Group Sparse Topical Coding: From Code to Topic

Lu Bai Jiafeng Guo Yanyan Lan Xueqi Cheng
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{bailu, guojiafeng}@software.ict.ac.cn, {lanyanyan, cxq}@ict.ac.cn

ABSTRACT

Learning low dimensional representations of text corpora is critical in many content analysis and data mining applications. It is even more desired and challenging to learn a sparse representation in practice for large scale text modeling. However, traditional probabilistic topic models (PTM) lack a mechanism to directly control the posterior sparsity of the inferred representations; While the emerged non-probabilistic models (NPM) can explicitly control sparsity using sparse constraint like ℓ_1 norm, they convey different limitations in latent representations. To address the existing problems, we propose a novel non-probabilistic topic model for discovering sparse latent representations of large text corpora, referred as *group sparse topical coding* (GSTC). Our model enjoys both the merits of the PTMs and NPMs. On one hand, GSTC can naturally derive document-level admixture proportions in topic simplex like PTMs, which is useful for semantic analysis, classification or retrieval. On the other hand, GSTC can directly control the sparsity of the inferred representations with *group lasso* by relaxing the normalization constraint. Moreover, the relaxed non-probabilistic GSTC can be effectively learned using coordinate descent method. Experimental results on benchmark datasets show that GSTC can discover meaningful compact latent representations of documents, and improve the document classification accuracy and time efficiency.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Algorithms, Experimentation, Performance, Theory

Keywords

Document Representation, Topic Model, Sparse Coding, Group Lasso

WSDM'13, February 4–8, 2013, Rome, Italy.

1. INTRODUCTION

With huge repositories of text data emerged over Web, learning low dimensional representations that captures latent semantics of the data becomes critical for efficiently processing many basic tasks such as classification, clustering, and information retrieval. It is even more desired and challenging to learn a sparse representation in practice for large scale text modeling [27, 31]. For example, it is very often that there are hundreds or thousands of topics within hundreds of millions of documents, while each document or word has only a few salient topical meanings or senses [27, 31]. Such a phenomenon is even more typical in large collections of short text data, like instant messages or tweets. By explicitly learning a sparse representation, we may better discover the salient semantic meanings of a document or word. Moreover, sparse representations have obvious computational benefits, by saving both processing time and storage space [38].

In Recent years, probabilistic topic models (PTM) such as probabilistic latent semantic indexing (PLSI) [16] and latent dirichlet allocation (LDA) [5] have gained remarkable success in text topical modeling. In these models, a document is typically modeled as a finite admixture of latent topics where each topic is a unigram distribution over a given vocabulary. The document-level admixture proportions can be regarded as a low dimensional representation of the document in the topic simplex. However, traditional PTMs lack a mechanism to directly control the posterior sparsity of the inferred representations, since the admixture proportions or topics are normalized distributions. Some attempts have been made to achieve sparsity in PTMs by introducing sparse priors [5] or using posterior regularizations [14, 27]. These methods often cannot yield truly sparse posterior representations due to indirect sparsity bias or the smoothness of the regularizer.

Meanwhile, the emerged non-probabilistic models (NPM) such as matrix factorization [20, 9] and sparse coding [23, 40], provide an elegant way to achieve sparsity by using sparse constraint like ℓ_1 norm [29] or other composite regularizer [33, 30] on the usually unnormalized latent representations. However, while getting the benefits from using the unnormalized latent representations, NPMs lose the clear semantic explanations over the latent representations. That is, the learned document-level latent representations no longer lie on the same topic simplex like PTMs, but could be in quite different scales over different documents. This will lead to difficulty in semantic comparison between documents, and in using the latent representations as robust features for future tasks (e.g., text classification) as shown in our work. Although post-normalization could be conducted over the latent representations of an entire document, the resulted normalized representations no longer represent the admixture proportions.

To address the above problems, we propose a novel non-probabilistic

topic model that can discover sparse latent representation (or admixture proportions) of the document, referred as *group sparse topical coding* (GSTC). In our model, each individual word count is consisted of a set of latent word counts, where each latent word count is generated from a Poisson distribution. Thus, we can *reconstruct* the individual word count from a linear combination of topic bases, where the coefficient vectors (i.e. codes) are unnormalized, and the topic bases are unigram distributions over a given vocabulary. We impose group lasso [38] over the unnormalized word codes to achieve the sparsity of the document-level latent representations. The admixture proportion of an entire document can be derived from the learned word codes and topic bases.

Our model enjoys both the merits of the PTMs and NPMs as follows: 1) by using Poisson distribution, GSTC can naturally derive document-level admixture proportions in topic simplex like PTMs; 2) as a non-probabilistic model, GSTC can directly control the sparsity of inferred representations by taking into account the structure of bag of words with group lasso; 3) by relaxing the normalization constraint of admixture proportions, GSTC can be effectively learned using coordinate descent method. All these nice properties make GSTC an appealing alternative formulation of topic models.

We conducted empirical experiments on benchmark datasets and compared GSTC with the state-of-the-art PTMs (i.e. LDA)and NPMs (i.e. NMF and STC). The experimental results show that GSTC can identify sparse and salient topical representations over documents. Moreover, the experiments demonstrate that with the compact representations of documents learned by GSTC, we can largely improve the document classification accuracy and obtain better time efficiency.

2. RELATED WORK

In this section, we will introduce some recent related work on probabilistic topic models, non-probabilistic topic models and sparse techniques.

2.1 Probabilistic Topic Models

Due to the theoretical soundness and explicit interpretation, probabilistic topic models(PTMs) has been widely used for data analysis in various applications such as such as information retrieval [34], network analysis [1], and recommender systems [32]. PTMs provide a clear probabilistic interpretation of the generative process of data. PTMs such as Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) have shown impressive empirical success in practise through the discovery of low-rank hidden structure. Typically, both PLSA and LDA posit that each document is an admixture of latent topics where each topic is a unigram distribution over the terms in a vocabulary. There are quite a lot of work extending these basic probabilistic models by incorporating the auxiliary knowledge as constraints or priors. Cai et al. [7] proposed the Laplacian pLSI(LapPLSI) by incorporating the manifold structure information to smooth the probabilistic density functions. Blei et al. [4] adopted the logistic normal distribution, a more flexible distribution for modeling covariance structure among components, to generate the documents.

However, how to achieve sparsity is a non-trivial problem for PTMs. The major reason lies in the fact that the admixture proportions and topics are normalized distributions in PTMs, leading to the failure of directly applying a sparsity inducing ℓ_1 regularizer as in lasso [29]. Although some alternative methods have been proposed to control sparsity in PTMs , these sparse-biased PTMs are not good in either effects or efficiency. For example, using a weak Dirichlet prior [5] may indirectly introduce some sparse bias over posterior representations. However, the sparsity is mostly resulted from the scarcity of document content and the control of sparsity in this way is very limited. Selecting subset of topics to construct documents is another common strategy [35, 31]. Usually a binomial prior is employed to decide which topics to be selected. Although these models are good in interpretation, the learning might be quite unstable. The major reason lies in the discrete generative process - topic components are either reserved or dropped from models. In this way, small changes in data or parameters can result in very different models. What is more, the optimization for these models is often hard to solve [39, 24]. Some studies tried to directly impose posterior regularizations [14, 27], which often do not obtain truly sparse posterior representations in practice due to the smoothness of the regularizer (e.g., entropic regularizer).

