparameters. So the Hessian matrix of $\ell(\mathbf{s}_t,\boldsymbol{\beta};\mathbf{w}_t)$ is bounded. □

To analyze the convergence of OSTC, we follow the method used in [16]. Suppose that we sample articles together with their word codes, then we can compute the expected gradient of the cost function. Since STC and OSTC perform MAP estimates and find a single value of each word code, we compute the expectation over $\mathbf{s}$ by using an impulse distribution with our estimate of the codes. Then, we can derive results which are similar as in [7] to ensure that our online algorithm converge to a stationary point, as shown in the following theorem.

Theorem 2. *Assume that the learning rate α^t satisfies $\sum_{t=1}^\infty(\alpha^t)^2<\infty$, $\sum_{t=1}^\infty\alpha^t=\infty$. Then, OSTC converges.*

Proof: The proof is partly based on [7]. We first define the *Lyapunow sequence* $h^t=\|\boldsymbol{\beta}^t-\boldsymbol{\beta}^*\|^2$ where $\boldsymbol{\beta}^*$ is a stationary point and prove that $\boldsymbol{\beta}^t$ converges based on the convergence of h^t. We denote the previous knowledge (i.e., $\boldsymbol{\beta}^{\hat{t}},\boldsymbol{\theta}_{\hat{t}},\mathbf{s}_{\hat{t}},\forall 0\leq\hat{t}\leq t$) by P^t. Then

$$\begin{aligned}\mathbb{E}[h^{t+1}-h^t|P^t] &= -2\alpha^t(\boldsymbol{\beta}^t-\boldsymbol{\beta}^*)\mathbb{E}_{\mathbf{w}_t}[\nabla_{\boldsymbol{\beta}}\ell(\mathbf{s}_t,\boldsymbol{\beta}^t;\mathbf{w}_t)|P^t]\\ &\quad+(\alpha^t)^2\mathbb{E}_{\mathbf{w}_t}[(\nabla_{\boldsymbol{\beta}}\ell(\mathbf{s}_t,\boldsymbol{\beta}^t;\mathbf{w}_t))^2|P^t]\end{aligned}$$

Note that the first order derivative is bounded and the second order term was also bounded by $A+B(\boldsymbol{\beta}^t-\boldsymbol{\beta}^*)^2$, where A and B are non-negative values. This is because the eigenvalues of Hessian matrix is bounded and the gradient will

not exceed a polynomial threshold. Transforming previous equation we get

$$\mathbb{E}[(h^{t+1} - (1-(\alpha^t)^2 B)h^t | P^t] = \\ -2\alpha^t(\boldsymbol{\beta}^t - \boldsymbol{\beta}^*)\mathbb{E}_{\mathbf{w}_t}[\nabla_{\boldsymbol{\beta}}\ell(\mathbf{s}_t, \boldsymbol{\beta}^t; \mathbf{w}_t)|P^t](\alpha^t)^2 A \quad (13)$$

Using the techniques in [7], if we replace h^t with a scaling term and choose $\alpha^t = \tau_0/(t+\tau)$ where τ and τ_0 are positive constants, we can prove that h^t converges and the infinite sum of the left hand side of Eq. (13) also converges. Therefore, the infinite sum of the right hand side of Eq. (13) also converges, i.e.,

$$\sum_{t=1}^{\infty} \alpha^t(\boldsymbol{\beta}^t - \boldsymbol{\beta}^*)\mathbb{E}_{\mathbf{w}_t}[\nabla_{\boldsymbol{\beta}}\ell(\mathbf{s}_t, \boldsymbol{\beta}^t; \mathbf{w}_t)|P^t] < \infty. \quad (14)$$

Since $\sum_{t=1}^{\infty}\alpha^t = \sum_{t=1}^{\infty}\tau_0/(t+\tau) = \infty$ and the first order derivative is bounded, we must have that $|\boldsymbol{\beta}^t - \boldsymbol{\beta}^*|$ converges to zero. □

In all the experiments, we set $\alpha^t = \tau_0/(t+10)$, which satisfies the assumptions in the above theorem.

4.2 Extensions

Before ending this section, we briefly present two extensions of the online learning algorithm for collapsed sparse topical coding and max-margin supervised dictionary learning.

4.2.1 Online Collapsed STC

STC was intentionally designed as having a hierarchical structure, similar as the hierarchical probabilistic topic models, for easy comparison. But for practical performance, it has been demonstrated in probabilistic topic models that collapsing some parts of the latent variables could potentially improve performance [15]. We take the analogy and develop a collapsed STC (CSTC), and show that our online learning algorithm can be naturally extended for CSTC.

