Sparse Latent Semantic Analysis

Xi Chen*   Yanjun Qi [†]   Bing Bai[†]   Qihang Lin [‡]   Jaime G. Carbonell[§]

Abstract

Latent semantic analysis (LSA), as one of the most popular unsupervised dimension reduction tools, has a wide range of applications in text mining and information retrieval. The key idea of LSA is to learn a projection matrix that maps the high dimensional vector space representations of documents to a lower dimensional latent space, i.e. so called latent *topic* space. In this paper, we propose a new model called *Sparse LSA*, which produces a sparse projection matrix via the ℓ_1 regularization. Compared to the traditional LSA, Sparse LSA selects only a small number of relevant words for each topic and hence provides a compact representation of topic-word relationships. Moreover, Sparse LSA is computationally very efficient with much less memory usage for storing the projection matrix. Furthermore, we propose two important extensions of Sparse LSA: group structured Sparse LSA and non-negative Sparse LSA. We conduct experiments on several benchmark datasets and compare Sparse LSA and its extensions with several widely used methods, e.g. LSA, Sparse Coding and LDA. Empirical results suggest that Sparse LSA achieves similar performance gains to LSA, but is more efficient in projection computation, storage, and also well explain the topic-word relationships.

1   Introduction

Latent Semantic Analysis (LSA) [5], as one of the most successful tools for learning the concepts or latent topics from text, has widely been used for the dimension reduction purpose in information retrieval. More precisely, given a document-term matrix $\mathbf{X} \in \mathbb{R}^{N \times M}$, where N is the number of documents and M is the number of words, and assuming that the number of latent topics (the dimensionality of the latent space) is set as D ($D \leq \min\{N, M\}$), LSA applies singular value decomposition (SVD) to construct a low rank (with rank-D) approximation of $\mathbf{X}$: $\mathbf{X} \approx \mathbf{U}\mathbf{S}\mathbf{V}^T$, where the column orthogonal matrices $\mathbf{U} \in \mathbb{R}^{N \times D}$ ($\mathbf{U}^T\mathbf{U} = \mathbf{I}$) and $\mathbf{V} \in \mathbb{R}^{M \times D}$ ($\mathbf{V}^T\mathbf{V} = \mathbf{I}$) represent document and word embeddings into the latent space. $\mathbf{S}$ is a diagonal matrix with the D largest singular values of $\mathbf{X}$ on the diagonal [1]. Subsequently, the so-called *projection matrix* defined as $\mathbf{A} = \mathbf{S}^{-1}\mathbf{V}^T$ provides a transformation mapping of documents from the word space to the latent topic space, which is less noisy and considers word synonymy (i.e. different words describing the same idea). However, in LSA, each latent topic is represented by all word features which sometimes makes it difficult to precisely characterize the topic-word relationships.

In this paper, we introduce a scalable latent topic model that we call "Sparse Latent Semantic Analysis" (Sparse LSA). Different from the traditional LSA based on SVD, we formulate a variant of LSA as an optimization problem which minimizes the approximation error under the orthogonality constraint of $\mathbf{U}$. Based on this formulation, we add the *sparsity* constraint of the projection matrix $\mathbf{A}$ via the ℓ_1 regularization as in the lasso model [23]. By enforcing the sparsity on $\mathbf{A}$, the model has the ability to automatically select the most relevant words for each latent topic. There are several important features of Sparse LSA model:

1. It is intuitive that only a part of the vocabulary can be relevant to a certain topic. By enforcing sparsity of $\mathbf{A}$ such that each row (representing a latent topic) only has a small number nonzero entries (representing the most relevant words), Sparse LSA can provide us a compact representation for topic-word relationship that is easier to interpret.

2. With the adjustment of sparsity level in projection matrix, we could control the granularity ("level-of-details") of the topics we are trying to discover, e.g. more generic topics have more nonzero entries in rows of $\mathbf{A}$ than specific topics.

3. Due to the sparsity of $\mathbf{A}$, Sparse LSA provides an efficient strategy both in the time cost of the projection operation and in the storage cost of the projection matrix when the dimensionality of latent space D is large.

---

*Machine Learning Department, Carnegie Mellon University
[†]NEC Lab America
[‡]Tepper School of Business, Carnegie Mellon University
[§]Language Technology Institute, Carnegie Mellon University

---

[1]Since it is easier to explain our Sparse LSA model in terms of document-term matrix, for the purpose of consistency, we introduce SVD based on the document-term matrix which is different from standard notations using the term-document matrix.

4. Sparse LSA could project a document q into a *sparse* vector representation $\widehat{q}$ where each entry of $\widehat{q}$ corresponds to a latent topic. In other words, we could know the topics that q belongs to directly form the position of nonzero entries of $\widehat{q}$. Moreover, sparse representation of projected documents will save a lot of computational cost for the subsequent retrieval tasks, e.g. ranking (considering computing cosine similarity), text categorization, etc.

Furthermore, we propose two important extensions based on Sparse LSA:

1. Group Structured Sparse LSA: we add group structured sparsity-inducing penalty as in [24] to select the most relevant groups of features relevant to the latent topic.

2. Non-negative Sparse LSA: we further enforce the non-negativity constraint on the projection matrix $\mathbf{A}$. It could provide us a pseudo probability distribution of each word given the topic, similar as in Latent Dirichlet Allocation (LDA) [3].

We conduct experiments on four benchmark data sets, with two on text categorization, one on breast cancer gene function identification, and the last one on topic-word relationship identification from NIPS proceeding papers. We compare Sparse LSA and its variants with several popular methods, e.g. LSA [5], Sparse Coding [16] and LDA [3]. Empirical results show clear advantages of our methods in terms of computational cost, storage and the ability to generate sensible topics and to select relevant words (or genes) for the latent topics.

The rest of this paper is as follows. In Section 2, we present the basic Sparse LSA model. In Section 3, we extend Sparse LSA to group structured Sparse LSA and non-negative Sparse LSA. Related work is discussed in Section 4 and the empirical evaluation of the models is in Section 5. We conclude the paper in Section 6.

2 Sparse LSA

2.1 Optimization Formulation of LSA
We consider N documents, where each document lies in an M-dimensional feature space $\mathcal{X}$, e.g. tf-idf [1] weights of the vocabulary with the normalization to unit length. We denote N documents by a matrix $\mathbf{X} = [X_1, \ldots, X_M] \in \mathbb{R}^{N \times M}$, where $X_j \in \mathbb{R}^N$ is the j-th feature vector for all the documents. For the dimension reduction purpose, we aim to derive a mapping that projects input feature space into a D-dimensional latent space where D is smaller than M. In the information retrieval content, each latent dimension is also called an hidden "topic".

Motivated by the latent factor analysis [9], we assume that we have D uncorrelated latent variables $U_1, \ldots, U_D$, where each $U_d \in \mathbb{R}^N$ has the unit length, i.e. $\|U_d\|_2 = 1$. Here $\|\cdot\|_2$ denotes the vector ℓ_2-norm. For the notation simplicity, we put latent variables $U_1, \ldots, U_D$ into a matrix: $\mathbf{U} = [U_1, \ldots, U_D] \in \mathbb{R}^{N \times D}$. Since latent variables are uncorrelated with the unit length, we have $\mathbf{U}^T \mathbf{U} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. We also assume that each feature vector X_j can be represented as a linear expansion in latent variables $U_1, \ldots, U_D$:

$$(2.1) \qquad X_j = \sum_{d=1}^{D} a_{dj} U_d + \epsilon_j,$$

or simply $\mathbf{X} = \mathbf{U}\mathbf{A} + \boldsymbol{\epsilon}$, where $\mathbf{A} = [a_{dj}] \in \mathbb{R}^{D \times M}$ gives the mapping from the latent space to the input feature space and ϵ is the zero mean noise. Our goal is to compute the so-called projection matrix $\mathbf{A}$.