2.2 Non-Probabilistic Topic Models

As an alternative, NPMs such as sparse coding [23] and matrix factorization [20] provide an elegant framework to achieve sparsity. With usually unnormalized code vector or dictionary, NPMs can directly control the sparsity with ℓ_1 norm or other composite regularizer [41, 30]. However, while getting the benefits from the unnormalized codes, NPMs lose the clear semantic explanations over the latent representations of documents. For example, most sparse coding methods [3, 18] only learn flat representations (i.e. the code at word level). One may take post-processing like average or max pooling [36] to achieve representations at document level, but the meaning of the averaged code vector is not clear. Although sparse topical coding (STC) [40] learns a hierarchical topical representations, the document codes (truncated averaging from the word codes) still lacks the semantic interpretation. Besides, matrix factorization methods (e.g. NMF [20] and LSI [9]) learn the latent representations of documents, which are usually unnormalized or even non-positive. (e.g. RLSI [33]). Mapping such learned representations to a topic simplex is not a trivial task. Unlike the existing methods, GSTC is a novel non-probabilistic topic model that can discover sparse admixture proportions of documents.

2.3 Sparse Techniques

The lasso [29] is a widely applied technique to produce sparse model by penalizing the loss function with a ℓ_1 norm. Increasing the ℓ_1 penalty will make more parameters be drawn to zeros instead of reducing all parameters simultaneously as the ℓ_2 norm does. Furthermore, in contrast with the ℓ_0 norm, the ℓ_1 norm is obviously convex. Therefore, the resulting optimization will be convex when the loss function is convex, which can be effectively solved by a variety of optimization algorithms [21, 11]. Actually, lasso can recover the sparse supports of a sparse model from data when the covariates of the model are not too related. Elastic net, a new regularization of the lasso, was further proposed by Zou et al. [41] for an unknown group of variables and for multicollinear predictors. It can be taken as a stabilized version of the lasso, which enjoys a sparse representation and encourages group effects.

However, the lasso lacks a mechanism to contain any prior information about the variables, e.g. some groups of variables should be picked jointly. Some efforts have been made to enforce the estimated models maintain the special sparsity patterns. Especially, group lasso [38] extends the lasso by making the correlated variables selected jointly rather than individual variables. It achieves this by applying a mixed ℓ_1/ℓ_2 norm (the sum (ℓ_1 norm) of the ℓ_2 norms) over different groups of variables. In this way, the predefined sparse pattern is preserved by selecting the group of variables with ℓ_2 norm[37]. Apparently, the group lasso constraint is still convexity, which is a desirable property for developing an efficient algorithm to solve the optimization problem. Other variants of the group

lasso are applied in selecting covariates in multi-task learning and enforcing the hierarchical selection of covariates, e.g. covariates are located in a hierarchical structure and one variable is selected if and only if its ancestors are selected as well. Recently, this ℓ_1/ℓ_2 norm for group lasso has been extended to model a more general settings where the groups can be overlapped and nested [17].

3. GROUP SPARSE TOPICAL CODING

GSTC aims to discover document-level sparse latent representation (or admixture proportion) for large scale text corpora, which can be regarded as a sparse representation of the document in the topic simplex. Suppose we are given a collection of documents $\boldsymbol{D}$ with size M, containing words from a vocabulary $\boldsymbol{V}$ with size N. A document is simply represented as a $|I|$-dimension vector $\boldsymbol{d} = \{w_1, ..., w_{|\boldsymbol{I}|}\}$, where $\boldsymbol{I}$ is the index set of words that appear and the n-th entry w_n ($n \in \boldsymbol{I}$) denotes the number of appearances of the specific word in this document. Let $\boldsymbol{\beta} \in \mathbb{R}^{K\times N}$ be a dictionary with K bases, where each base is assumed to be a topic base, i.e. a unigram distribution over $\boldsymbol{V}$. For a given document $\boldsymbol{d}$, GSTC projects $\boldsymbol{d}$ into a semantic space spanned by a set of automatically learned topic bases $\boldsymbol{\beta}$ and directly obtain the unnormalized word code $\boldsymbol{s}_{d,n\cdot} \in \mathbb{R}^K$ for each individual word in document $\boldsymbol{d}$. For simplicity, we use $\boldsymbol{s}_{n\cdot}$ to describe the word code in current document. The admixture proportion of the entire document $\boldsymbol{d}$ can then be derived from the learned word code set $\boldsymbol{s} = \{\boldsymbol{s}_{1\cdot}, ..., \boldsymbol{s}_{|\boldsymbol{I}|\cdot}\}$ and the topic bases $\boldsymbol{\beta}$. To better understand the proposed GSTC model, we start with describing a probabilistic generative procedure.

3.1 Probabilistic Generative Process for GSTC

The basic idea of GSTC is that the appearances of each individual word in a document come from a set of latent topics, where each topic is characterized by a distribution over words. The graphical representation of GSTC is depicted in Figure 1. We first sample a dictionary $\boldsymbol{\beta}$ from a prior distribution (e.g. uniform distribution in this work) on a $(N-1)$-simplex. Then, each document $\boldsymbol{d}$ in a collection $\boldsymbol{D}$ is assumed to be generated by the following process:

1. For each topic $k \in \{1, \ldots, K\}$:

   Sample a word code vector $\boldsymbol{s}_{\cdot k} \in \mathbb{R}^N \sim$ M-Laplace(λ).

2. For each observed word $n \in \boldsymbol{I}$:

   For each topic $k \in \{1, \ldots, K\}$:

   Sample a latent word count $w_{nk} \sim$ Poisson($\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}$).

3. Obtain the word count $w_n = \sum_{k=1}^K w_{nk}$.

Several simplifying assumptions are made in our model. We assume that for each document the observed word counts are independent given their latent representations $\boldsymbol{s}$ [40]. We also assume that the latent word counts of a given word over different topics are independent given its latent representation $\boldsymbol{s}$. In the above generative process, the word code vectors under each topic are first generated from a Multi-Laplacian distribution, which is defined as:

$$\text{M-Laplace}(\boldsymbol{s}|0, \lambda^{-1}) \propto \lambda^{N/2} \exp(-\lambda \parallel \boldsymbol{s} \parallel_2). \tag{1}$$

For each individual word appeared in the document, the observed word count w_n is obtained by aggregating the latent word counts over different topics, where each latent word count is generated from a Poisson distribution, that is defined as

$$\text{Poisson}(w_{nk}|s_{nk}\beta_{kn}) \propto (\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn})^{w_n}(e^{-s_{nk}\beta_{kn}}) \tag{2}$$

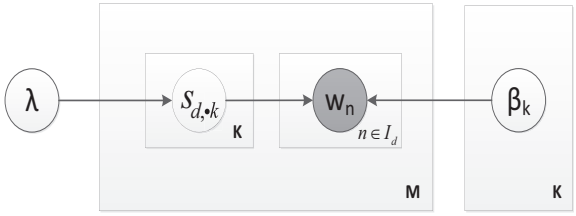

Figure 1: The graphical model of group sparse topical coding.