Specifically, as described in Section 3.2, STC is a MAP estimate of a hierarchical Bayesian model. When using a normal prior on $\boldsymbol{\theta}$, we can derive the collapsed STC by marginalizing out $\boldsymbol{\theta}$. For each document d, we have the collapsed distribution

$$p(\mathbf{s}_d, \mathbf{w}_d|\boldsymbol{\beta}) \propto \zeta_d \int_{\boldsymbol{\theta}_d} \exp\Big\{-\lambda\|\boldsymbol{\theta}_d\|_2^2 - \frac{\gamma}{2}\sum_{n\in I_d}\|\mathbf{s}_{dn} - \boldsymbol{\theta}_d\|_2^2\Big\} \\ \propto \zeta_d \exp\Big\{-a\sum_{n\in I_d}\|\mathbf{s}_{dn}\|_2^2 + 2b\sum_{m\neq m'}\mathbf{s}_{dm}^\top \mathbf{s}_{dm'}\Big\},$$

where $\zeta_d = \exp\{-\ell(\mathbf{s}_d, \boldsymbol{\beta}) - \rho\|\mathbf{s}_d\|_1\}$ is independent of $\boldsymbol{\theta}_d$, $a = \frac{\gamma}{2} - \frac{\gamma^2}{4(\lambda + \frac{\gamma|I_d|}{2})}$ and $b = \frac{\gamma^2}{4(\lambda+\frac{\gamma|I_d|}{2})}$. Then, by performing MAP estimation, we derive the collapsed STC as solving

$$\min_{\mathbf{S},\boldsymbol{\beta}} \quad \mathrm{tr}(\mathbf{s}_d^\top \Lambda \mathbf{s}_d) + \ell(\mathbf{s}_d, \boldsymbol{\beta}) + \rho\|\mathbf{s}_d\|_1 \quad (15)$$
$$\text{s.t.:} \quad \mathbf{S} \geq 0;\ \boldsymbol{\beta}_{k\cdot} \in P, \forall k,$$

where $\Lambda = (a-b)I + bE$ and $\mathbf{s}_d$ is an $K \times |I_d|$ matrix, of which the column n corresponds to $\mathbf{s}_{dn}$.

The problem is again biconvex, i.e., convex over $\mathbf{S}$ or $\boldsymbol{\beta}$ when the other is fixed. Both batch and online algorithms can be developed to solve Eq. (15), since the dictionary learning step is the same as in STC. The difference is on the sparse coding step, which is now to find the optimal word codes for each document. We can also derive a coordinate descent algorithm, of which each substep has a closed-form solution. Specifically, the optimal solution of s_{dn}^k is $\max(0, \nu_{dn}^k)$, where ν_{dn}^k is the larger solution of the quadratic equation

$$2a\beta_{kn}(\nu_{dn}^k)^2 + c\beta_{kn}\nu_{dn}^k + c\sum_{k'\neq k} s_{dn}^{k'}\beta_{k'n} - w_{dn}\beta_{kn} = 0$$

where $c = \beta_{kn} + \rho + 2b\sum_{m\neq n} s_{dm}^k$.

4.2.2 Online Max-margin STC

Both STC and CSTC learn dictionaries and infer sparse representations of unlabeled samples. But with the increasing availability of free on-line information such as image tags, user ratings, etc., various forms of "side-information" that can potentially offer "free" supervision have lead to a need for new topic models and training schemes that can make an effective use of such information to achieve better results, such as more discriminative latent representations of text contents and more accurate classifiers [4, 41]. In [42], a supervised max-margin STC (MedSTC) was developed to learn predictive representations and a supervised dictionary [26] by exploring the available side-information.

