Learning Sparse Topical Representations

Jun Zhu[†] Aonan Zhang[†] Eric P. Xing[‡]
[†]Dept. of CS & T, TNList Lab, State Key Lab of ITS, Tsinghua University,
Beijing 100084, China
{dcszj, zan12}@mail. tsinghua. edu. cn
[‡]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
epxing@cs. cmu. edu

1 Introduction

Learning a representation that captures the latent semantics of a large collection of data is an important problem in many scientific and engineering applications. Probabilistic topic models such as LDA (latent Dirichlet allocation)[Blet *et al*. 2003] posits that each document is an admixture of latent topics where each topic is a unigram distribution over the terms in a vocabulary. The document-specific admixture proportion vector can be regarded as a representation of the document in the latent topic space, which can be used for classification [Zhu *et al*. 2009], retrieval [Hofmann 1999] or visualizing the otherwise unstructured collection; and the inferred word-level topic assignment distributions can be useful for word sense induction [Brody and Lapata 2009] or disambiguation [Boyd-Graber *et al*. 2007].

However, such a probabilistic topic model is largely limited in two aspects. First, it lacks a mechanism to explicitly control the sparsity of the inferred representations. Sparsity of the representations in a semantic space is a desirable property in text modeling[Shashanka *et al*. 2007;Wang and Blei 2009] and human vision [Olshausen and

Field 1996]. For example, very often it makes intuitive sense to assume that each document or each word has a few salient topical meanings or senses [Shashanka *et al*. 2007;Wang and Blei 2009], rather than letting every topic or sense make a non-zero contribution; this is especially important in practice for large scale text mining endeavors such as those undertaken in Google or Yahoo, where it is not uncommon to learn hundreds if not thousands of topics for hundreds of millions 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. Second, the probabilistic nature of such topic models could make it computationally difficult to incorporate supervised side information [Wang *et al*. 2009] or a rich set of features [Zhu and Xing 2010]. This is because each component in such a probabilistic model needs to be a normalized distribution, in which the normalization factor or log-partition function could make the inference extremely hard.

To achieve sparsity in a probabilistic topic model is non-trivial. Existing attempts, such as imposing posterior regularization (e. g. , using entropic priors[Shashanka *et al*. 2007] or moment constraints [Ganchev *et al*. 2010]), introducing auxiliary variables [Wang and Blei 2009], or using a sparse exponential prior in LDA [Yang *et al*. 2010], can in principle introduce a bias toward a posterior distribution that is concentrated on a small number of components (e. g. , topics). However, due to the smoothness of the regularizer (e. g. , entropic regularizer) or uncertainty of auxiliary variables, such methods often do not yield truly sparse results in practice. Moreover, the aforementioned methods aim either at achieving sparse document-level representations [Shashanka *et al*. 2007; Yang *et al*. 2010] or sparse topic vectors [Wang and Blei 2009]. To the best of our knowledge, no systematical study exists on discovering sparse word-level representations. For the second limitation, the non-probabilistic latent variable/factor models, such as non-negative matrix factorization (NMF) [Lee and Seung 1999] and sparse coding (SPC) methods, provide inspiring ideas to relax the strict normalization condition in probabilistic models.

As we have stated, the reason for the second limitation is that probabilistic models require to define normalized distributional components. Similarly, a technical reason for the difficulty in achieving sparsity in a probabilistic topic model is also that the admixing proportions or topics take the form as a normalized vector that defines a distribution.