We can achieve this by solving the following optimization problem which minimizes the rank-D approximation error subject to the orthogonality constraint of $\mathbf{U}$:

$$(2.2) \qquad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2$$
$$\text{subject to:} \quad \mathbf{U}^T \mathbf{U} = \mathbf{I},$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm. The constraint $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ is according to the uncorrelated property among latent variables.

At the optimum of Eq. (2.2), $\mathbf{U}\mathbf{A}$ leads to the best rank-D approximation of the data $\mathbf{X}$. In general, larger the D is, the better the reconstruction performance. However, larger D requires more computational cost and large amount memory for storing $\mathbf{A}$. This is the issue that we will address in the next section.

After obtaining $\mathbf{A}$, given a new document $q \in \mathbb{R}^M$, its representation in the lower dimensional latent space can be computed as:

$$(2.3) \qquad \widehat{q} = \mathbf{A}q.$$

2.2 Sparse LSA
As discussed in the introduction, one notable advantage of sparse LSA is due to its good interpretability in topic-word relationship. Sparse LSA automatically selects the most relevant words for each latent topic and hence provides us a clear and compact representation of the topic-word relationship. Moreover, for a new document q, if the words in q has no intersection with the relevant words of d-th topic (nonzero entries in $\mathbf{A}^d$, the d-th row of $\mathbf{A}$), the d-th element of $\widehat{q}$, $\mathbf{A}^d q$, will become zero. In other words, the *sparse* latent representation of $\widehat{q}$ clearly indicates the topics that q belongs to.

Another benefit of learning sparse $\mathbf{A}$ is to save computational cost and storage requirements when D is large. In traditional LSA, the topics with larger singular values will cover a broader range of concepts than the ones with smaller singular values. For example, the first few topics with largest singular values are often too general to have specific meanings. As singular values decrease, the topics become more and more specific. Therefore, we might want to enlarge the number of latent topics D to have a reasonable coverage of the topics. However, given a large corpus with millions of documents, a larger D will greatly increase the computational cost of projection operations in traditional LSA. In contrary, for Sparse LSA, projecting documents via a highly sparse projection matrix will be computationally much more efficient; and it will take much less memory for storing $\mathbf{A}$ when D is large.

The illustration of Sparse LSA from a matrix factorization perspective is presented in Figure 1(a). An example of topic-word relationship is shown in Figure 1(b). Note that a latent topic ("law" in this case) is only related to a limited number of words.

In order to obtain a sparse $\mathbf{A}$, inspired by the lasso model in [23], we add an entry-wise ℓ_1-norm of $\mathbf{A}$ as the regularization term to the loss function and formulate the Sparse LSA model as:

$$(2.4) \qquad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_1$$
$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I},$$

where $\|\mathbf{A}\|_1 = \sum_{d=1}^{D}\sum_{j=1}^{M}|a_{dj}|$ is the entry-wise ℓ_1-norm of $\mathbf{A}$ and λ is the positive regularization parameter which controls the density (the number of nonzero entries) of $\mathbf{A}$. In general, a larger λ leads to a sparser $\mathbf{A}$. On the other hand, a too sparse $\mathbf{A}$ will miss some useful topic-word relationships which harms the reconstruction performance. Therefore, in practice, we need to try to select larger λ to obtain a more sparse $\mathbf{A}$ while still achieving good reconstruction performance. We will show the effectiveness of λ in more details in Section 5.

2.3 Optimization Algorithm In this section, we propose an efficient optimization algorithm to solve Eq. (2.4). Although the optimization problem is non-convex, fixing one variable (either $\mathbf{U}$ or $\mathbf{A}$), the objective function with respect to the other is convex. Therefore, a natural approach to solve Eq. (2.4) is by the alternating approach:

1. When $\mathbf{U}$ is fixed, let A_j denote the j-th column of $\mathbf{A}$; the optimization problem with respect to $\mathbf{A}$:

$$\min_{\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\| + \lambda\|\mathbf{A}\|_1,$$
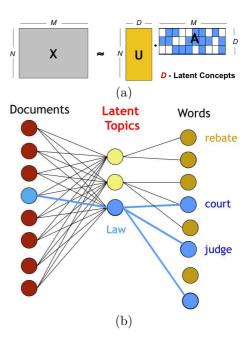

(a)

(b)

Figure 1: Illustration of Sparse LSA (a) View of Matrix Factorization, white cells in $\mathbf{A}$ indicates the zero entries (b) View of document-topic-term relationship.

can be decomposed in to M independent ones:
$$(2.5)$$
$$\min_{A_j} \frac{1}{2}\|X_j - \mathbf{U}A_j\|_2^2 + \lambda\|A_j\|_1; \quad j = 1,\dots,M.$$

Each subproblem is a standard lasso problem where X_j can be viewed as the response and $\mathbf{U}$ as the design matrix. To solve Eq. (2.5), we can directly apply the state-of-the-art lasso solver in [7] which is essentially a coordinate descent approach.

2. When $\mathbf{A}$ is fixed, the optimization problem is equivalent to:

$$(2.6) \qquad \min_{\mathbf{U}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2$$
$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

The objective function in Eq. (2.6) can be further written as:

$$\frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2$$
$$= \frac{1}{2}\text{tr}((\mathbf{X} - \mathbf{U}\mathbf{A})^T(\mathbf{X} - \mathbf{U}\mathbf{A}))$$
$$= -\text{tr}(\mathbf{A}^T\mathbf{U}^T\mathbf{X}) + \frac{1}{2}\text{tr}(\mathbf{X}^T\mathbf{X}) + \frac{1}{2}\text{tr}(\mathbf{A}^T\mathbf{U}^T\mathbf{U}\mathbf{A})$$
$$= -\text{tr}(\mathbf{A}^T\mathbf{U}^T\mathbf{X}) + \frac{1}{2}\text{tr}(\mathbf{X}^T\mathbf{X}) + \frac{1}{2}\text{tr}(\mathbf{A}^T\mathbf{A}),$$

where the last equality is according to the constraint that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. By the fact that

$\mathrm{tr}(\mathbf{A}^T\mathbf{U}^T\mathbf{X}) \equiv \mathrm{tr}(\mathbf{U}^T\mathbf{X}\mathbf{A}^T)$, the optimization problem in Eq. (2.6) is equivalent to

$$(2.7) \qquad \max_{\mathbf{U}} \quad \mathrm{tr}(\mathbf{U}^T\mathbf{X}\mathbf{A}^T)$$
$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

Let $\mathbf{V} = \mathbf{X}\mathbf{A}^T$. In fact, $\mathbf{V}$ is the latent topic representations of the documents $\mathbf{X}$. Assuming that $\mathbf{V}$ is full column rank, i.e. with rank$(\mathbf{V}) = D$, Eq. (2.7) has the closed form solution as shown in the next theorem:

THEOREM 2.1. *Suppose the singular value decomposition (SVD) of $\mathbf{V}$ is $\mathbf{V} = \mathbf{P}\Delta\mathbf{Q}$, the optimal solution to Eq. (2.7) is $\mathbf{U} = \mathbf{P}\mathbf{Q}$.*

The proof of the theorem is presented in appendix. Since N is much larger than D, in most cases, $\mathbf{V}$ is full column rank. If it is not, we may approximate $\mathbf{V}$ by a full column rank matrix $\widetilde{\mathbf{V}} = \mathbf{P}\widetilde{\Delta}\mathbf{Q}$. Here $\widetilde{\Delta}_{dd} = \Delta_{dd}$ if $\Delta_{dd} \neq 0$; otherwise $\widetilde{\Delta}_{dd} = \delta$, where δ is a very small positive number. In the later context, for the simplicity purpose, we assume that $\mathbf{V}$ is always full column rank.