The basic idea of MedSTC is to use document codes as input features for max-margin classifiers, e.g., the multi-class SVM [10]. Formally, MedSTC solves the problem

$$\min_{\boldsymbol{\Theta},\mathbf{S},\boldsymbol{\beta},\boldsymbol{\eta}} \quad f(\boldsymbol{\Theta}, \mathbf{S}, \boldsymbol{\beta}) + C\mathcal{R}(\boldsymbol{\Theta}, \boldsymbol{\eta}) + \frac{1}{2}\|\boldsymbol{\eta}\|_2^2 \quad (16)$$
$$\text{s.t.:} \quad \boldsymbol{\Theta} \geq 0; \mathbf{S} \geq 0; \boldsymbol{\beta}_k \in P, \forall k,$$

where $f(\boldsymbol{\Theta}, \mathbf{S}, \boldsymbol{\beta})$ is the objective function of STC and

$$\mathcal{R}(\boldsymbol{\Theta}, \boldsymbol{\eta}) = \frac{1}{D}\sum_d \max_y[\Delta(y_d, y) + \boldsymbol{\eta}_y^\top \boldsymbol{\theta}_d - \boldsymbol{\eta}_{y_d}^\top \boldsymbol{\theta}_d]$$

is the multiclass hinge loss with parameters $\boldsymbol{\eta} = [\boldsymbol{\eta}_1; \cdots; \boldsymbol{\eta}_L]$ for L classes, of which each $\boldsymbol{\eta}_l$ is a K-dimensional vector associated with class l. The loss function $\Delta(y_d, y)$ measures the cost of making a prediction y if the ground truth label is y_d. Normally, we assume $\Delta(y, y) = 0$, i.e., no cost for a correct prediction.

The problem is again biconvex, i.e., convex over $(\Theta, \mathbf{S})$ or $(\boldsymbol{\beta}, \boldsymbol{\eta})$ when the other is fixed. In [42], a batch algorithm was developed to alternately solve for $(\boldsymbol{\Theta}, \mathbf{S})$ and $(\boldsymbol{\beta}, \boldsymbol{\eta})$. Since $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ are not coupled, we can solve for each of them separately. For $\boldsymbol{\eta}$, the subproblem is to learn a linear multi-class SVM. Based on the above online dictionary learning algorithm and the existing high-performance online learning algorithm for SVMs [32], we can develop an online learning algorithm for MedSTC, which is still guaranteed to converge. We denote this method by OMedSTC.

Before presenting all the details of the experiments, we use Fig. 3 to illustrate the change of the perplexity on held-out documents and the word codes along the iterations of online learning. We present the results of OMedSTC with 70 topics on the 20Newsgroup data with a standard train/test split, which will be clear in the next section. Fig. 3 shows the perplexity of the test set and the average word codes of the five popular words, of which each one is from a different category, at different stages of online learning. For each word, we calculate the average word code over the test documents that are from the category as that particular word. For example, the average word code of *bike* is the mean of

all the word codes for *bike* in the *rec.motorcycles* category. We can see the held-out perplexity goes down when scanning more articles while at the same time the average word codes for each word grows sparser and at the end of training most words are dominated by a few topics. It is also nice to see that the semantic meanings of the most salient topics describing the selected words become clearer by listing their top words (i.e., words that have highest values in the topic). For example, the average word code for the word *christians* was dominated by some not-clearly-meaningful topics when we scan 20K articles, while at the end of our algorithm it was captured by only one topic that has a very clear topical meaning, with the top five words being *god*, *people*, *jesus*, *christian*, and *bible*, all relating to the target word *christians*.

5. EXPERIMENTS

Now, we present all the details of our empirical results on a dataset with 6.6M articles collected from Wikipedia and the 20Newsgroups dataset to evaluate the effectiveness of online learning algorithms for STC, MedSTC and CSTC. We set the learning rate $\alpha^t = \tau_0/(t+10)$ and tune τ_0 for models with different batch sizes[5]. All the experiments are done on a standard desktop with 2.67GHz processors and 2GB RAM. Note that to reduce the influence of network speed, all the datasets were pre-collected. Thus, the experiments are not really online. But they suffice to evaluate the effectiveness and efficiency of the online learning algorithms.

5.1 Experiments on the Wikipedia Dataset

We first report the results on the unsupervised Wikipedia dataset. We use perplexity as the performance measure, which is defined as the geometric mean of the inverse marginal probability of each word in a held-out set of documents $\mathbf{W}^{test}$. Here, we randomly select 1000 articles as the held-out set. We compare OSTC with the ordinary batch STC and the online LDA (OLDA) using variational inference[6] [16]. We note that other versions of OLDA have been developed by doing hybrid variational inference and Monte Carlo sampling [27], which could improve the time efficiency of OLDA. But since our main focus is on topic sparsity[7], we compare with the variational OLDA, whose procedure is more similar as OSTC. We will discuss the influence of various inference methods for LDA on perplexity later. In the experiments, we set $K = 100$, which is sufficient to fit the data well[8].

Below we first explain the perplexity measure we use for our STC models, which is slightly different from the commonly used perplexity for probabilistic models like LDA.