We choose to use Multi-Laplacian distribution to achieve sparse codes at the document level by taking into account the structure of bag of words. Similar to the relationship between the ℓ_1 norm and Laplace distribution, the multi-Laplacian prior over each group of coefficients turns the classical group-lasso constraints into a MAP-solution in the log-space. The parameter λ in formula 1 has the same role of a fixed Lagrange parameter that deals with the concentration of the probability mass towards zero. Unlike STC which introduces a document code $\boldsymbol{\theta}$, we only model the word code in GSTC and view the document code as a natural by-product, e.g. an simple aggregation of the individual codes of all its words on each topic. To achieve document-level sparse representation, we may expect that there are only a set of topics that convey positive aggregated codes, while others are zero (i.e. all the word codes under that topic are zero). This could be achieved by grouping the word codes under the same topic together and sample the groups of variables from a Multi-laplacian distribution [12, 25]

Moreover, we adopt the Poisson distribution with the *mean* parameter $\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}$ to generate the latent word count from the k-th topic w_{nk} due to the following reasons. 1) Since observed word counts as well as latent word counts are non-negative discrete number, it is natural to use Poisson distribution to model such variables. 2) We use the combination as mean parameter to naturally constrain the feasible domains (i.e. non-negative) of word codes for good interpretation. Moreover, imposing non-negativity constraints could potentially result in sparser and more interpretable patterns [20]. 3) Last but not least, Poisson distribution has two nice properties which plays a critical role in GSTC, namely additive property and Moran's property. The additive property says that if $X_1 \sim \text{Poisson}(\tau_1)$ and $X_2 \sim \text{Poisson}(\tau_2)$ are independent, then $X_1 + X_2 \sim \text{Poisson}(\tau_1 + \tau_2)$. Based on additive property, we have that the word count w_n follows a Poisson distribution with the mean parameter defined as a linear combination of topic bases $\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n}$, since the latent word counts over different topics are assumed to be independent. Similar methods have been used to define exponential family distributions [22, 6, 40]. The Moran's property further helps us derive the admixture proportions based on the learned word codes and topic bases, which will be discussed later.

3.2 Formulation of GSTC

For a document $\boldsymbol{d}$, the above generative process defines a joint distribution of word codes and word counts shown as below.

$$\begin{aligned} p(\boldsymbol{s}, \boldsymbol{w}|\boldsymbol{\beta}) &= \prod_{k=1}^K p(\boldsymbol{s}_{\cdot k}) \left(\prod_{n=1}^{|\boldsymbol{I}_d|} \prod_{k=1}^K p(w_{nk}|s_{nk}, \beta_{kn}) \right) \\ &= \prod_{k=1}^K p(\boldsymbol{s}_{\cdot k}) \prod_{n=1}^{|\boldsymbol{I}_d|} p(w_n|\boldsymbol{s}_{n\cdot}, \boldsymbol{\beta}), \end{aligned} \tag{3}$$

where $p(\boldsymbol{s}_{\cdot k}) = \text{M-Laplace}(\boldsymbol{s}_{\cdot k}|0, \lambda^{-1})$, $p(w_n|\boldsymbol{s}_{n\cdot}, \boldsymbol{\beta}) = \text{Poisson}(w_n; \boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n})$ The hyper-parameter λ is non-negative which can be selected via cross-validation or integrated out by introducing hyper-priors [19,

25]. The second equation is derived according to the additive property of Poisson distribution.

Now, we formally define GSTC as finding the MAP estimate of the above probabilistic model. Given a collection of documents $\boldsymbol{D}$ with size M, GSTC solves the following optimization problem:

$$\begin{aligned}
\min_{\boldsymbol{\Theta},\boldsymbol{\beta}} \mathcal{L}(\Theta,\boldsymbol{\beta}) &= -\ln P(\Theta,\boldsymbol{\beta}|D) \\
&= \min_{\boldsymbol{\Theta},\boldsymbol{\beta}} \sum_{d=1}^{M}\sum_{n=1}^{|\boldsymbol{I}_d|} \ell(\boldsymbol{s}_{d,n\cdot},\boldsymbol{\beta}) + \sum_{d=1}^{M}\sum_{k=1}^{K} \lambda\|s_{d,\cdot k}\|_2 + C \\
&= \min_{\boldsymbol{\Theta},\boldsymbol{\beta}} \sum_{d=1}^{M}\sum_{n=1}^{|\boldsymbol{I}_d|} \left(\sum_{k=1}^{K} s_{d,nk}\beta_{kn} - w_{d,n}\ln(\sum_{k=1}^{K} s_{d,nk}\beta_{kn}) \right) \\
&\quad + \sum_{d=1}^{M}\sum_{k=1}^{K} \lambda\|s_{d,\cdot k}\|_2 + C, \\
s.t. \quad & \boldsymbol{s}_{d,n\cdot} \geq 0, \forall d, n \in \boldsymbol{I}_d, \\
& \sum_{n=1}^{N} \beta_{kn} = 1, \forall k, \qquad (4)
\end{aligned}$$

where the loss function $\ell(\boldsymbol{s}_{d,n\cdot},\boldsymbol{\beta}) = -\log \text{Poisson}(\boldsymbol{s}_{d,n}^{\top}\boldsymbol{\beta}_{\cdot n})$, $\Theta = \{\boldsymbol{s}_{d,\cdot\cdot}|\boldsymbol{d} \in \{1,\cdots,M\}\}$, C is a constant. It is shown that minimizing the log-Poisson loss is actually equivalent to minimizing an unnormalized KL-divergence between observed word counts $w_{d,n}$ and their reconstructions $\boldsymbol{s}_{d,n}^{\top}\boldsymbol{\beta}_{\cdot n}$ [28].

Although the objective function of GSTC is derived from a probabilistic perspective, it can be directly interpreted in a non-probabilistic way. Like the ridge regression, the first part of formula 4 is the reconstruction error for the observed word counts in terms of dictionary $\boldsymbol{\beta}$. The second parts is a Group-lasso, i.e. a mixed ℓ_1/ℓ_2 norm, for the matrix of reconstruction coefficients. This form of regularization promotes solutions that are sparse at the level of *groups* of variables, that is, an entire word code vector under a topic may drop out of the model. This in turn achieves the desired document-level sparse coding.

By optimizing the objective function $\mathcal{L}(\Theta,\boldsymbol{\beta})$, we can learn all the word codes in the document collection as well as the dictionary. As aforementioned, the goal of GSTC is to discover document-level sparse admixture proportions, which can be regarded as a low dimensional representation of the document in the topic simplex. Here we show how to derive the document-level admixture proportion with the learned word codes and dictionary.

As traditional topic models, GSTC assumes that the occurrences of a word in a document may be generated from different topics. Therefore, in GSTC the observed word count w_n is consisted of a set of latent word counts $\{w_{nk}\}_{k=1}^{K}$. If we have known exactly which topic each word occurrence comes from, i.e. the latent word counts $\{w_{nk}\}_{k=1}^{K}, \forall n \in \boldsymbol{I}$ are observed, the document-level admixture proportion would be simply a normalization over the aggregated occurrence counts under different topics. However, since the latent word counts are not observed, we calculate the expected admixture proportion for the document instead.

For this purpose, we first introduce the Moran's property of Poisson distribution.

Lemma 1. ***[Moran's Property of Poisson Distribution]*** *If variables $X_1, X_2, \cdots, X_n$ are independent Poisson random variables with parameters $\tau_1, \tau_2, \cdots, \tau_n$, then*

$$X_i|\sum_{j=1}^{n} X_j \sim Binom\left(\sum_{j=1}^{n} X_j, \frac{\tau_i}{\sum_{j=1}^{n}\tau_j}\right).$$

With the help of Moran's property, the document proportion can be directly derived in GSTC, as shown in the following theorem.