It is worthy to note that since D is usually much smaller than the vocabulary size M, the computational cost of SVD of $\mathbf{V} \in \mathbb{R}^{N \times D}$ is much cheaper than SVD of $\mathbf{X} \in \mathbb{R}^{N \times M}$ in LSA.

As for the starting point, any $\mathbf{A}^0$ or $\mathbf{U}^0$ stratifying $(\mathbf{U}^0)^T\mathbf{U}^0 = \mathbf{I}$ can be adopted. We suggest a very simple initialization strategy for $\mathbf{U}^0$ as following:

$$(2.8) \qquad \mathbf{U}^0 = \begin{pmatrix} \mathbf{I}_D \\ \mathbf{0} \end{pmatrix},$$

where $\mathbf{I}_D$ the D by D identity matrix. It is easy to verify that $(\mathbf{U}^0)^T\mathbf{U}^0 = \mathbf{I}$.

The optimization procedure can be summarized in Algorithm 1.

As for the stopping criteria, let $\|\cdot\|_\infty$ denote the matrix entry-wise ℓ_∞-norm, for the two consecutive iterations t and $t+1$, we compute the maximum change for all entries in $\mathbf{U}$ and $\mathbf{A}$: $\|\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}\|_\infty$ and $\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\|_\infty$; and stop the optimization procedure when both quantities are less than the prefixed constant τ. In our experiments, we set $\tau = 0.01$.

3 Extension of Sparse LSA

In this section, we propose two important extensions of Sparse LSA model.

3.1 Group Structured Sparse LSA
Although entry-wise ℓ_1-norm regularization leads to the sparse

---

Algorithm 1 Optimization Algorithm for Sparse LSA

Input: $\mathbf{X}$, the dimensionality of the latent space D, regularization parameter λ

Initialization: $\mathbf{U}^0 = \begin{pmatrix} \mathbf{I}_D \\ \mathbf{0} \end{pmatrix}$,

Iterate until convergence of U and A:

1. Compute $\mathbf{A}$ by solving M lasso problems as in Eq. (2.5)

2. Project $\mathbf{X}$ onto the latent space: $\mathbf{V} = \mathbf{X}\mathbf{A}^T$.

3. Compute the SVD of $\mathbf{V}$: $\mathbf{V} = \mathbf{P}\Delta\mathbf{Q}$ and let $\mathbf{U} = \mathbf{P}\mathbf{Q}$.

Output: Sparse projection matrix $\mathbf{A}$.

---

projection matrix $\mathbf{A}$, it does not take advantage of any prior knowledge on the structure of the input features (e.g. words). When the features are naturally clustered into groups, it is more meaningful to enforce the sparsity pattern at a group level instead of each individual feature; so that we can learn which groups of features are relevant to a latent topic. It has many potential applications in analyzing biological data. For example, in the latent gene function identification, it is more meaningful to determine which pathways (groups of genes with similar function or near locations) are relevant to a latent gene function (topic).

Inspired by the group lasso [24], we can encode the group structure via a ℓ_1/ℓ_2 mixed norm regularization of $\mathbf{A}$ in Sparse LSA. Formally, we assume that the set of groups of input features $\mathcal{G} = \{g_1, \ldots, g_{|\mathcal{G}|}\}$ is defined as a subset of the power set of $\{1, \ldots, M\}$, and is available as prior knowledge. For the purpose of simplicity, we assume that groups are non-overlapped. The group structured Sparse LSA can be formulated as:

$$(3.9) \quad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda\sum_{d=1}^{D}\sum_{g \in \mathcal{G}} w_g\|\mathbf{A}_{dg}\|_2$$

$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I},$$

where $\mathbf{A}_{dg} \in \mathbb{R}^{|g|}$ is the subvector of $\mathbf{A}$ for the latent dimension d and the input features in group g; w_g is the predefined regularization weight each group g, λ is the global regularization parameter; and $\|\cdot\|_2$ is the vector ℓ_2-norm which enforces all the features in group g for the d-th latent topic, $\mathbf{A}_{dg}$, to achieve zeros simultaneously. A simple strategy for setting w_g is $w_g = \sqrt{|g|}$ as in [24] so that the amount of penalization is adjusted by the size of each group.

To solve Eq. (3.9), we can adopt the alternating approach as described in Section 2.3. When $\mathbf{A}$ is

fixed, optimization with respect to $\mathbf{U}$ is the same as Eq. (2.6) which can be solved by SVD as described in Theorem 2.1. When $\mathbf{U}$ is fixed, the optimization problem becomes:
(3.10)

$$\min_{\mathbf{A}} f(\mathbf{A}) \equiv \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda \sum_{d=1}^{D}\sum_{g \in \mathcal{G}} w_g \|\mathbf{A}_{dg}\|_2.$$

To solve Eq. (3.10), we propose an efficient block coordinate descent algorithm: at each iteration, the objective function is minimized with respect to $\mathbf{A}_{dg}$ while the other entries in $\mathbf{A}$ are held fixed.

More precisely, assume that now fix a particular latent dimension d and a group g; we optimize $f(\mathbf{A})$ with respect to $\mathbf{A}_{dg}$. Denote the i-th row of $\mathbf{U}$ as U^i and the first part of $f(\mathbf{A})$ as $g(\mathbf{A}) \equiv \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2$, the gradient of $g(\mathbf{A})$ over $\mathbf{A}_{dg}$ is a $|g|$ dimensional vector where the $j \in g$-th element takes the following form:

$$\left(\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}_{dg}}\right)_{j \in g} = \sum_{i=1}^{N} u_{id}((U^i)^T A_j - x_{ij}).$$

To further write $\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}_{dg}}$ in the vector form, let $C_d = \sum_{i=1}^{N} u_{id}^2$ and $\mathbf{B}_{dg}$ be the vector of length $|g|$ such that:

$$(3.11) \qquad (\mathbf{B}_{dg})_{j \in g} = \sum_{i=1}^{N} u_{id}(x_{ij} - \sum_{k \neq d} u_{ik} a_{kj}).$$

The vector form of $\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}_{dg}}$ can be written as:

$$\frac{\partial g(\mathbf{A})}{\partial \mathbf{A}_{dg}} = C_d \mathbf{A}_{dg} - \mathbf{B}_{dg}.$$