5.1.1 Perplexity for STC models

Perplexity is a common measure of topic models' ability to generalize to test data. It is defined as the geometric mean of word likelihood. For probabilistic models, word likelihood is a marginal of the joint distribution of words and topic assignment, where the topic distribution is inferred from test data. But for STC, since we do not have a distribution of word codes, we then have our perplexity definition different with probabilistic topic models. We now use LDA as an example of probabilistic topic models to explicitly discuss its perplexity definition compared with STC.

For probabilistic topic models, the perplexity was defined as follows. Let n_i^{test} denote all words in a test document i and N_i^{test} is the total word counts in document i. Then the perplexity is the geometric mean of word likelihood in the test set:

$$\text{perplexity} = \exp\Big\{ - \frac{\sum_i \log p(n_i^{test})}{\sum_i N_i^{test}} \Big\}. \tag{17}$$

For LDA and OLDA, since exact inference is intractable, a variational bound was developed to approximate the perplexity [16]. However, this variational bound utilize words in the held-out set and may over-fit the test data. Here we use a 'document completion' method [30] to evaluate the held-out perplexity and this is done by first using half of the test words (denoted by n_{i1}^{test}) to infer document codes for the test documents and then evaluating the held-out perplexity by sampling word code for the other half of words in the test data (denoted by n_{i2}^{test}). This method avoid overfitting since n_{i2}^{test} was not used for inference. Precisely, the perplexity of LDA is computed as

$$\text{perplexity}_{LDA} \approx \exp\Big\{ - \frac{\sum_i \log p(n_{i2}^{test} | p(n_{i1}^{test}, \alpha, \boldsymbol{\beta})}{\sum_i |N_{i2}^{test}|} \Big\}. \tag{18}$$

For STC and OSTC, we do not define a posterior distribution of word codes, which means we can not compute the marginal of the joint distribution of words and topic assignment as in probabilistic topic models. However, in STC we can use a similar strategy as done in LDA by first utilizing half of the test terms (denoted by w_{i1}^{test}) to infer the document codes for the test set and then sample word codes for the other half of terms (denoted by w_{i2}^{test}) to calculate the held-out perplexity as

$$\text{perplexity}_{STC} \approx \exp\Big\{ - \frac{\sum_i \log p(w_{i2}^{test} | w_{i1}^{test}, \boldsymbol{\beta})}{\sum_i |I_{i2}^{test}|} \Big\}. \tag{19}$$

From above discussions, we argue that both perplexity definations are proper for their own settings. To further check this, we also provide an 'interchange' experiment in the Appendix. In the following experiments we will use the Eq. (18) to calculate perplexity for LDA models and Eq. (19) for our STC models.

5.1.2 Experiments on 99K subset

To compare with OLDA, we follow the same settings in [16] and randomly choose a 99K subset of the whole Wikipedia data. Fig. 4(a) shows the perplexity of OSTC (with batch size $M = 64$), batch STC and OLDA ($M = 64$). We can see that OSTC converges much faster than batch STC because of its effective exploration of document redundancy. We also observe that OSTC has a lower perplexity than OLDA. The main reason is that STC uses un-normalized word codes, which offer an additional freedom compared to the normalized probability in LDA. This extra freedom could lead to better fitness of the observed data.

[5]Since τ_0 may affect the convergence speed, we tune τ_0 for the best performance. Similar as in [16], we set a smaller τ_0 for a larger batch size.

[6]We use the authors' implementation:
http://www.cs.princeton.edu/∼blei/downloads/onlineldavb .tar

[7]Although sampling methods for LDA often result in sparse topic representations due to the limited number of samples, both LDA and OLDA are not sparse models. In contrast, both STC and OSTC are sparse due to a soft-thresholding operators as presented in Section 3.3.

[8]We tried K=100, 150, 200 and found no big difference in held-out perplexity.