Theorem 1. *Let θ be the topic proportion vector of document d. Assume the document is generated as described in section 3.1, we will have the kth topic proportion $\theta_k = \frac{\sum_{n=1}^{|\boldsymbol{I}|}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}$.*

Proof. Recall that in GSTC, each w_{nk} is independently generated from a Poisson distribution Poisson($\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}$) , according to the Moran's property we have:

$$w_{nk}|\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk} \sim \text{Binom}\left(\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk}, \frac{\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}\right).$$

Therefore, the expected number of word occurrences from the k-th topic can be expressed as:

$$\begin{aligned}
\mathbb{E}[\textstyle\sum_{n=1}^{|\boldsymbol{I}|} w_{nk}|\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk}] &= \sum_{n=1}^{|\boldsymbol{I}|}\mathbb{E}[w_{nk}|\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk}] \\
&= \sum_{n=1}^{|\boldsymbol{I}|}\left(\left(\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk}\right)\frac{\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}\right) \\
&= \left(\sum_{n=1}^{|\boldsymbol{I}|} w_n\right)\left(\frac{\sum_{n=1}^{|\boldsymbol{I}|}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}\right). \qquad (5)
\end{aligned}$$

The expected admixture proportion on k-th topic can then be obtained by

$$\begin{aligned}
\theta_k &= \mathbb{E}\left[\frac{\sum_{n=1}^{|\boldsymbol{I}|} w_{nk}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K} w_{nk}}\right] \\
&= \frac{\mathbb{E}[\sum_{n=1}^{|\boldsymbol{I}|} w_{nk}]}{\sum_{n=1}^{|\boldsymbol{I}|} w_n} \\
&= \frac{\left(\sum_{n=1}^{|\boldsymbol{I}|} w_n\right)\left(\frac{\sum_{n=1}^{|\boldsymbol{I}|}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}\right)}{\sum_{n=1}^{|\boldsymbol{I}|} w_n} \\
&= \frac{\sum_{n=1}^{|\boldsymbol{I}|}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}{\sum_{n=1}^{|\boldsymbol{I}|}\sum_{k=1}^{K}\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn}}. \qquad (6)
\end{aligned}$$

Obviously, the obtained $\{\theta_k\}_{k=1}^{K}$ satisfies $\theta_k \geq 0, \forall k$; $\sum_{k=1}^{K}\theta_k = 1$. □

3.3 Optimization

The objective function $\mathcal{L}(\Theta,\boldsymbol{\beta})$ is bi-convex, that is, convex over either Θ or $\boldsymbol{\beta}$ when the other is fixed. We omit the proof procedures here which is easy to make. Furthermore, the feasible set is a convex set. Therefore, a typical solution to solve this bi-convex problem is coordinate descent algorithm [21]. The algorithm alternatively performs the optimization over Θ and β as shown in Algorithm 1.

Optimize over Θ: this step aims to find the word codes Θ when dictionary $\boldsymbol{\beta}$ is fixed. Due to the conditional independency, we can perform this step for each document separately by solving the optimization problem:

$$\begin{aligned}
&\min_{\boldsymbol{s}} \sum_{n=1}^{|\boldsymbol{I}|} \ell(\boldsymbol{s}_{n\cdot},\boldsymbol{\beta}) + \sum_{k=1}^{K} \lambda\|\boldsymbol{s}_{\cdot k}\|_2 \\
&s.t. \quad \boldsymbol{s}_{n\cdot} \geq 0 \quad \forall n \in \boldsymbol{I}. \qquad (7)
\end{aligned}$$

The above optimization problem cannot directly be solved using traditional gradient methods, since the objective function in formula (7) is not derivable in the domain of $\boldsymbol{s}$. Since the word codes are grouped in terms of topics and each group is separable, here we adopt the block coordinate decent method for this optimization [3,

Algorithm 1 Calculate $\min_{\{s,\beta\}} L(D)$

Require: $T, K \in Z_+, \epsilon > 0, D, W$
  for $k = 1 : K$ **do**
    initialize $\beta_{k\cdot} \in \mathbb{R}^{|W|}$ randomly
  end for
  initialize $s \in \mathbb{R}^{|D|\times|W|\times K}$ randomly
  for $t = 1 : T$ **do**
    for $d \in D$ **do**
      Learn Coding of s_d with algorithm 2
    end for
    $l \leftarrow$ calculate loss function value as formula (4)
    if $l - l^{old} < \epsilon$ **then**
      break
    end if
    $l^{old} = l$
    Learn topic β with algorithm 3
  end for
  return s, β

13]. Therefore, we only focus on one group, e.g. the k-th group, and obtain the objective function as:

$$\mathcal{L}(\boldsymbol{s}_{\cdot k}) = \sum_{n=1}^{|\boldsymbol{I}|} (\boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn} - w_n \ln(\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n})) + \lambda \|\boldsymbol{s}_{\cdot k}\|_2. \tag{8}$$

Taking the subderivative on $\boldsymbol{s}_{\cdot k}$, the subgradient equations are

$$\boldsymbol{\beta}_{nk} - \frac{w_n \boldsymbol{\beta}_{nk}}{\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n}} + \lambda \boldsymbol{\zeta}_{nk} = 0; \quad \forall n \in \boldsymbol{I}, \tag{9}$$

where $\boldsymbol{\zeta}_{nk} = \frac{\boldsymbol{s}_{nk}}{\|\boldsymbol{s}_{\cdot k}\|_2}$ when $\boldsymbol{s}_{\cdot k} \neq 0$, and $\boldsymbol{\zeta}_{\cdot k}$ is a k-dimension vector satisfying $\|\boldsymbol{\zeta}_{\cdot k}\|_2 \leq 1$ otherwise. After transforming the formula (9), we get,

$$\|\boldsymbol{\zeta}_{\cdot k}\|_2 = \sqrt{\frac{1}{\lambda}\sum_{n=1}^{|\boldsymbol{I}|}\left(-\beta_{kn} + \frac{w_n \beta_{kn}}{\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n}}\right)^2}. \tag{10}$$

Therefore, we can determine whether the vector of $\boldsymbol{s}_{\cdot k}$ is zero by checking the value of $\|\boldsymbol{\zeta}_{\cdot k}\|_2$. If $\|\boldsymbol{\zeta}_{\cdot k}\|_2 \leq 1$ we set the vector $\boldsymbol{s}_{\cdot k}$ to a zero vector, otherwise we solve the optimization problem in formula (8) using coordinate descent method. Specifically, for each $\boldsymbol{s}_{nk}$, given $\boldsymbol{s}_{jk}, j \neq n$ fixed, the objective function can be written as

$$\mathcal{L}(\boldsymbol{s}_{nk}) \quad = \boldsymbol{s}_{nk}\boldsymbol{\beta}_{kn} - w_n \ln(\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n}) + \lambda\sqrt{\boldsymbol{s}_{nk}^2 + C_{nk}} \tag{11}$$

where $C_{nk} = \sum_{i\in\boldsymbol{I}\wedge i\neq n} \boldsymbol{s}_{ik}^2 \geq 0$ relative to $\boldsymbol{s}_{nk}$ is a constant. $\boldsymbol{s}_{nk}$.

The gradient of formula (11) is

$$\nabla_{\boldsymbol{s}_{nk}}\mathcal{L}(\boldsymbol{s}_{nk}) = (1 - \frac{w_n}{\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n}})\boldsymbol{\beta}_{kn} + \lambda\frac{\boldsymbol{s}_{nk}}{\sqrt{\boldsymbol{s}_{nk}^2 + C_{nk}}}, \tag{12}$$

and the second derivative of $\mathcal{L}(\boldsymbol{s}_{nk})$ is

$$\frac{\partial^2 \mathcal{L}(\boldsymbol{s}_{nk})}{\partial s_{nk}^2} = w_n \beta^2 (\boldsymbol{s}_{n\cdot}^\top \boldsymbol{\beta}_{\cdot n})^{-2} + \lambda C_{nk}(s_{nk}^2 + C_{nk})^{-\frac{3}{2}}. \tag{13}$$