Now we show that minimization of $f(\mathbf{A})$ with respect to $\mathbf{A}_{dg}$ has a simple closed-form solution. Take the subgradient of $f(\mathbf{A})$ over $\mathbf{A}_{dg}$:

$$(3.12) \frac{\partial f(\mathbf{A})}{\partial \mathbf{A}_{dg}} = \frac{\partial g(\mathbf{A})}{\partial \mathbf{A}_{dg}} + \frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}}$$
$$= C_d \mathbf{A}_{dg} - \mathbf{B}_{dg} + \lambda w_g \frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}},$$

where
(3.13)

$$\frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}} = \begin{cases} \frac{\mathbf{A}_{dg}}{\|\mathbf{A}_{dg}\|_2} & \mathbf{A}_{dg} \neq \mathbf{0} \\ \{\boldsymbol{\alpha} \in \mathbb{R}^{|g|} | \|\boldsymbol{\alpha}\|_2 \leq 1\} & \mathbf{A}_{dg} = \mathbf{0} \end{cases}$$

According to Theorem 3.1.15 in [18], $\mathbf{A}_{dg}^*$ is the optimal solution if an only if $0 \in \frac{\partial f(\mathbf{A}_{dg})}{\partial \mathbf{A}_{dg}}$ at $\mathbf{A}_{dg}^*$. Therefore, the closed-form solution of $\mathbf{A}_{dg}^*$ can be given in the following proposition.

Algorithm 2 Optimization for $\mathbf{A}$ with group structure

Input: $\mathbf{X}$, $\mathbf{U}$, the dimensionality of the latent space D, the global regularization parameter λ, group structure $\mathcal{G}$, regularization weights of groups $\{w_g\}_{g \in \mathcal{G}}$.

  while $\mathbf{A}$ has not converged **do**
    for $d = 1, 2, \ldots, D$ **do**
      Compute $C_d = \sum_{i=1}^{N} u_{id}^2$
      for all $g \in \mathcal{G}$ **do**
        Compute $\mathbf{B}_{dg}$ according to Eq. (3.11)
        if $\|\mathbf{B}_{dg}\|_2 > \lambda w_g$ **then**
          $\mathbf{A}_{dg} \leftarrow \frac{\mathbf{B}_{dg}(\|\mathbf{B}_{dg}\|_2 - \lambda w_g)}{C_d \|\mathbf{B}_{dg}\|_2}$
        else
          $\mathbf{A}_{dg} \leftarrow \mathbf{0}$
        end if
      end for
    end for
  end while

Output: Sparse projection matrix $\mathbf{A}$.

PROPOSITION 3.1. *The optimal $\mathbf{A}_{dg}^*$ for the minimization of $f(\mathbf{A})$ with respect to $\mathbf{A}_{dg}$ takes the following form:*

$$(3.14) \quad \mathbf{A}_{dg}^* = \begin{cases} \frac{\mathbf{B}_{dg}(\|\mathbf{B}_{dg}\|_2 - \lambda w_g)}{C_d \|\mathbf{B}_{dg}\|_2} & \|\mathbf{B}_{dg}\|_2 > \lambda w_g \\ \mathbf{0} & \|\mathbf{B}_{dg}\|_2 \leq \lambda w_g \end{cases}.$$

The proof is shown in the appendix.

The entire block coordinate descent for optimizing $\mathbf{A}$ is summarized in Algorithm 2.

3.2 Non-negative Sparse LSA It is natural to assume that each word has a non-negative contribution to a specific topic, i.e. the projection matrix $\mathbf{A}$ should be non-negative. In such a case, we may normalize each row of $\mathbf{A}$ to 1:

$$\widetilde{a}_{dj} = \frac{a_{dj}}{\sum_{j=1}^{M} a_{dj}}.$$

Since a_{dj} measures the relevance of the j-th word, w_j, to the d-th topic t_d, from the probability perspective, $\widetilde{a}_{dj}$ can be viewed as a pseudo probability of the word w_j given the topic t_d, $\mathbb{P}(w_j|t_d)$. Similar to topic modeling in the Bayesian framework such as LDA [3], the non-negative Sparse LSA can also provide the most relevant/likely words to a specific topic.

More formally, the non-negative Sparse LSA can be formulated as the following optimization problem:

$$(3.15) \qquad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda \|\mathbf{A}\|_1$$
$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \quad \mathbf{A} \geq 0.$$

According to the non-negativity constraint of $\mathbf{A}$, $|a_{dj}| = a_{dj}$ and Eq. (3.15) is equivalent to:

$$(3.16) \quad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda \sum_{d=1}^{D}\sum_{j=1}^{J} a_{dj}$$

$$\text{subject to:} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I}, \quad \mathbf{A} \geq 0.$$

We still apply the alternating approach to solve Eq. (3.16). When $\mathbf{A}$ is fixed, the optimization with respect to $\mathbf{U}$ is the same as that in Eq. (2.6). When $\mathbf{U}$ is fixed, following in the same strategy as in Section 2.3, the optimization over $\mathbf{A}$ can be decomposed in to M independent subproblems, each one corresponds to a column of $\mathbf{A}$:

$$(3.17) \quad \min_{A_j \geq \mathbf{0}} f(\mathbf{A}_j) = \frac{1}{2}\|X_j - \mathbf{U}A_j\| + \lambda \sum_{d=1}^{D} a_{dj}.$$

We still apply the coordinate descent method to minimize $f(\mathbf{A}_j)$: we fix a dimension of the latent space d; optimize $f(\mathbf{A}_j)$ with respect to a_{dj} while keep other entries in $f(\mathbf{A}_j)$ fixed and iterate over d. More precisely, for a fixed d, the gradient of $f(\mathbf{A}_j)$ with respect to a_{dj}:

$$(3.18) \quad \frac{\partial f(\mathbf{A}_j)}{\partial a_{dj}} = c_d a_{dj} - b_d + \lambda,$$

where $c_d = \sum_{i=1}^{N} u_{id}^2$,

$$b_d = \sum_{i=1}^{N} u_{id}\left(x_{ij} - \sum_{k \neq d} u_{ik} a_{kj}\right).$$

It is easy to verify that when $b_d > \lambda$, setting $a_{dj} = \frac{b_d - \lambda}{c_d}$ will make $\frac{\partial f(\mathbf{A}_j)}{\partial a_{dj}} = 0$. On the other hand, if $b_d \leq \lambda$, $\frac{\partial f(\mathbf{A}_j)}{\partial a_{dj}} \geq 0$ for all $a_{dj} \geq 0$, which further implies that $f(\mathbf{A}_j)$ is a monotonic increasing function with a_{dj} and the minimal is achieved when $a_{dj} = 0$. We summarize the optimal solution a_{dj}^* in the following proposition.

PROPOSITION 3.2. *The optimal a_{dj}^* for the minimization $f(A_j)$ with respect to a_{dj} takes the following form:*

$$(3.19) \quad a_{dj}^* = \begin{cases} \frac{b_d - \lambda}{c_d} & b_d > \lambda \\ 0 & b_d \leq \lambda \end{cases}.$$

4 Related Work

There exist numerous related works in a larger context of the matrix factorization. Here, we briefly review those work mostly related to us and point out the difference from our model.