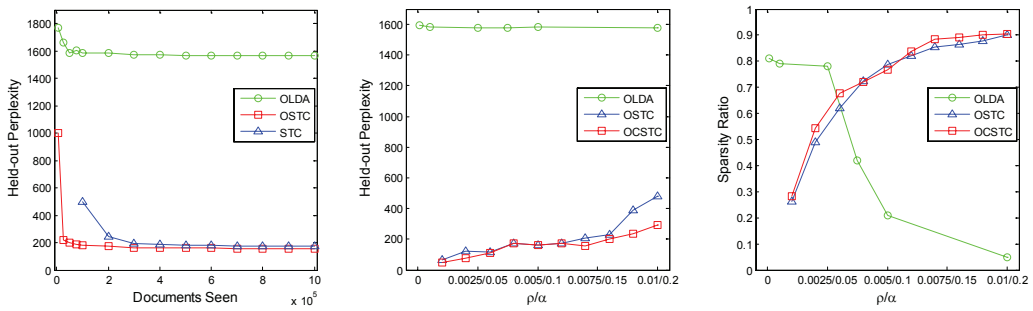


Figure 4: (a) held-out perplexity of STC, online STC and online LDA on the 99K Wikipedia dataset; (b,c) perplexity and sparsity of OSTC and OLDA when the hyper-parameters ρ and α change.

Table 1: Perplexity of LDA, CG-LDA and STC on two datasets.

	LDA	CG-LDA	STC
Wikipedia	1609.16	1503.85	265.37
20Newsgroups	5656.38	4847.65	1588.59

To examine the influence of approximate inference algorithms on perplexity, Table 1 further compares the perplexity of STC with those of the LDA models using variational mean-field as well as the collapsed Gibbs sampling [15]. We denote the LDA using collapsed Gibbs sampling by CG-LDA. We can see that although using collapsed Gibbs sampling can improve the performance of LDA, its perplexity is still significantly higher than that of STC.

Fig. 4(b) and Fig. 4(c) further compare the held-out perplexity and word code sparsity of OSTC and OLDA when their hyper-parameters change. Both models have a single pass on the 99K subset. For OSTC, we fix $\lambda = \gamma = 0.025$ and only change ρ (changing both ρ and γ will lead to even better results), and for OLDA, the hyper-parameter is the Dirichlet parameter α. We can see that for both models, the hyper-parameter affects the word code sparsity much. But for OLDA, the held-out perplexity doesn't change much, all remaining at a level of about 1,600. In contrast, ρ affects much on the perplexity of OSTC. At all points, OSTC obtains a smaller perplexity than OLDA. Moreover, when ρ is set at a relatively large value (e.g., 0.01), OSTC obtains much lower perplexity and higher word code sparsity. Our observations are consistent with those in [42, 21], whose experiments demonstrate the effectiveness of STC on discovering sparse (and interpretable) topical representations.

We also investigate the performance of collapsed STC using online learning. From Fig. 4(b) and Fig. 4(c), we can see that the collapsed OSTC (i.e., OCSTC) outputs slightly sparser word codes and achieves even lower perplexity than OSTC, when both methods using the same hyper-parameters. This performance gain comes from relaxation of conditional independence constraints in the inference step.

5.1.3 *Experiments on 6.6M Wikipedia corpus*

Now, we use the whole 6.6M Wikipedia dataset to examine the scalability of OSTC. Fig. 5 shows the perplexity of OSTC with different batch sizes, as a function of the running time. We can see that the convergence speeds of different algorithms vary[9]. First, since batch algorithm suffers from writing disk operations due to its huge memory cost[10], its performance is much worse than those of the online alternatives. Second, online algorithms with medium batch sizes (e.g., $M = 256$) converge faster than others. When we use a too small batch size (e.g., $M = 4$), it takes a long time to converge because we update the dictionary too frequently in each iteration without enough evidence. Finally, we also note that as the batch size becomes too large (e.g., $M = 4096$), the convergence speed of online algorithm approaches the very slow batch algorithm.

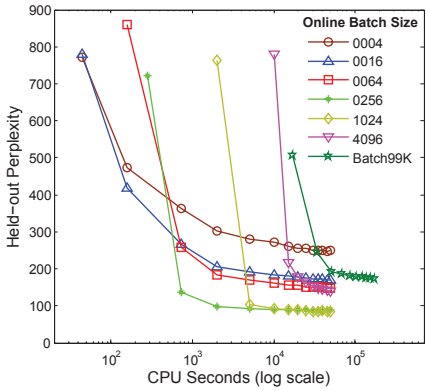


Figure 5: held-out perplexity of online STC using different batch sizes on the whole 6.6M Wikipedia dataset.

5.2 Experiments on 20Newsgroups Dataset

The 20Newsgroups dataset consists of 18,774 documents from 20 different newsgroups with a standard train/test split[11] of 11,269/7,505. The vocabulary contains 61188 terms, and we remove a standard list of 524 stop words as in [42].