Since $\frac{\partial^2 \mathcal{L}(\boldsymbol{s}_{nk})}{\partial s_{nk}^2} > 0$ for $\forall s_{nk}$, the formula (11) is strictly convex with respect to $\boldsymbol{s}_{nk}$. By setting the gradient $\nabla_{\boldsymbol{s}_{nk}}\mathcal{L}(\boldsymbol{s}_{nk}) = 0$ equal to to zero, we have that $\boldsymbol{s}_{nk}$ is the solution of the following quartic problem:

$$\begin{aligned}
&\left(\boldsymbol{\beta}_{kn}^2(\boldsymbol{\beta}_{kn} + \eta)^2 - \boldsymbol{\beta}_{kn}^2\lambda^2\right)\boldsymbol{s}_{nk}^4 \\
&-\left(2X_{nk}\boldsymbol{\beta}_{kn}(\boldsymbol{\beta}_{kn} + \eta) + 2\boldsymbol{\beta}_{kn}\lambda^2 B_{nk}\right)\boldsymbol{s}_{nk}^3 \\
&+\left(X_{nk}^2 + (\boldsymbol{\beta}_{kn} + \eta)^2\boldsymbol{\beta}_{kn}^2 C_{nk} - B_{nk}^2\lambda^2\right)\boldsymbol{s}_{nk}^2 \\
&-2X_{nk}(\boldsymbol{\beta}_{kn} + \eta)\boldsymbol{\beta}_{kn}C_{nk}\boldsymbol{s}_{nk} \\
&+X_{nk}^2 C_{nk} = 0 \\
&s.t. \quad \boldsymbol{s}_{nk} \geq 0,
\end{aligned} \tag{14}$$

where $B_{nk} = \sum_{i=1\wedge i\neq k}^{K} \boldsymbol{s}_{ni}\boldsymbol{\beta}_{in}$ and $X_{nk} = w_n\boldsymbol{\beta}_{kn} - B_{nk}(\boldsymbol{\beta}_{kn} + \eta)$. The feasible solution of $\boldsymbol{s}_{nk}$ is either in the roots of Equation 14) or the boundary point which is zero [40]. Solving this quartic equation is very efficient using modern numerical tools. The detailed encoding algorithm is presented in Algorithm 2.

Algorithm 2 Encoding Algorithm in GSTC

Require: s_d, β, D, ϵ
  for $t = 1 : T$ **do**
    for $k = 1 : K$ **do**
      if $\|\boldsymbol{\zeta}_{\cdot k}\|_2 \leq 1$ **then**
        $s_{d,\cdot k} \leftarrow \mathbf{0}$
        continue;
      end if
      for $n = 1 : |I_d|$ **do**
        calculate $s_{d,nk}^\star$ by solving Eq. (14)
        $s_{d,nk} \leftarrow \max(s_{d,nk}^\star, 0)$
      end for
    end for
    $l \leftarrow$ calculate the loss function value with formula (4)
    if $l - l^{old} < \epsilon$ **then**
      break
    end if
    $l^{old} = l$
  end for
  return s_d

Optimize over $\boldsymbol{\beta}$: After we have inferred all the latent word codes of the collection, we update the dictionary $\boldsymbol{\beta}$ by minimizing the log-Poisson loss. It is a convex optimization problem which can be efficiently solved with high-performance methods. Here we adopt the limited-memory projected quasi-Newton(L-PQN) algorithm [26] for our problem. The L-PQN algorithm is suitable for large-scale optimization problems where the evaluation of the function is substantially more expensive than the projection onto the constraint set. It fits our problem as the calculation of the loss function is time consuming, while the projecting a vector to a probabilistic simplex can be conducted in linear time [10].

The L-PQN algorithm uses the gradient of loss function for optimization which is defined as Equation (15) for all n, k.

$$\frac{\partial \mathcal{L}(\beta)}{\partial \beta_{kn}} = \sum_{d=1}^{M}\left(s_{d,nk} - \frac{w_{d,n\cdot}s_{d,nk}}{\sum_{k=1}^{K} s_{d,nk}\beta_{kn}}\right). \tag{15}$$

To further speed-up the optimization process, we do not use the coordinate descent algorithm to optimize the $\beta_{k\cdot}$ one by one. Alternatively, we transform the topic matrix β to a large vector $\vec{\beta}$. The $\vec{\beta}$ has $K \times N$ elements, and includes K normalization constraints on every N elements. In this way, we can simultaneously learn the whole topics matrix using L-PQN. The topic learning procedure is described in Algorithm 3, and for the detailed steps of L-PQN one may refer to [26].

Algorithm 3 Topic Learning Algorithm in GSTC

Require: s, β, D

transform matrix β to vector $\vec{\beta}$
$pr \leftarrow$ a function to do simplex projection
$obj \leftarrow$ a function to calculate both the loss value as formula (4)
$grd \leftarrow$ a function to calculate the gradient as formula (15)
$\vec{\beta} \leftarrow$ L-PQN($\beta, s, D, pr, obj, grd$)
transform vector $\vec{\beta}$ back to matrix β
return β

Table 1: Comparison among Different Topic Models

Property	PTMs	NPMs	GSTC
Efficient Sparse Control	No	Yes	Yes
Semantic Interpretation	Yes	No	Yes
Learning Efficiency	Good	Better	Better

3.4 Discussions

GSTC is a non-probabilistic topic model for discovering sparse latent representations of large text corpora. Here we compared GSTC with exiting PTMs and NPMs to show the benefits of GSTC on text modelling. The advantages and weakness of different topic models are summarized in the table 1, and the detailed explanations are presented as follows.

The PTMs (e.g. LDA[5]) construct the document in a probabilistic way, where a document is typically modeled as a finite admixture of latent topics, and each topic is a unigram distribution over a given vocabulary. The document-level admixture proportions can be regarded as a low dimensional representation of the document in topic simplex. When compared with unnormalized document codes (typically obtained by NPMs), such admixture proportions would be beneficial in both understanding and semantic comparison between documents as they clearly convey the relative importance of topics in documents. Besides, when using the latent representations as features for future tasks (e.g. text classification, text clustering and so on), such admixture proportions (within the interval $[0, 1]$) might be more robust and preferred than usually scale-unstrained document codes. GSTC enjoys the merits of latent semantic representations as PTMs. Although GSTC relaxed the normalization constraint in modeling, we can naturally recover the admixture proportions afterwards.

A limitation of PTMs is that it is non-trivial to achieve sparsity since the admixture proportions or topics are normalized distribution. This problem can be well addressed in the non-probabilistic GSTC. Due to the same reason of having to define a normalized likelihood function, another limitation of PTMs is that they usually have to deal with a hard-to-compute *log-sum-exp* function [40]. By relaxing the normalization constraint, the learning problem in GSTC can be effectively solved using coordinate descent method, which has closed-form solutions for updating word codes. Empirical results in Sec 4.4 show that GSTC is much more efficient than LDA in training.

The advantage of NPMs lies in that they provide an elegant framework to achieve sparsity in topic modeling by using sparse constraint like ℓ_1 norm as lasso [21]. As a non-probabilistic model, GSTC enjoys the same merit by directly control the sparsity of inferred representations at document-level with the help of group lasso. Unlike SPC [22] and standard NMF [20, 15] encode all the words in a vocabulary, GSTC only encodes the words with nonzero counts as STC, which makes GSTC more efficient and scalable to large vocabulary. Compared to STC, GSTC has less variables to be estimated since we do not need to model the document codes θ. The additional variables θ make the STC require more data in learning.

One limitation of existing NPMs, as aforementioned, is that they lose the clear semantic explanations over the latent representations. Although some NPMs like STC and NMF can directly learn document-level latent representations, the learned codes no longer lie on the same topic simplex like PTMs, but could be in quite different scales over different documents. Mapping the codes into a semantic simplex is not natural and straightforward. An ad-hoc normalized code vector no longer represents the admixture proportions in topic simplex. While in GSTC, we can derive the expected admixture proportions based on the learned word codes and dictionary.