Therefore, it is unhelpful to directly use a sparsity inducing ℓ_1-regularizer as in Lasso [Tibshirani 1996;Meinshausen and Yu, 2009]. In contrast, the non-probabilistic sparse coding [Olshausen and Field 1996] provides an elegant framework to achieve sparsity on the usually un-normalized code vector or dictionary (i. e. , a basis set) by using the theoretically sound ℓ_1-regularizer or other composite regularizers [Kim and Xing 2010; Jenatton *et al*. 2010;Jacob *et al*. 2009;Bengio *et al*. 2009]. Although much work has been done on learning a structured dictionary [Jenatton *et al*. 2010;Bengio *et al*. 2009], existing sparse coding methods typically discover flat representations, such as the single-layer sparse codes of small image patches or word terms [Jenatton *et al*. 2010; Bengio *et al*. 2009]. In order to achieve a representation of an entire image or document from the sparse codes of its components, a post-processing such as average or max pooling [Yang *et al*. 2009] is needed. This two-step procedure can be rather sub-optimal because it lacks a channel to provide direct correlations between individual component representations [Hyvärinen and Hoyer 2001], or to leverage the possibly available high-level weak supervision (e. g. , document categories) to discover predictive representations [Zhu *et al*. 2009] or learn a supervised dictionary [Mairal *et al*. 2008].

To address the above limitations, we present sparse topical coding (STC), a novel statistical method for learning sparse hierarchical representations of input samples, such as text documents. In STC, each noisy individual input feature (e. g. , a word count) is reconstructed from a sparse linear combination of a set of bases, and the representation of an entire document is derived via an aggregation strategy (e. g. , averaging or truncated averaging) from the sparse codes of all its individual word features. By using a log-loss under the broad exponential family of distributions, STC can model both discrete and continuous data. When applied to text, we use the log-Poisson loss to model discrete word counts and learn the bases that are unigram distributions over the terms in a vocabulary, also known as topics. We present an efficient coordinate descent algorithm to solve the hierarchical sparse coding problem, and the dictionary learning is efficiently done with projected gradient descent. Our algorithm provides a systematic (both algorithmic and empirical) comparison between STC and probabilistic LDA models [Blet *et al*. 2003].

In addition, we also describe a supervised STC (MedSTC) to show how to incorporate supervising side-information when it is available into the STC to discover more predictive representations and learn a more accurate document classifier. Finally, we provide some empirical studies on text modeling and classification. Our results show that STC can learn meaningful topical bases, infer sparse topical representations of documents, and identify sparse topical senses of words which would be useful for word sense induction [Pantel and Lin 2002;Brody and Lapata 2009] or disambiguation [Boyd-Graber *et al*. 2007]. We report that both the unsupervised STC and supervised MedSTC outperform several competing methods on document classification and are significantly more efficient (an order of magnitude speed up) on training and testing.

This chapter is structured as follows. Section 2 introduces related work. Section 3 presents STC and an efficient coordinate descent algorithm. Section 4 describes a collapsed version of STC and MedSTC. Section 5 presents empirical studies, and Section 6 concludes with future directions discussed.

2 Related Work

Sparse coding is a powerful technique that can learn a generic dictionary from an unlabeled corpus. The learned dictionary can be further used to encode a data sample and find a new representation, which is useful for visualization, clustering, classification, or self-taught learning[Raina *et al*. 2007]. However, by treating the inputs as independent samples and using the flat ℓ_1-norm regularizer, the standard sparse coding has limitations because of its incapacity to learn structured dictionary and structured representations of the input samples. Much work has been done focusing on addressing the first problem to learn structured dictionary, such as [Jenatton *et al*. 2010] by using a structured sparsity regularizer (e. g. , group-wise Lasso [Jacob *et al*. 2009] or tree-guided Lasso [Kim and Xing 2010]), [Varshney *et al*. 2008] by designing a structure among dictionary elements, or [Jost *et al*. 2006] by using a clustering algorithm to construct a tree structure. However, much less work has been done on learning structured sparse representations of input samples. Sparse topical coding is a hierarchical sparse coding technique, and it has close relationships with latent Dirichlet allocation (LDA) [Blet *et al*. 2003] and non-negative matrix factorization (NMF) [Lee and Seung 1999], as detailed below.

2.1 Probabilistic LDA

STC is a hierarchical sparse coding method that shares the similar goal as the probabilistic LDA [Blet *et al*. 2003] for inferring latent representations of text documents. Before formally introducing STC, we discuss potential drawbacks of LDA on the model aspect.