4.1 PCA Principal component analysis (PCA) [9], which is closely related to LSA, has been widely applied for the dimension reduction purpose. In the content of information retrieval, PCA first center each document by subtracting the sample mean. The resulting document-term matrix is denoted as $\mathbf{Y}$. PCA computes the covariance matrix $\Sigma = \frac{1}{N}\mathbf{Y}^T\mathbf{Y}$ and performs SVD on Σ keeping only the first D eigenvalues: $\Sigma \approx \mathbf{P}\Delta_{D \times D}\mathbf{P}^T$. For each centered document $\mathbf{y}$, its projected image is $\mathbf{P}^T\mathbf{y}$. In recent years, many variants of PCA, including kernel PCA [21], sparse PCA [26], non-negative sparse PCA [25], robust PCA [14], have been developed and successfully applied in many areas.

However, it is worthy to point out that the PCA based dimension reduction techniques are not suitable for the large text corpus due to the following two reasons:

1. When using English words as features, the text corpus represented as $\mathbf{X}$ is a highly sparse matrix where each rows only has a small amount of nonzero entries corresponding to the words appeared in the document. However, by subtracting the sample mean, the centered documents will become a dense matrix which may not fit into memory since the number of documents and words are both very large.

2. PCA and its variants, e.g. sparse PCA, rely on the fact that $\mathbf{Y}^T\mathbf{Y}$ can be fit into memory and SVD can be performed on it. However, for large vocabulary size M, it is very expensive to store M by M matrix and computationally costly to perform SVD on $\mathbf{Y}^T\mathbf{Y}$.

In contrast with PCA and its variants (e.g. sparse PCA), our method directly works on the original sparse matrix without any standardization or utilizing the covariance matrix, hence is more suitable for the text learning task.

4.2 Sparse Coding Sparse coding, as another unsupervised learning algorithm, learns basis functions which capture higher-level features in the data and has been successfully applied to image processing [19] and speech recognition [8]. Although the original form of sparse coding is formulated based on the term-document matrix, for the easy of comparison, in our notations, sparse coding [16] can be modeled as:

$$(4.20) \quad \min_{\mathbf{U},\mathbf{A}} \quad \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda\|\mathbf{U}\|_1$$

$$\text{subject to:} \quad \|A_j\|_2^2 \leq c, \quad j = 1, \ldots M,$$

where c is a predefined constant, $\mathbf{A}$ is called dictionary in sparse coding context; and $\mathbf{U}$ are the coefficients.

Instead of projecting the data via $\mathbf{A}$ as our method, sparse coding directly use $\mathbf{U}$ as the projected image of $\mathbf{X}$. Given a new data q, its projected image in the latent space can be computed by solving the following lasso type of problem:

$$(4.21) \qquad \min_{\widehat{q}} \frac{1}{2}\|q - \mathbf{A}^T\widehat{q}\| + \lambda\|\widehat{q}\|_1.$$

In the text learning, one drawback is that since the dictionary $\mathbf{A}$ is dense, it is hard to characterize the topic-word relationships from $\mathbf{A}$. Another drawback is that for each new document, the projection operation in Eq. (4.21) is computationally very expensive.

4.3 LDA Based on the LSA, *probabilistic LSA* [10] was proposed to provide the probabilistic modeling, and further *Latent Dirichlet Allocation* (LDA) [3] provides a Bayesian treatment of the generative process. One great advantage of LDA is that it can provide the distribution of words given a topic and hence rank the words for a topic. The non-negative Sparse LSA proposed in Section 3.2 can also provide the most relevant words to a topic and can be roughly viewed as a discriminative version of LDA. However, when the number of latent topics D is very large, LDA performs poorly and the posterior distribution is almost the same as prior. On the other hand, when using smaller D, the documents in the latent topic space generated by LDA are not discriminative for the classification or categorization task. In contrast, as we show in experiments, our method greatly outperforms LDA in the classification task while providing reasonable ranking of the words for a given topic.

4.4 Matrix Factorization Our basic model is also closely related to matrix factorization which finds the low-rank factor matrices $\mathbf{U}$, $\mathbf{A}$ for the given matrix $\mathbf{X}$ such that the approximation error $\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2$ is minimized. Important extensions of matrix factorization include non-negative matrix factorization [15], which enforces the non-negativity constraint to $\mathbf{X}$, $\mathbf{U}$ and $\mathbf{A}$; probabilistic matrix factorization [20], which handles the missing values of $\mathbf{X}$ and becomes one of the most effective algorithms in collaborative filtering; sparse non-negative matrix factorization [11, 13], which enforces sparseness constraint on either $\mathbf{U}$ or $\mathbf{A}$; and orthogonal non-negative matrix factorization [6], which enforces the non-negativity and orthogonality constraints simultaneously on $\mathbf{U}$ and/or $\mathbf{A}$ and studies its relationship to clustering.

As compared to sparse non-negative matrix factorization [11, 13], we add orthogonality constraint to the $\mathbf{U}$ matrix, i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, which enforces the soft clustering effect of the documents. More specifically, each dimension of the latent space (columns of $\mathbf{U}$) can be viewed as a cluster and the value that a document has on that dimension as its fractional membership in the cluster. The orthogonality constraint tries to cluster the documents into different latent topics. As compared to orthogonal non-negative matrix factorization [6], instead of enforcing both non-negativity and orthogonality constraints, we only enforce orthogonality constraint on $\mathbf{U}$, which further leads to a closed-form solution for optimizing $\mathbf{U}$ as shown in Theorem 2.1. In summary, based on the basic matrix factorization, we combine the orthogonality and sparseness constraints into a unified framework and use it for the purpose of semantic analysis. Another difference between Sparse LSA and matrix factorization is that, instead of treating $\mathbf{A}$ as factor matrix, we use $\mathbf{A}$ as the projection matrix.

5 Experimental Results

In this section, we conduct several experiments on real world datasets to test the effectiveness of Sparse LSA and its extensions.

5.1 Text Classification Performance In this subsection, we consider the text classification performance after we project the text data into the latent space. We use two widely adopted text classification corpora, 20 Newsgroups (20NG) dataset [2] and RCV1 [17]. For the 20NG, we classify the postings from two newsgroups *alt.atheism* and *talk.religion.misc* using the tf-idf of the vocabulary as features. For RCV1, we remove the words appearing fewer than 10 times and standard stopwords; pre-process the data according to [2] [3]; and convert it into a 53 classes classification task. More statistics of the data are shown in Table 1.

We evaluate different dimension reduction techniques based on the classification performance of linear SVM classifier. Specifically, we consider

1. Traditional LSA;
2. Sparse Coding with the code from [16] and the regularization parameter is chosen by cross-validation on train set;
3. LDA with the code from [3];
4. Sparse LSA;
5. Non-negative Sparse LSA (NN Sparse LSA).

After projecting the documents to the latent space, we randomly split the documents into training/testing

Table 1: The statistics of the experimental datasets

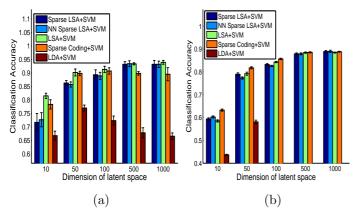|	20NG	RCV1
No. of Samples	1425	15,564
No. of Words	17,390	7,413
No. of Classes	2	53

(a)

(b)

Figure 2: Classification accuracy vs the dimensionality of latent space for (a) 20NG; (b) RCV1.