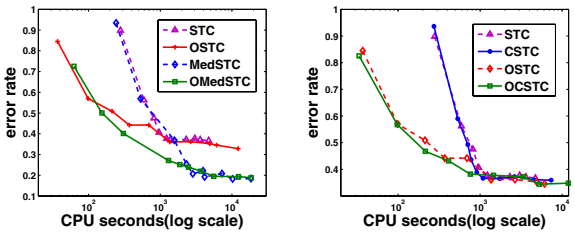
[9] Almost all the OSTC models with different batch sizes converge before scanning the whole corpus.

[10] If we use float type and assume each document has on average 100 words, we will need about 4GB memory to store the word codes for the 99K subset when $K = 100$. For the 6.6M dataset, we will need about 250GB.

[11] http://people.csail.mit.edu/jrennie/20Newsgroups/

Table 2: Classification accuracy of LDA, STC and MedSTC on the 20Newsgroups dataset.

	LDA		STC		MedSTC	
batch size	accuracy(%)	time(ks)	accuracy(%)	time(ks)	accuracy(%)	time(ks)
1	52.3	61.2	53.1	41.1	65.3	44.4
8	58.3±1.4	17.9	64.7±1.2	7.0	80.0	14.1
16	60.5±0.7	8.5	66.1±0.7	3.9	81.2	12.3
32	**61.7±0.7**	6.2	**66.3±1.0**	2.7	80.5	**8.8**
64	60.9±0.9	**4.0**	65.2±1.6	**2.2**	81.3	10.9
batch	61.4±0.7	8.6	62.7±0.6	4.7	**81.6**	18.4



(a) (b)

Figure 6: (a) error rates of STC and MedSTC as a function of running time; (b) error rates of STC and CSTC as a function of running time.

In these experiments, we focus on comparing both time efficiency and test accuracy between STC and online STC with different batch sizes. The results of other supervised topic models, including MedLDA and sLDA, were reported in [42]. We choose the parameters $K = 60$, $\Delta(y, y') = 3600\mathbb{I}(y \neq y')$, $\rho = 0.1$ and $\lambda = \gamma = 0.01$, which produce good results as shown in [42].

Table 2 presents the classification accuracy of different models with different batch sizes. We can observe that the online STC obtains higher accuracy while with less running time than the online LDA using the same batch size. For STC, online learning algorithms generally improve the time efficiency in order to get a good classification model. For instance, the online STC with a batch size of 32 takes about a half of the running time of the batch STC, and its classification performance is surprisingly much better; for MedSTC, when the batch size is 16, the online MedSTC performs comparably with the batch MedSTC, while taking less running time. We also observe that batch sizes can affect the convergence and classification performance of various online topic models. The reason is that too small batches update $\boldsymbol{\beta}$ slowly since $\boldsymbol{\beta}$ is high dimensional, while large batches tend to reach another extreme of being ineffective in exploring data redundancy.

Fig. 6(a) shows the error rates of STC and MedSTC, using both batch and online learning algorithms, as a function of running time. We can see that by cycling on the medium-sized 20Newsgroups dataset, the online algorithms generally reach a good model faster than the batch algorithms. In the unsupervised setting, the online algorithm performs better both in time and classification accuracy. As has been demonstrated on the Wikipedia articles, we can expect large improvements in a much larger and redundant corpus.

Then we report the evaluation of the collapsed STC on the 20Newsgroups dataset for prediction performance, again using both batch and online learning algorithms. Fig. 6(b) presents the error rates as a function of running time. We can see that the online learning algorithms generally converge faster to fairly good results. But the collapsed STC does not shows dramatic improvements compared with STC. This is probably due to the fact that the problem of STC can be solved very well on the this dataset using the coordinate descent algorithm with a hierarchical sparse coding, and the collapsed sparse coding does not help a lot.

Finally, to examine the semantics of the learned topics, Table 3 presents top words (i.e., words that have highest values in the topic) of the most salient topic learned by the online MedSTC for each category (i.e., topic that has highest value in the average document code of each category) on the 20Newsgroups dataset. We can generally see the strong association of the categories and the learned topics.

6. CONCLUSIONS AND DISCUSSIONS

We have presented a sparse online topic model for modeling dynamic text streams and discovering topic representations from large-scale datasets. The online dictionary learning algorithm is efficient and guaranteed to converge. Extensive empirical studies on Wikipedia and 20Newsgroups data have shown appealing performance in terms of held-out perplexity, word code sparsity and prediction accuracy.