4. EXPERIMENTS

In this section, we provide both qualitative and quantitative evaluation of GSTC. We first introduce the dataset the baseline methods. Then we conduct extensive experiments to compare the performance of GSTC with some state-of-the-art methods. The empirical results demonstrate the effectiveness and efficiency of the proposed model.

4.1 Dataset and Baseline Methods

We conducted empirical experiments based on the 20-Newsgroup dataset. The 20-Newsgroup is a benchmark text collection widely used in topic modeling, which organizes the postings to Usenet newsgroups into 20 related categories almost evenly. Here we make use of a preprocessed version downloaded from Cai's webpage[1], which includes $18,846$ documents and $26,214$ distinct words. The vocabulary is generated by stemming each term to its root and removing the stop words.

We evaluation the performance of GSTC by comparing with several state-of-art PTMs and NTMs, including:

- Latent Dirchelet Allocation (LDA). The code of LDA is provided by Blei[2], which employs variational EM in the training process. The Dirichlet parameters are estimated by the Newton-Raphson method [5].
- Sparse Topical Coding (STC). The code of STC is available online[3], which employs the coordinate descent algorithm to learn the model.
- Non-negative Matrix Factorization (NMF). NMF is solved with an alternating constrained least squares algorithm [2].

For comparison fairness, all the baseline methods are initialized randomly similar as GSTC, the training parameters of these models are set according to the original papers' suggestions , and the regularization constants are selected by cross-validation.

4.2 Sparse Latent Representations

To demonstrate the effectiveness in learning meaningful topics by our model, we first show 9 randomly picked learned topics in Table 2. For each topic, we list several top words according to their probabilities under the corresponding topic. Each column is a distinct topic, where the topic labels are produced by human judges for better understanding. Although the top words listed here are in their stemmed form, it is obvious to see that the learned topics are

[1]http://www.zjucadcg.cn/dengcai/Data/TextData.html
[2]http://www.cs.princeton.edu/ blei/lda-c/
[3]http://www.cs.cmu.edu/ junzhu/stc.htm

Table 2: Top words for 4 randomly selected learned topics

COMPUTER	LOVE	DRUG	WEAPON	BICYCLE	TERROR	DIET	MEDICATION	SPACE-FLIGHT
pc	love	drug	weapon	motorcycl	terror	diet	medic	launch
printer	promis	illeg	gun	bike	soldier	vitamin	diseas	orbit
hardware	rose	alcohol	arm	ride	civilian	fat	infect	satellit
software	lie	addict	firearm	guid	iraqi	nutrition	patient	shuttl
system	marriag	heroin	automat	rider	bomber	headach	clinic	solar
lan	marri	cocain	rifl	biker	casualti	dose	diet	mar
inkjet	grave	marijuana	pistol	road	defenseless	intak	syndrom	spacecraft

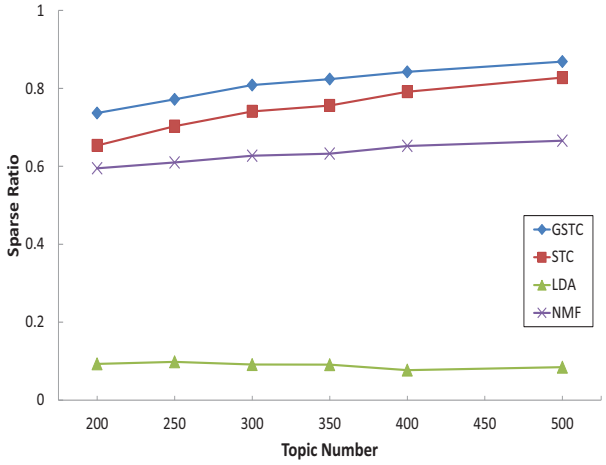

Figure 4: The sparse ratio of document-level latent representations learned by different models.

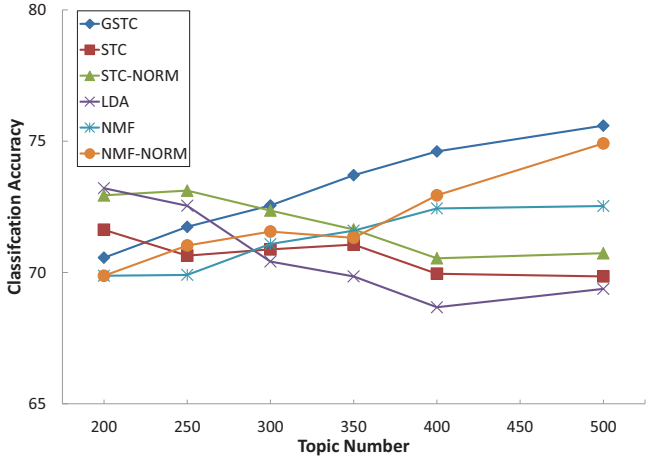

Figure 5: The classification accuracy of different models.

quite clear and semantically concentrated. For example, we can find *pc, hardware*, and *software* in the topic about computer, and *weapon, gun*, and *pistol* in the topic about weapon.

We compare the sparsity of the learned latent representations of documents from different models. We randomly selected two news groups and sampled two documents from each. The inferred latent representations of documents from different models under 500 topics are shown in Fig. 2 and Fig. 3. We can see that LDA tends to learn a broad spectrum of topics for each document, while the other three models can discover more sparse and salient patterns. Take the first document in the autos group as an example, LDA assigned 456 topics to it, while STC, NMF and GSTC discovered 33, 162, and 28 topics respectively. Note that STC seems sparser than GSTC since the document codes learned by STC varies largely and thus some small codes cannot be clearly observed due to the image scale. From Fig. 2 and Fig. 3, we can also find that the document codes learned by STC can be in quite different scales. For example, the most salient document codes in autos group are around 8 while in hardware group are around 11. It is difficult to use such codes to directly explain the importance of topics or conduct semantic comparison between documents.

We further quantitatively evaluate the average sparse ratio on latent representations of documents from different models, as shown in Fig. 4. Not surprisingly, we can see that the representations learned by LDA are very dense. Since there is no direct sparsity control over the posterior representations in LDA, the sparsity in inferred admixture proportions is mainly caused by the data scarcity. Meanwhile, the sparse ratio of NMF is lower than STC and GSTC. It indicates that directly imposing sparse constraints (e.g. lasso or group lasso) can achieve better sparsity than using the non-negative constraints alone. By comparing GSTC and STC, we can see that using group lasso can further improve the sparsity ratio (and classification performance as shown later). The possible reason is that group lasso uses structural information to align variables to sparse and salient factors, and such additional constraints over variables may lead to better sparsity.

4.3 Document Classification

To quantitatively evaluate the quality of learned representations from different models, we report the performance on document classification. Specifically, we use all the 20-newsgroup data to learn the parameters of different models. Then we use the training documents with their inferred representations as features to build multi-class SVM classifiers. Note that in LDA and our GSTC, the inferred latent representations of documents are admixture proportions; while in STC and NMF, the latent representations of documents are unnormalized code vectors. Here we also conducted post-normalization on the document codes and used the normalized codes as input features. We denote the results based on these normalized codes as STC-Norm and NMF-Norm, respectively. We adopted the LIBSVM toolbox[4] as our classifier implementation. The one-against-one approach is employed in multi-class classification by LIBSVM. The train/test split is the same as [8]. We performed cross validation to select the coefficient C of SVM as well as the regularizer parameters in GSTC and STC.