First, LDA does not have an explicit definition of word code. In LDA, a document is represented as a *sequence* of words $\tilde{w} = (w_1, w_2, \cdots, w_M)$, where M denotes document length and w_m is an N-dimensional indicator vector, that is, $w_{mn} = 1$ if word n appears in the mth position of the document; otherwise 0. LDA associates each position m with a topic assignment indicator variable Z_m and assumes that the topics of all the words in a document are sampled from the same document-level topic mixing proportion, which will be denoted by $\tilde{\theta}$. By assuming a Dirichlet prior over the topic mixing proportion $\tilde{\theta}$, LDA defines a joint distribution for a document

$$p(\tilde{\theta}, \mathbf{z}, \tilde{w} \mid \alpha, \beta) = p(\tilde{\theta} \mid \alpha) \prod_{m=1}^{M} p(z_m \mid \tilde{\theta}) p(w_m \mid z_m, \beta) \tag{1}$$

where both the topic assignment model $p(z_m \mid \tilde{\theta})$ and the word generating model $p(w_m \mid z_m, \beta)$ are normalized multinomial *distributions* and α are Dirichlet parameters. For comparison, an equivalence to word code can be defined as the *empirical* word-topic assignment distribution $\tilde{p}(z(n)=k) \propto \sum_m w_{mn} q(z_{mk}=1 \mid \tilde{w})$, where $z(n)$ is the topic of word n. The distribution $\tilde{p}(z(n))$ can be regarded as a representation of word n in the topic space, and it can be inferred using sampling [Brody and Lapata 2009] or variational methods [Blet *et al*. 2003].

Second, LDA lacks an explicit sparcification procedure on the inferred representations. Although we can adjust α to make $\tilde{\theta}$ concentrate much of its mass on a small number of topics *a priori*, it only indirectly influences the sparsity of inferred posterior representations [Ganchev *et al*. 2010]. In practice, using a Dirichlet prior is ineffective in controlling the posterior sparsity of LDA. Fig. 1 shows the sparsity ratio of word code (i. e., number of zeros in the code divided by topic number K) and classification accuracy with different pre-specified Dirichlet parameter α of LDA using variational inference[①]. We

[①] In theory, variational methods don't produce zero code elements because of the exponential update rule. But in practice, it is safe to truncate very small values to be zero. Similarly, sampling methods don't have a direct control on the posterior sparisty either.

can see that a small α (i. e. , a weak Dirichlet smoothing [Blet *et al*. 2003]) can yield sparse representations because of data scarcity, but this sparsity is not good for classification. Using a large α (i. e. , a strong Dirichlet smoothing) can increase the accuracy, but it dramatically reduces the sparsity ratio. Also, there is a sharp change point around $\alpha=10^{-3}$.

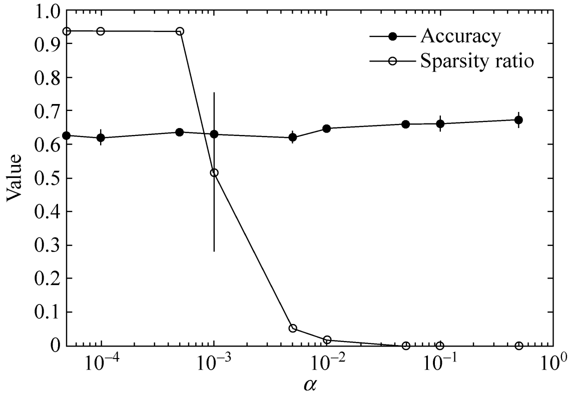

Fig. 1 Sparsity ratio of word codes and classification accuracy (please see Section 5. 1, 5. 2 for details) of 70-topic LDA with different pre-specified Dirichlet parameter α.