Table 2: Density of $\mathbf{A}$ (%)

(a) 20NG

Dimension	10	50	100	500	1000
Sparse LSA	1.48	0.80	0.74	0.32	0.18
NN Sparse LSA	1.44	0.72	0.55	0.31	0.17
Other Methods	100	100	100	100	100

(b) RCV1

Dimension	10	50	100	500	1000
Sparse LSA	13.52	7.46	7.40	2.71	1.13
NN Sparse LSA	11.65	4.97	0.40	1.91	0.79
Other Methods	100	100	100	100	100

(a)

(b)

Figure 3: Classification Accuracy vs effective dimension for (a) 20NG (b) RCV1

set with the ratio 2 : 1 and perform the linear SVM using the package LIBSVM [4] with the regularization parameter $C_{\mathrm{svm}} \in \{1e-4, \ldots, 1e+4\}$ selected by 5-fold cross-validation.

Firstly, following the traditional way of comparing different dimension reduction methods, we vary the dimensionality of the latent space and plot the classification accuracy in Figure 2. For Sparse LSA and NN Sparse LSA, the regularization parameter λ is fixed to be 0.05 and the corresponding densities (proportion of nonzero entries) of $\mathbf{A}$ are shown in Table 2.
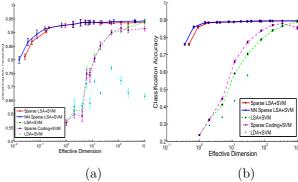
It can be seen that the performances of LSA and

Sparse Coding are comparable. Sparse Coding is slightly worse than LSA for 20NG while slightly better for RCV1. When the dimensionality of latent space is small, Sparse LSA is slightly worse than LSA. It is expectable since the number of parameters in the projection matrix is already very few, sparse model may miss important topic-word relationships. In contrast, for higher dimensional latent space, Sparse LSA shows its advantage in the sense that it can achieve similar classification performance with a highly sparse model (see Table 2). A sparse $\mathbf{A}$ will further save both computational and storage cost as shown in the next section. NN Sparse LSA achieves similar classification performance as Sparse LSA with a more sparse $\mathbf{A}$. LDA performs not well for the classification task, which is also expectable since LDA is a generative model designed for the better interpretability instead of dimension reduction performance [4].

Since Sparse LSA has fewer effective parameters (nonzero entries) in projection matrix, for more fair comparisons, we introduce a concept called *effective dimension* which has been widely adopted in sparse learning. We define the effective dimension of $\mathbf{A}$ to be $\frac{\#\mathrm{nz}(\mathbf{A})}{M}$, where $\#\mathrm{nz}(\mathbf{A})$ is the number of nonzero entries of $\mathbf{A}$ and M is the vocabulary size [5]. For other methods, including LSA, Sparse Coding, LDA, the effective dimension is just the dimensionality of the latent space since all the parameters affects the projection operation. In other words, we compare the classification performance of different methods based on the same number of learned nonzero parameters for the projection.

---

[4]For RCV1 using LDA, when the dimensionality of the latent space exceeds 100, the classification performance is very poor (nearly random guess). Therefore, we omit the result here.

[5]Effective dimension might be less than 1 if $\#\mathrm{nz}(\mathbf{A}) < M$.

Table 3: Computational Efficiency and Storage

(a) 20NG

	Proj. Time (ms)	Storage (MB)
Sparse LSA	0.25 (4.05E-2)	0.6314
NN Sparse LSA	0.22 (2.78E-2)	0.6041
LSA	31.6 (1.10)	132.68
Sparse Coding	1711.1 (323.9)	132.68
	Density of Proj. Doc. (%)	Acc. (%)
Sparse LSA	35.81 (15.39)	93.01 (1.17)
NN Sparse LSA	35.44 (15.17)	93.00 (1.14)
LSA	100 (0)	93.89 (0.58)
Sparse Coding	86.94 (3.63)	90.54 (1.55)

(b) RCV1

	Proj. Time (ms)	Storage (MB)
Sparse LSA	0.59 (7.36E-2)	1.3374
NN Sparse LSA	0.46 (6.66E-2)	0.9537
LSA	13.2 (0.78)	113.17
Sparse Coding	370.5 (23.3)	113.17
	Density of Proj. Doc. (%)	Acc. (%)
Sparse LSA	55.38 (11.77)	88.88 (0.43)
NN Sparse LSA	46.47 (11.90)	88.97 (0.49)
LSA	100 (0)	89.38 (0.58)
Sparse Coding	83.88 (2.11)	88.79 (1.55)

The result is shown in Figure 3. For Sparse LSA and NN Sparse LSA, we fix the number of latent topics to be $D = 1000$ and vary the value of regularization parameter λ from large number (0.5) to small one (0) to achieve different #nz($\mathbf{A}$), i.e. different effective dimensions. As we can see, Sparse LSA and NN Sparse LSA greatly outperform other methods in the sense that they achieve good classification accuracy even for highly sparse models. In practice, we should try to find a λ which could lead to a sparser model while still achieving reasonably good dimension reduction performance.

In summary, Sparse LSA and NN Sparse LSA show their advantages when the dimensionality of latent space is large. They can achieve good classification performance with only a small amount of nonzero parameters in the projection matrix.

5.2 Efficiency and Storage In this section, we fix the number of the latent topics to be 1000, regularization parameter $\lambda = 0.05$ and report the projection time, storage and the density of the projected documents for different methods in Table 3[6]. The Proj. time is computed as the CPU time for the projection operation and the density of projected documents is the proportion of nonzero entries of $\hat{q} = \mathbf{A}q$ for a document q. Both quantities are computed for 1000 randomly selected documents in the corpus. Storage is the memory for storing the $\mathbf{A}$ matrix.

For Sparse LSA and NN Sparse LSA, although the classification accuracy is slightly worse (below 1%), the projection time and memory usage are smaller by orders of magnitude than LSA and Sparse Coding. In practice, if we may need to project millions of documents, e.g. web-scale data, into the latent space in a timely fashion (online setting), Sparse LSA and NN Sparse LSA will greatly cut computational cost. Moreover, given a new document q, using Sparse LSA or NN Sparse LSA, the projected document will also be a sparse vector.

5.3 Topic-word Relationship In this section, we qualitatively show that the topic-word relationship learned by NN Sparse LSA as compared to LDA. We use the benchmark data: NIPS proceeding papers[7] from 1988 through 1999 of 1714 articles, with a vocabulary 13,649 words. We vary the λ for NN Sparse LSA so that each topic has at least ten words. The top ten words for the top 7 topics [8] are listed in Table 4.

It is very clear that NN Sparse LSA captures different hot topics in machine learning community in 1990s, including neural network, reinforcement learning, mixture model, theory, signal processing and computer vision. For the ease of comparison, we also list the top 7 topics for LDA as in Table 5. Although LDA also gives the representative words, the topics learned by LDA are not very discriminative in the sense that all the topics seems to be closely related to neural network.