The classification results are depicted in the Fig 5. We can see that GSTC performs better than all the other three models, especial-

[4]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

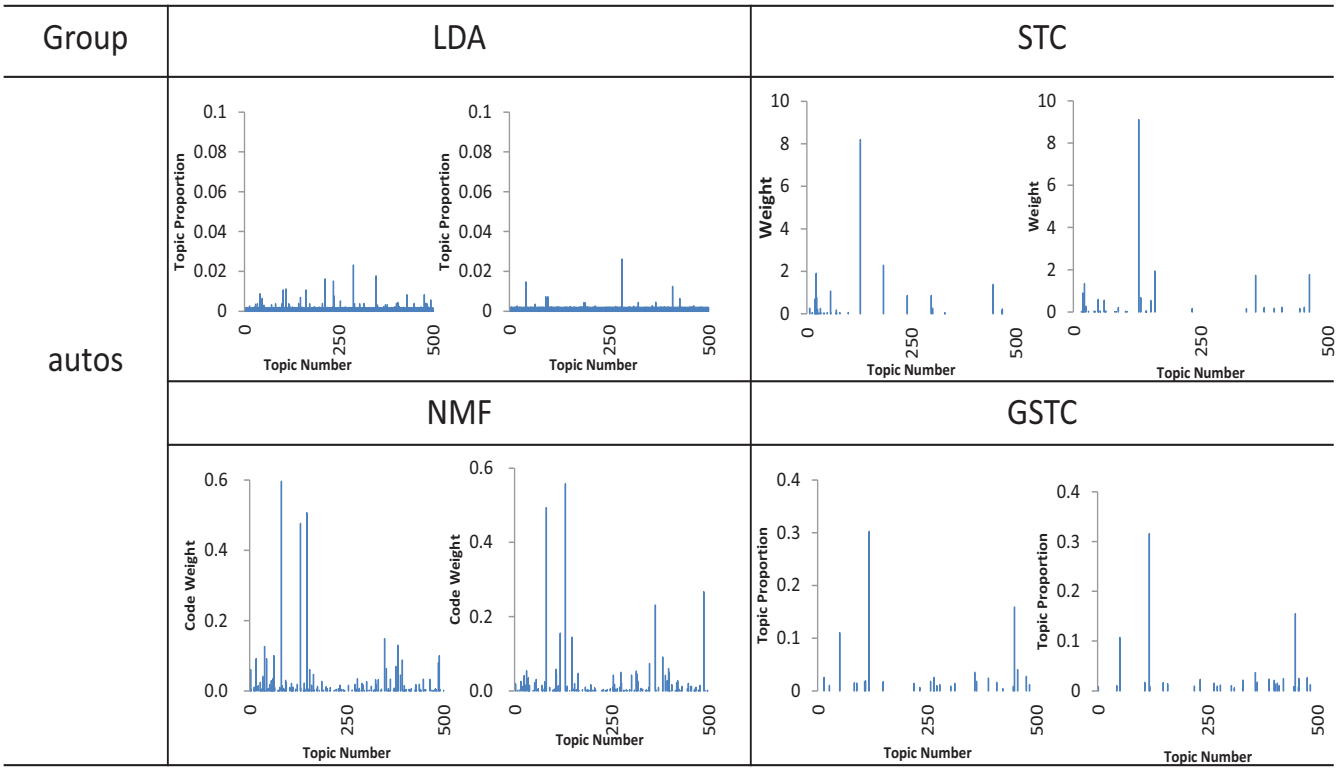

Figure 2: The latent representations of example documents from autos category discovered by different models.

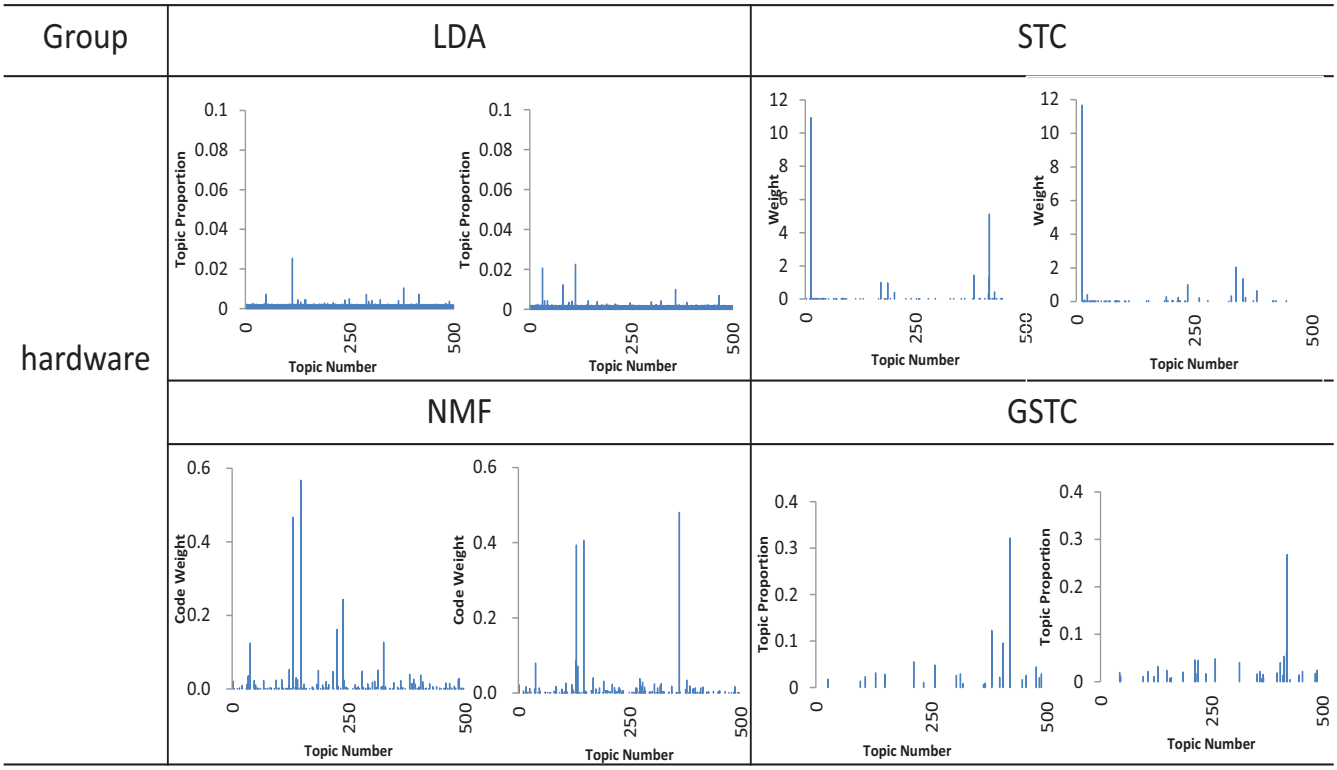

Figure 3: The latent representations of example documents from hardware category discovered by different models.