2. 2 Non-negative Matrix Factorization

STC is essentially a sparse hierarchical non-negative matrix factorization (NMF) [Lee and Seung 1999]. Let $\boldsymbol{X}$ denote the observed $N \times D$ word count matrix, where rows represent terms in a dictionary and columns represent documents. Then, NMF is to find non-negative matrices $\boldsymbol{U} \in \boldsymbol{R}^{N \times K}$ and $\boldsymbol{V} \in \boldsymbol{R}^{N \times D}$ such that $\boldsymbol{X} \approx \boldsymbol{UV}$, where K is the rank which is usually much smaller than N. Each column of the matrix U represents a basis and each row of V is the non-negative coefficient vector for a particular document. In [Lee and Seung 1999], a similar log-Poisson loss is used to estimate the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$. However, NMF uses one document-specific coefficient vector V_d to reconstruct all the observed word counts w_d in the same document. This assumption is often too limiting to effectively model a large collection of documents. In contrast, as we will see later, STC allows different words in one document to exhibit different sparsity patterns via using different word codes. This difference is analogous to the difference between latent Dirichlet allocation (LDA) and mixture of unigrams [Blet *et al*. 2003].

Empirically, the sparsity ratio of the "word codes" for NMF (i.e., the document-specific coefficient vector, which is the same for all the words in a document) is much smaller (around 0.005 for different numbers of topics ranging from 10 to 110) than that of STC. Therefore, NMF is limiting in using one document-specific coefficient vector to reconstruct all the word counts in that document and cannot identify the sparse topical meanings of each individual word. Although using a sparsity-inducing constraint [Hoyer 2004] can improve the sparseness of the coefficient vector in NMF, it still cannot identify the sparse topical meanings of each individual words because of the intrinsic limitation. Moreover, as shown in Fig. 2, NMF performs worse than STC on classification accuracy when the topic number is large (e.g., larger than 60) and the standard multiplicative algorithm [Lee and Seung 1999] is more expensive than our coordinate descent algorithm for training STC.

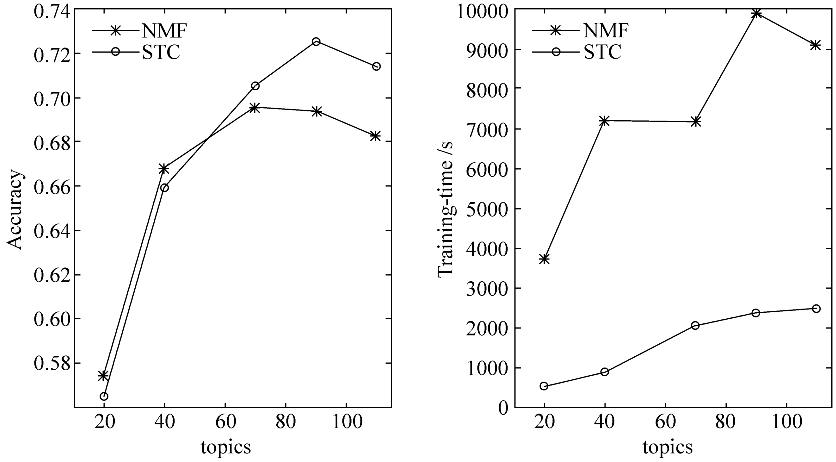

Fig. 2 (L) classification accuracy and (R) training time of NMF and STC when using different number of topics on the 20 Newsgroups data.

3 Sparse Topical Coding

Now, we present more details about sparse topical coding for modeling text documents. But the method is applicable to other types of data, such as continuous image features. Let N be the number of terms in a given vocabulary $V=\{1,2,\cdots,N\}$. Using a bag-of-words model, we represent a document as a vector $w=(w_1, w_2, \cdots, w_{|I|})^{\mathrm{T}}$, where I denotes the index set of words that appear and each w_n $(n\in I)$ represents the number of appearances of word n in this document. STC projects the input w into a semantic latent space spanned by a set of automatically learned bases (a basis set is also called a dictionary) and achieve a high-level representation of the entire document jointly. Both document-level and word-level latent representations can be used for many tasks [Blet *et al*. 2003; Boyd-Graber *et al*. 2007; Brody and Lapata 2009]. For the ease of understanding, we start with a probabilistic generating procedure.