For future work, we are interested in various extensions and improvements, including cleverly adjusting the learning rates during learning and dealing with large-scale complex data analysis problems, such as relational network analysis.

7. ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program (973 Program) of China (Nos. 2013CB329403, 2012CB316301), National Natural Science Foundation of China (Nos. 91120011, 61273023), and Tsinghua University Initiative Scientific Research Program (No. 20121088071).

8. APPENDIX

An alternative way to compare STC and LDA

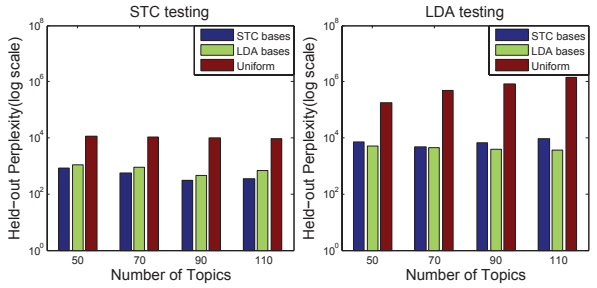
Due to different definitions, more careful analysis should be done on comparing the perplexity between STC and LDA. We now do an interesting 'interchange' experiment. The idea is that although the inference procedure is different between STC and LDA, they both learn normalized topical bases (i.e. the dictionary). So we can turn to test the quality of their bases to see whether one model is strictly better than the other. To do this we first train bases with each model and then calculate the STC held-out perplexity and the LDA held-out perplexity using both bases by Eq. (19) and Eq. (18) separately. For example, we can use STC for training bases (STC bases) and LDA for calculating held-out perplexity (LDA testing). As an upper bound, we

Table 3: Example topics learned by OMedSTC. For each category, we show the most salient topic.

comp.					misc.	talk.			
graphics	ms-windows	ibm.pc	mac	windows.x	forsale	politics.misc	politics.guns	politics.mideast	religion.misc
compass	allocation	dma	gnd	widget	trade	mov	gun	cosmo	incoming
cols	windows	drive	init	entry	msdos	hitler	cranston	power	taoism
rows	yap	aspi	vv	libx	bid	time	guns	erzurum	allocation
graphics	cfg	wires	applelink	xsizehints	toshiba	stephanopoulos	militia	armenian	aleph
rtheta	mywinobj	compaq	mac	libxmu	laptop	viability	people	turks	jesus
ellipse	vb	harddisk	apple	converter	baud	government	weapons	negotiations	bible
sphinx	dos	isa	nubus	accelerators	modem	throws	firearms	turkish	objective
image	file	scsi	backlit	decnet	mpc	chancellor	fire	bayonet	morality
files	bitmap	card	wolves	focus	coupons	president	fbi	labor	christ
color	files	pc	drive	myhint	send	african	law	armenians	christian

sci.				rec.				alt.	soc.
crypt	electronics	med	space	autos	motorcycles	baseball	hockey	atheism	christian
mov	pin	jl	ics	car	gun	roster	pt	contradictory	babylon
nffutils	compass	hiv	incoming	writes	bike	lefthanded	period	rapist	god
maxbyte	tesla	polio	het	tint	zephyr	baseball	switzerland	god	pentecostals
db	hook	oily	space	article	teflon	idle	italy	depression	husband
nist	wire	spect	nasa	carburetor	dog	year	aids	writes	jesus
push	brightness	methanol	launch	lojack	shaft	team	norway	people	senses
offset	doherty	tinnitus	orbit	cars	ride	ball	czech	don	ceremonial
trinomials	power	eye	moon	vw	good	game	austria	allah	people
encryption	blinker	patients	earth	good	hawk	players	qtr	article	christian
key	circuit	msg	shuttle	volvo	back	pitching	game	islam	church

also report the results by using the non-informative uniform basis. Experimental results using different number of topics on the 20Newsgroups dataset are shown below.



(a) (b)

Figure 7: (a) STC testing perplexity for different bases; (b) LDA testing perplexity for different bases.

The left figure shows held-out perplexity by STC using Eq. (19) and the right one shows held-out perplexity by LDA using Eq. (18). Each figure compares among bases learned by both models and the uniform bases (as a baseline). The red bar shows the perplexity calculated by uniform bases as an upper bound. Obviously, both STC and LDA learn meaningful bases and their held-out perplexity is significantly lower than the perplexity produced by the uniform bases (In both figures we use log scale for the perplexity axis.). In the left figure when we calculate held-out perplexity by STC, we achieve a lower perplexity by using STC bases. However, LDA bases get a lower perplexity in the other setting in the right figure. Thus, using the same model for training and testing achieves better results. The bases learned by other models can be useful, but not as accurate as the original one. Finally, we also note that in general, we get lower perplexity when using STC for testing.

9. REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, (9):1981–2014, 2008.

[2] A. Asuncion, M. Welling, P. Smyth, and Y. Teh. On smoothing and inference for topic models. In *Conference on Uncertainty in Artificial Intelligence*, pages 27–34, 2009.

[3] D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems*, pages 147–154, 2005.

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

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

[6] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *International Conference on Machine Learning*, pages 113–120, 2006.

[7] L. Bottou. *Online Learning and Stochastic Approximations*, chapter On-line learning in neural networks. 1998.

[8] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

[9] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1024–1033, 2007.

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

[11] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In *International Conference on Machine Learning*, pages 272–279, 2008.

[12] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, pages 524–531, 2005.

[13] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1816–1823, 2005.

[14] W. Fu, J. Wang, Z. Li, H. Lu, and S. Ma. Learning semantic motion patterns for dynamic scenes by improved sparse topical coding. In *International Conference on Multimedia and Expo*, pages 296–301, 2012.

[15] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, (101):5228–5235, 2004.

[16] M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 156–164, 2010.

[17] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.

[18] P. Hoyer. Non-negative sparse coding. In *IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[19] A. Hyvarinen. Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, (11):1739–1768, 1999.

[20] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *Conference on Knowledge Discovery and Data Mining*, pages 663–672, 2010.

[21] R. Ji, L. Duan, J. Chen, and W. Gao. Towards compact topical descriptors. In *Conference on Computer Vision and Pattern Recognition*, pages 2925–2932, 2012.

[22] J. J. Kivinen, E. B. Sudderth, and M. I. Jordan. Learning multiscale representations of natural scenes using Dirichlet processes. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[23] D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788 – 791, 1999.

[24] H. Lee, R. Raina, A. Teichman, and A. Ng. Exponential family sparse coding with applications to self-taught learning. In *International Joint Conferences on Artificial Intelligence*, pages 1113–1119, 2009.

[25] L.-J. Li, J. Zhu, H. Su, E. Xing, and L. Fei-Fei. Multi-level structured image coding on high-dimensional image representation. In *Asian Conference on Computer Vision*, 2012.

[26] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2008.

[27] D. Mimno, M. Hoffman, and D. Blei. Sparse stochastic inference for latent Dirichlet allocation. In *International Conference on Machine Learning*, 2012.

[28] D. Mimno, H. Wallach, J. Naradowsky, D. A. Smith, and A. McCallum. Polylingual topic models. In *Conference on Empirical Methods in Natural Language Processing*, pages 880–889, 2009.

[29] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[30] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Conference on Uncertainty in Artificial Intelligence*, pages 487–494, 2004.

[31] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1605–1614, 2006.

[32] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning*, pages 807–814, 2007.

[33] M. Shashanka, B. Raj, and P. Smaragdis. Sparse overcomplete latent variable decomposition of counts data. In *Advances in Neural Information Processing Systems*, pages 1313–1320, 2007.

[34] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their locatioins in images. In *IEEE International Conference on Computer Vision*, pages 370–377, 2005.

[35] S. Sra, D. Kim, and B. Schölkopf. Non-monotonic Poisson likelihood maximization. *Tech. Report, MPI for Biological Cybernetics*, 2008.

[36] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1353–1360, 2007.

[37] C. Wang, D. Blei, and L. Fei-Fei. Simultaneous image classification and annotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1903–1910, 2009.

[38] Q. Wang, J. Xu, H. Li, and N. Craswell. Regularized latent semantic indexing. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 685–694, 2011.

[39] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Conference on Knowledge Discovery and Data Mining*, pages 937–946, 2009.

[40] K. Zhai, J. Boyd-Graber, N. Asadi, and M. Alkhouja. Mr. LDA: A flexible large scale topic modeling package using variational inference in MapReduce. In *Proceedings of World Wide Web Conference*, pages 879–888, 2012.

[41] J. Zhu, A. Ahmed, and E. Xing. MedLDA: Maximum margin supervised topic models for regression and classification. In *International Conference on Machine Learning*, pages 1257–1264, 2009.

[42] J. Zhu and E. Xing. Sparse topical coding. In *Conference on Uncertainty in Artificial Intelligence*, pages 831–838, 2011.