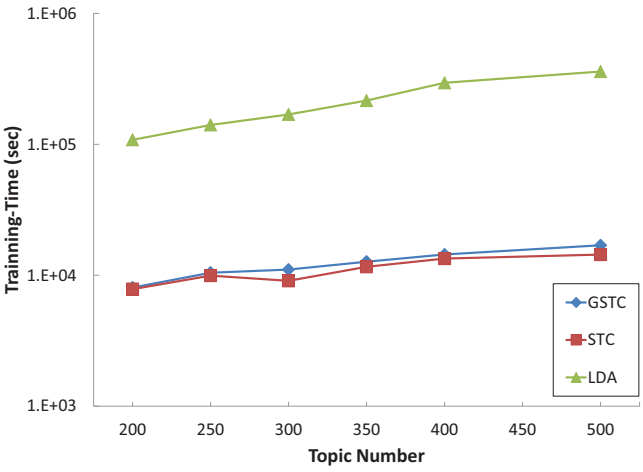

Figure 6: The training time of different models.

ly when the topic number is large. As compared with LDA, we can find that LDA performs well when the topic number is small, but the performance drops when topic number increases. It indicates that by learning compact representations for documents, especially when there are a large number of potential topics in the text corpora, we can achieve better prediction performance. As compared with NMF and STC, one possible reason for the improvement is that GSTC learns the admixture proportions which lay different documents in the topic simplex. Such latent representations may better help compare or classify documents than the unnormalized code vectors which are often in different scales. This can also be verified if we compare STC and NMF with their normalized counterparts respectively, where using normalized features can always performs better. Meanwhile, one may notice that the curve of STC in Fig 5 is downward-sloping. This phenomenon can also be observed in the original paper of STC 5 when topic number is larger than 100. One possible reason is that by setting the same weight over the sparse control and the alignment components (as suggested in [40]), the adaptability of STC may get hurt when topic number become large. Finally, by comparing GSTC with STC-Norm and NFM-Norm, we can see that with the learned admixture proportions, one can obtain better classification accuracy than using the normalized document codes which lack clear semantic explanations.

4.4 Time Efficiency

We further compare the learning efficiency of different models. Since GSTC, LDA and STC are all implemented in C++ while NMF is implemented in Matlab, here we compare the training time of the first three but omit the result of NMF for fair comparison.[5] All the experiments are conducted on the same workstation with a 2.33GHZ intel processor. Each model runs 10 times with random initialization.

Fig 6 shows the average training time of different models. Clearly, both GSTC and STC are much more efficient than LDA. The main reason is that the coordinate descent algorithm employed by both GSTC and STC is much more efficient than the probabilistic inference of LDA. Besides, as the learning process proceeds, the sparse representations obtained by GSTC and STC can further speed-up the learning process, while the LDA almost keeps the same learning speed for every iteration. By comparing GSCT and STC, we can see that both obtain comparable time efficiency in learning. It seems that in GSTC the saving on the cost of explicitly learning the document codes, compensates the cost of solving group lasso rather than lasso in STC.

[5]In fact, it is shown that the time cost of NMF is much worse than STC in the appendix of STC [40]. From our results later, therefore, we may anticipate the running time of NMF will also be longer than GSTC

5. CONCLUSIONS AND FUTURE WORK

In this paper, we present group sparse topical coding (GSTC), a novel non-probabilistic topic model for discovering sparse latent representations (i.e. admixture proportions) of documents over large text corpora. GSTC enjoys both the merits of the PTMs and NPMs. By relaxing the normalization constraint made in PTMs, GSTC can directly control the sparsity of the inferred representations and learn with efficient algorithm. Meanwhile, GSTC can naturally derive document-level admixture proportions as in PTMs with learned word codes and dictionary. Empirical results on benchmark datasets show that GSTC can identify sparse topics, and improve the document classification accuracy and time efficiency.

For the future work, it would be interesting to further consider the sparsity of dictionary, and develop parallel learning algorithm for GSTC for large-scale applications. Besides, we will also try to extend the GSTC by integrating the discriminative features of document, such as document structure, the case of words, label of documents and so on. These discriminative information may help the GSTC learn the codes and topics more accurately and thus further improve the classification performance. Finally, it would also be interesting to apply the GSTC in other fields, such as computer vision and signal processing.

6. ACKNOWLEDGMENTS

This research is funded by the National Natural Science Foundation of China under Grant No. 60933005, No. 61173008, No. 61003166 , No. 61203298, 973 Program of China under Grants No. 2012CB316303, and the National 242 Project of China under Grant No. 2011F65, 2011A001.

7. REFERENCES

[1] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.

[2] R. Albright, J. Cox, D. Duling, A. N. Langville, and C. D. Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. *Matrix*, (919):1–18, 2006.

[3] S. Bengio, F. C. N. Pereira, Y. Singer, D. Strelow, and D. Strelow. Group sparse coding. In *NIPS*, pages 82–89, 2009.

[4] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[5] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:2003, 2003.

[6] W. Buntine and A. Jakulin. Discrete component analysis. *Subspace Latent Structure and Feature Selection*, 3940:23–25, 2006.

[7] D. Cai, Q. Mei, J. Han, and C. Zhai. Modeling hidden topics on document manifold. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 911–920, New York, NY, USA, 2008. ACM.

[8] D. Cai, S. Member, X. He, J. Han, and S. Member. Srda: An efficient algorithm for large-scale discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):1–12, 2008.

[9] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[10] J. Duchi, S. S. Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the L1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 272–279, New York, NY, USA, 2008. ACM.

[11] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[12] T. Eltoft, T. Kim, and T.-W. Lee. On the multivariate Laplace distribution. *IEEE Signal Processing Letters*, 13(5):300–303, May 2006.

[13] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv10010736*, pages 1–8, 2010.

[14] J. Graca, K. Ganchev, B. Taskar, and F. Pereira. Posterior vs Parameter Sparsity in Latent Variable Models. In *Proceedings of NIPS*, 2009.

[15] M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *J. Mach. Learn. Res.*, 7:1385–1407, December 2006.

[16] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning*, page 2001, 2001.

[17] L. Jacob, G. Obozinski, J.-P. Vert, and J.-P. Vert. Group lasso with overlap and graph lasso. In *ICML*, page 55, 2009.

[18] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, pages 487–494, 2010.

[19] M. Kyung, J. Gill, M. Ghosh, and G. Casella. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–412, 2010.

[20] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct. 1999.

[21] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808, 2006.

[22] H. Lee, R. Raina, A. Teichman, and A. Y. Ng. Exponential family sparse coding with applications to self-taught learning. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1113–1119, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[23] 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, June 1996.

[24] J. Paisley, L. Carin, and D. Blei. Variational inference for stick-breaking beta process priors. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 889–896, New York, NY, USA, June 2011. ACM.

[25] S. Raman, T. J. Fuchs, P. J. Wild, E. Dahl, and V. Roth. The bayesian group-lasso for analyzing contingency tables. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 881–888, New York, NY, USA, 2009. ACM.

[26] M. Schmidt, E. V. D. Berg, M. P. Friedl, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *Proc. of Conf. on Artificial Intelligence and Statistics*, pages 456–463, 2009.

[27] M. Shashanka, B. Raj, and P. Smaragdis. Sparse overcomplete latent variable decomposition of counts data. *Advances in Neural Information Processing Systems 20*, 20(1):1–8, 2008.

[28] S. Sra, D. Kim, S. Sra, D. Kim, and B. Schölkopf. Non-monotonic poisson likelihood maximization. Technical report, 2008.

[29] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[30] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal Of The Royal Statistical Society Series B*, 67(1):91–108, 2005.

[31] C. Wang and D. Blei. Decoupling Sparsity and Smoothness in the Discrete Hierarchical Dirichlet Process. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1982–1989. 2009.

[32] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 448–456, New York, NY, USA, 2011. ACM.

[33] Q. Wang, J. Xu, H. Li, and N. Craswell. Regularized latent semantic indexing. In *SIGIR*, pages 685–694, 2011.

[34] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

[35] S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The ibp compound dirichlet process and its application to focused topic modeling. In *ICML*, pages 1151–1158, 2010.

[36] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1794–1801, 2009.

[37] X. Yang, S. Kim, and E. Xing. Heterogeneous multitask learning with joint sparsity constraints. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2151–2159. 2009.

[38] M. Yuan, M. Yuan, Y. Lin, and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[39] M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and poisson factor analysis. 2011.

[40] J. Zhu and E. P. Xing. Sparse topical coding. In *UAI*, pages 831–838, 2011.

[41] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.