5.4 Gene Function Identification with Gene Groups Information For text retrieval task, it is not obvious to identify the separated group structures among words. Instead, one important application for the group structured Sparse LSA is in gene-set identifications associated to hidden functional structures inside cells. Genes could be naturally separated into groups according to their functions or locations, known as pathways. We use a benchmark breast cancer dataset from [12], which includes a set of cancer tumor examples and each example is represented by a vector of real values,

---

[6] "Proj.", "Doc", "ACC." are abbreviations for "projection/projected", "document" and "classification accuracy", respectively.

[7] Available at http://cs.nyu.edu/~roweis/data/

[8] We use $D = 10$. However, due to the space limit, we report the top 7 topics.

Table 4: Topic-word learned by NN Sparse LSA

Topic 1	Topic 2	Topic 3	Topic 4
network	learning	network	model
neural	reinforcement	learning	data
networks	algorithm	data	models
system	function	neural	parameters
neurons	rule	training	mixture
neuron	control	set	likelihood
input	learn	function	distribution
output	weight	model	gaussian
time	action	input	em
systems	policy	networks	variables

Topic 5		Topic 6		Topic 7	
function		input		image	
functions		output		images	
approximation		inputs		recognition	
linear		chip		visual	
basis		analog		object	
threshold		circuit		system	
theorem		signal		feature	
loss		current		figure	
time		action		input	
systems		policy		networks	

Table 5: Topic-word learned by LDA

Topic 1	Topic 2	Topic 3	Topic 4
learning	figure	algorithm	single
data	model	method	general
model	output	networks	sets
training	neurons	process	time
information	vector	learning	maximum
number	networks	input	paper
algorithm	state	based	rates
performance	layer	function	features
linear	system	error	estimated
input	order	parameter	neural

Topic 5		Topic 6		Topic 7	
rate		algorithms		function	
unit		set		neural	
data		problem		hidden	
time		weight		networks	
estimation		temporal		recognition	
node		prior		output	
set		obtain		visual	
input		parameter		noise	
neural		neural		parameters	
properties		simulated		references	

e.g. the quantities of different genes found in the data example. Essentially, group structured Sparse LSA analyzes relationships between the cancer examples and genes they contain by discovering a set of "hidden gene functions" (i.e. topics in text case) related to the cancer and the gene groups. And it is of great interest for biologists to determine which sets of gene groups, instead of individual genes, associate to the same latent functions relevant to a certain disease.

Specifically, the benchmark cancer data consists of gene expression values from 8141 genes in 295 breast cancer examples (78 metastatic and 217 non-metastatic). Based each gene's associated "biological process" class in the standard "gene ontology" database [22], we split these 8141 genes into 1689 non-overlapped groups, which we use as group structures in applying group structured Sparse LSA.

We set the parameter $\lambda = 1.2$ and select the first three projected "functional" components which are relevant to 45 gene groups totally. The selected gene groups make a lot of sense with respect to their association with the breast cancer disease. For instance, 12 gene groups are relevant to the second hidden "function". One se-

lected group covering 10 gene variables involves in the so called "cell cycle M phase". The cell cycle is a vital process by which a single-celled fertilized egg develops into a mature organism, or by which hair, skin, blood cells, and some internal organs are renewed. Cancer is a disease where regulation of the cell cycle goes wrong and normal cell growth and behavior is lost. When the cells multiply uncontrollably, there forms a tumor. Thus, the association of this important process to cancer seems very reasonable. Other chosen groups involve functional enrichments of "cytoskeleton organization", "regulation of programmed cell death" or "microtubule-based process", etc. Clearly this hidden function (the 2nd projection) space involves the critical "cell cycle" components relevant to important regulatory changes leading to characteristic cell grow and death. With similar analyses, we found that the first projection maps to the space of "immune system" processes (e.g. response to hormone stimulus) and the third hidden function factor involves in the "extracellular matrix space" among which gene products are not uniformly attached to the cell surface.

Alternatively, we also perform the dimension reduc-

tion on this cancer data using the basic Sparse LSA without considering the group structures among genes. The resulting functional components could not be analyzed as clear and as easy as the group structured Sparse LSA case. The reason of the difficulty is that the discovered gene functions are quite large gene groups (i.e. more than 100 genes involved). They represent relatively high level biological processes which are essential for cells in any case, but not necessarily limited to the certain cancer disease this data set is about. It is hard to argue the relationship between such a large amount of genes to the target "breast cancer" cause.

6   Conclusion

In this paper, we introduce a new model called Sparse Latent Semantic Analysis, which enforces the sparsity on the projection matrix based on LSA using the ℓ_1 regularization. Sparse LSA could provide a more compact and precise projection by selecting only a small number of relevant words for each latent topic. We further propose two important extensions of Sparse LSA, group structured LSA and non-negative Sparse LSA. A simple yet efficient alternating algorithm is adopted to learn the sparse projection matrix for all these models. We conduct experiments on several real-world datasets to illustrate the advantages of our model from different perspectives. In the future, we plan to improve the scalability of Sparse LSA. One possible direction is to utilize the online learning scheme to learn web-scale datasets.

References

[1] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Addison-Wesley Harlow, England, 1999.

[2] R. Bekkerman and M. Scholz. Data weaving: Scaling up the state-of-the-art in data clustering. In *Proceedings of ACM International Confernece on Information and Knowledge Management*, 2008.

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

[4] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[5] S. Deerwester, S. T. Dumais, G. W. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, 1990.

[6] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *ACM SIGKDD*, 2006.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.

[8] S. Garimella, S. Nemala, M. Elhilali, T.Tran, and H. Hermansky. Sparse coding for speech recognition. In *IEEE International Confernece on Acoustics, Speech and Signal Processing*, 2010.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[10] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 289–296, 1999.

[11] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.

[12] L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. In *Proceedings of International Conference on Machine Learning*, 2009.

[13] H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23:1495–1502, 2007.

[14] F. D. la Torre and M. Black. Robust principal component analysis for computer vision. In *Internation Conference on Computer Vision*, 2001.

[15] D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[16] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[17] D. Lewis, Y.Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization. *Journal of Machine Learning Research*, 5:361–397, 2004.

[18] Y. Nesterov. *Introductory lectures on convex optimization: a basic course*. Kluwer Academic Publishers, 2003.

[19] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

[20] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, 2007.

[21] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. In *Advances in Kernel Methods—Support Vector Learning*. MIT Press, 1999.

[22] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–9, 2000.

[23] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

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

[25] R. Zass and A. Shashua. Nonnegative sparse pca. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[26] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal

component analysis. *Journal of Computational and Graphical Statistics*, 15, 2004.