3.1 A Probabilistic Generative Process

Let β denote a dictionary with K bases, of which each row β_k is an N dimensional basis. We assume that β_k is a unigram distribution over the N terms in V, that is, $\beta_k\in\mathcal{P}$, where $\mathcal{P}$ is an $(N-1)$-simplex. A distributional basis is also known as a topic. We will use $\beta_n\in\mathbb{R}^K$ to denote the nth column of β. Graphically, sparse topical coding (STC) is a hierarchical latent variable model, as shown in Fig. 3, where $\theta\in\mathbb{R}^K$ is the latent representation of an entire document while each $s_n\in\mathbb{R}^K$ is a latent representation of the individual word n. We call $s_n=(s_{n1}, s_{n2}, \cdots, s_{nK})^{\mathrm{T}}$ as word code and $\theta=(\theta_{n1}, \theta_{n2}, \cdots, \theta_K)^{\mathrm{T}}$ as document code.

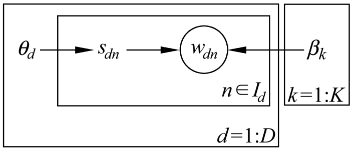

Fig. 3 A two layer sparse topical coding model

We make the assumption that for each document the word codes s_n are conditionally independent given its document code θ and the observed word counts are independent given their latent representations $\boldsymbol{s}$. We first sample a dictionary β from a uniform distribution[①]. Then, the generative process of document d is described as:

1) sample the document code θ_d from a prior distribution $p(\theta)$.

[①] Using sophisticated priors (e.g., Dirichlet prior) is our future study.

2) for each observed word $n \in I_d$

a) sample the word code $\boldsymbol{s}_{dn}$ from a conditional distribution $p(\boldsymbol{s}_n | \theta)$

b) sample the observed word count w_{dn} from a distribution with the mean being $\boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n}$.

The key idea is that we treat $\boldsymbol{s}_{dn}$ as a coefficient vector and use the linear combination $\boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n}$ to reconstruct the observed word count w_{dn}, under some loss measure; and the document code θ_d is obtained via an aggregation (e.g., average or truncated average as to be presented) of the individual codes of all its terms. In order to fully specify an STC model, we need to choose three distributions. The choices of $p(\theta)$ and $p(s_n | \theta)$ reflect our bias on the latent representations, and how θ and $\boldsymbol{s}$ are connected. We will discuss them in the next section, along with the algorithm development. Now, we define the last step of generating observed features, i.e., word counts in text documents. Here, we adopt the class of exponential family distributions to make STC flexible to model different types of data. Formally, we use the linear combination $\boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n}$ as the mean parameter of an exponential family distribution that generates the observations (e.g., w_{dn} for text documents). In other words, we find an exponential family distribution $p(w_{dn} | \boldsymbol{s}_{dn}, \beta)$ that satisfies

$$\mathbb{E}_{p(w_{dn} | s_{dn}, \beta)} [T(w_{dn})] = \boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n} \tag{2}$$

where $T(w_{dn)}$ denotes the sufficient statistics of the observations. In [Lee et al. 2009], the similar linear combination of bases is used as the natural parameter of an exponential family distribution. We choose to use it as mean parameter because of two reasons. First, using the linear combination as mean parameter makes it natural to constrain the feasible domains (e.g., nonnegative domain for modeling word counts) of the word codes in order to have a good interpretation, while it is reluctant to do so when using the linear combination as natural parameter and thus would lose good interpretation. As shown in [Lee and Seung 1999], imposing non-negativity constraints could result in significantly sparser and more interpretable patterns. Second, in many cases, such as Poisson, Bernoulli and Gaussian, the distribution is commonly expressed with mean parameters. [Buntine and Jakulin 2006] uses a similar method as ours in defining exponential family distributions.