7 Appendix

7.1 Proof of Theorem 2.1

The vector form of optimization problem in Eq. (2.7) is:

$$\text{(7.22)} \qquad \min_{\mathbf{U}} \qquad -\sum_{d=1}^{D} (U_d)^T V_d$$

$$\text{(7.23)} \quad \text{subject to:} \quad (U_d)^T U_d = 1, \quad d = 1, \dots, D$$
$$\text{(7.24)} \qquad\qquad\qquad (U_k)^T U_l = 0 \quad k \neq l$$

Associate the Lagrangian multipliers θ_{dd} to the constraints in (7.23) and θ_{kl} to the constraints (7.24). Suppose $\mathbf{U}^*$ is the optimal solution to Eq. (7.22), according to the KKT condition, we have

$$\text{(7.25)} \qquad \frac{\partial L}{\partial U_d}\mid_{U=U^*} = 0, \quad d = 1, \dots, D,$$

where

$$L = -\sum_{d=1}^{D} (U_d)^T V_d + \sum_{d=1}^{D} \theta_{dd}((U_d)^T U_d - 1) + \sum_{k \neq l} \theta_{kl}(U_k)^T U_l.$$

And Eq. (7.25) gives

$$V_d = \sum_{k=1}^{D} \theta_{dk} U_k^*, \quad d = 1, \dots D$$

or simply $\mathbf{V} = \mathbf{U}^* \boldsymbol{\Theta}$, where $\boldsymbol{\Theta}_{kl} = \theta_{lk}$. According the symmetry that $(U_k)^T U_l = (U_l)^T U_k$, $\boldsymbol{\Theta}$ is a symmetric matrix. Since $\mathbf{U}^*$ and $\mathbf{V}$ are all full column rank, Θ is invertible. By the assumption that $\mathbf{V} = \mathbf{P}\Delta\mathbf{Q}$ with $\mathbf{P}^T \mathbf{P} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$, $\mathbf{U}^* = \mathbf{V}\boldsymbol{\Theta}^{-1} = \mathbf{P}\Delta\mathbf{Q}\boldsymbol{\Theta}^{-1}$. So we only need to obtain $\boldsymbol{\Theta}$ to get the optimal $\mathbf{U}^*$.

To compute $\boldsymbol{\Theta}$, plugging $\mathbf{U}^* = \mathbf{P}\Delta\mathbf{Q}\boldsymbol{\Theta}^{-1}$ and $\mathbf{V} = \mathbf{P}\Delta\mathbf{Q}$ back into the original optimization problem in Eq. (2.7), the objective now becomes:

$$\begin{aligned}
\text{(7.26)} \quad \text{tr}((\mathbf{U}^*)^T \mathbf{V}) &= \text{tr}(\boldsymbol{\Theta}^{-1}\mathbf{Q}^T\Delta\mathbf{P}^T\mathbf{P}\Delta\mathbf{Q}) \\
&= \text{tr}(\boldsymbol{\Theta}^{-1}\mathbf{Q}^T\Delta^2\mathbf{Q}) \\
&= \text{tr}(\mathbf{Q}\boldsymbol{\Theta}^{-1}\mathbf{Q}^T\Delta^2)
\end{aligned}$$

Define $\mathbf{R} \equiv \mathbf{Q}\boldsymbol{\Theta}^{-1}\mathbf{Q}^T$, Eq. (7.26) becomes:

$$\text{tr}((\mathbf{U}^*)^T \mathbf{V}) = \text{tr}(\mathbf{R}\Delta^2) = \sum_{d=1}^{D} r_{dd}\delta_{dd}^2.$$

According to the constraint that $(\mathbf{U}^*)^T\mathbf{U}^* = \mathbf{I}$:

$$\begin{aligned}
\text{(7.27)} \quad \mathbf{I} &= \mathbf{Q}\mathbf{I}\mathbf{Q}^T = \mathbf{Q}(\mathbf{U}^*)^T\mathbf{U}^*\mathbf{Q}^T \\
&= \mathbf{Q}\boldsymbol{\Theta}^{-1}\mathbf{Q}^T\Delta\mathbf{P}^T\mathbf{P}\Delta\mathbf{Q}\boldsymbol{\Theta}^{-1}\mathbf{Q}^T \\
&= \mathbf{R}^T\Delta^2\mathbf{R}
\end{aligned}$$

From Eq. (7.27), for any d, we have $\sum_{k=1}^{D} r_{dk}^2\delta_{kk}^2 = 1$ which further implies that:

$$\text{(7.28)} \qquad r_{dd}^2\delta_{dd}^2 \leq \sum_{k=1}^{D} r_{dk}^2\delta_{kk}^2 = 1.$$

Therefore we have $r_{dd}\delta_{dd} \leq 1$ and upper bound of the objective value is:

$$\text{tr}((\mathbf{U}^*)^T\mathbf{V}) = \sum_{d=1}^{D} r_{dd}\delta_{dd}^2 \leq \sum_{d=1}^{D} \delta_{dd},$$

The equality holds if and only if for any $d = 1, \dots, D$, Eq. (7.28) holds as equality which further implies $\mathbf{R}$ is diagonal and $\mathbf{R} = \Delta^{-1}$.

According to the definition of $\mathbf{R}$, $\mathbf{Q}\boldsymbol{\Theta}^{-1} = \mathbf{R}\mathbf{Q} = \Delta^{-1}\mathbf{Q}$ and the optimal solution

$$\mathbf{U}^* = \mathbf{P}\Delta\mathbf{Q}\boldsymbol{\Theta}^{-1} = \mathbf{P}\Delta\Delta^{-1}\mathbf{Q} = \mathbf{P}\mathbf{Q}.$$

7.2 Proof of Proposition 3.1

According to Theorem 3.1.15 in [18], $\mathbf{A}_{dg}$ is the optimal solution if an only if

$$\text{(7.29)} \quad 0 \in \frac{\partial f(\mathbf{A}_{dg})}{\partial \mathbf{A}_{dg}} = C_d\mathbf{A}_{dg} - \mathbf{B}_{dg} + \lambda w_g \frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}}.$$

Assume $\mathbf{A}_{dg} \neq \mathbf{0}$, plugging $\frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}} = \frac{\mathbf{A}_{dg}}{\|\mathbf{A}_{dg}\|_2}$ into Eq. (7.29), we have:

$$\text{(7.30)} \qquad \mathbf{A}_{dg}(C_d + \lambda w_g \frac{1}{\|\mathbf{A}_{dg}\|_2}) = \mathbf{B}_{dg}$$

Taking the vector ℓ_2-norm on both side of Eq. (7.30):

$$\text{(7.31)} \qquad \|\mathbf{A}_{dg}\|_2 = \frac{\|\mathbf{B}_{dg}\|_2 - \lambda w_g}{C_d}.$$

Since $\mathbf{A}_{dg} \neq \mathbf{0}$, $\|\mathbf{A}_{dg}\|_2$ should be a positive real value, which requires that $\|\mathbf{B}_{dg}\|_2 > \lambda w_g$. Plugging Eq. (7.31) back into Eq. (7.30), we obtain that if the condition $\|\mathbf{B}_{dg}\|_2 > \lambda w_g$ holds:

$$\mathbf{A}_{dg} = \frac{\mathbf{B}_{dg}(\|\mathbf{B}_{dg}\|_2 - \lambda w_g)}{C_d\|\mathbf{B}_{dg}\|_2}.$$

On the other hand, if $\|\mathbf{B}_{dg}\|_2 \leq \lambda w_g$, plugging $\mathbf{A}_{dg} = \mathbf{0}$ into Eq. (7.29), we obtain that $\frac{\partial \|\mathbf{A}_{dg}\|_2}{\partial \mathbf{A}_{dg}} = \frac{\mathbf{B}_{dg}}{\lambda w_g}$. It is easy to verify that $\frac{\mathbf{B}_{dg}}{\lambda w_g}$ is a valid subgraident of $\|\mathbf{A}_{dg}\|_2$ when $\mathbf{A}_{dg} = \mathbf{0}$ since $\|\mathbf{B}_{dg}\|_2 \leq \lambda w_g$.