3.2 STC for MAP Estimation

Now, we formally define STC as finding the MAP estimate of the above probabilistic model, under a bias towards finding sparse representations. Specifically, the generating procedure defines a joint distribution for document d, which is factorized as:

$$p(\theta_d, \boldsymbol{S}_d, w_d \mid \beta) = p(\theta_d) \prod_{n \in I_d} p(\boldsymbol{s}_{dn} \mid \theta_d) p(w_{dn} \mid \boldsymbol{s}_{dn}, \beta)$$

where $\boldsymbol{S}_d = \{\boldsymbol{s}_{dn} : n \in I_d\}$ denotes the set of word codes for document d. For discrete word counts, we use the Poisson distribution to generate the observations, i. e. , $p(w_{dn} \mid \boldsymbol{s}_{dn}, \beta) = Poisson\ (w_{dn}; \boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n})$, where $Poisson\ (x; \nu) = \frac{\nu^x e^{-\nu}}{x!}$. In order to achieve sparse document codes and word codes, we choose the Laplace prior $p(\theta_d) \propto \exp\ (-\lambda \| \theta_d \|_1)$ and define $p(\boldsymbol{s}_{dx} | \theta_d)$ as a product of two component distribution

$$p(\boldsymbol{s}_{dx} \mid \theta_d) \propto p(\boldsymbol{s}_{dn} \theta_{d,\gamma}) p(\boldsymbol{s}_{dn} \mid \rho) \propto \exp\left(-\frac{\gamma}{2} \| \boldsymbol{s}_{dn} - \theta_d \|_2^2 - \rho \| \boldsymbol{s}_{dn} \|_1\right) \qquad (3)$$

where we have defined $p(\boldsymbol{s}_{dn} | \theta_{d,\gamma})$ as an isotropic Gaussian distribution with mean θ_d and precision parameter γ and $p(\boldsymbol{s}_{dn} | \rho)$ is a Laplace distribution with parameter ρ. This composite distribution is super-Gaussian [Hyvärinen 1999] and the ℓ_1-norm term will bias towards finding sparse word codes. The parameters (λ, γ, ρ) are non-negative constants and they can be selected via cross-validation. For $p(\theta)$, another natural choice is a normal prior. We will discuss it along the algorithm development.

Let $\Theta = \{\theta_d\}$ and $\boldsymbol{S} = \{\boldsymbol{S}_d\}$ denote all the document codes and word codes for the corpus, respectively. Sparse topical coding solves the constrained optimization problem:

$$\begin{aligned} &\min_{\Theta, \boldsymbol{S}, \beta} \ \ell(\boldsymbol{S}, \beta; W) + \lambda \Omega(\Theta) + \frac{\gamma}{2} \sum_{d, n \in I_d} \| \boldsymbol{s}_{dn} - \theta_d \|_2^2 + \rho \Psi(\boldsymbol{S}) \\ &\text{s. t.} \quad \Theta \geqslant 0;\ \boldsymbol{S} \geqslant 0; \quad \beta_k \in \mathcal{P}, \forall k \end{aligned} \qquad (4)$$

where ℓ is a loss function and the regularization terms are $\Omega(\Theta) = \sum_d \| \theta_d \|_1$ and $\Psi(\boldsymbol{S}) = \sum_{d, n \in I_d} \| \boldsymbol{s}_{dn} \|_1$. Due to the conditional independence we have made in STC, we define the loss function as a summation over individual terms, that is,

$$\ell(\boldsymbol{S}, \beta; w) = \sum_{d, n \in I_d} \ell(w_{dn}, \boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n})$$

Then, the objective function is the negative logarithm of the posterior $p(\Theta, \boldsymbol{S}, \beta | W)$ with a constant omitted. For text documents, we define the log-Poisson loss for each individual word

$$\ell(w_{dn}, \boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n}) = -\log Poisson\ (w_{dn}; \boldsymbol{s}_{dn}^{\mathrm{T}} \beta_{.n}) \qquad (5)$$

Since word counts are non-negative, a negative θ or $\boldsymbol{s}$ will lose interpretability.