Sparse Latent Semantic Analysis

Xi Chen [1] **Yanjun Qi**[2] **Bing Bai**[2] **Qihang Li**[1] **Jaime Carbonell**[1]

[1] Carnegie Mellon University, Pittsburg, PA
{xichen,qihangl,jgc}@cs.cmu.edu

[2]NEC Labs America, Princeton, NJ
{bbai,yanjun}@nec-labs.com

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 the nonnegative Sparse LSA as an extension of Sparse LSA. We conduct experiments on several benchmark datasets and compare Sparse LSA and its extension 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 hidden concepts from text, has widely been used for the dimension reduction purpose in information retrieval and text mining. A key component of LSA is to learn a projection matrix that converts the high dimensional vector space representations of documents to a lower dimensional space built with latent factors. These latent factors are learned from unlabeled data and called as latent *concepts or topics*.

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 latent factors. Based on this formulation, we add the *sparsity* constraint of the projection matrix via the ℓ_1 regularization as in the lasso model [11]. By enforcing the sparsity on the projection matrix, the model has the ability to automatically select the most relevant words for each latent topic. Furthermore, we propose one important extension based on Sparse LSA: Nonnegative Sparse LSA where we further enforce the nonnegativity constraint on the projection matrix. This could provide us a pseudo probability distribution of each word given the topic, similar to Latent Dirichlet Allocation (LDA) [3].

There exist numerous related work in a larger context of the matrix factorization. For instance, principal component analysis (PCA) [7], which is closely related to LSA, has been widely applied for the dimension reduction purpose. In the content of information retrieval, PCA first centers each document by subtracting the sample mean. This step makes PCA or its variants, e.g. sparse PCA [12] unsuitable for processing the large text corpus since (1) the centered document-term matrix will become a dense matrix which may not fit into memory since the number of documents and

words are both very large; (2) the convariance matrix required to be computed in PCA also gets very large for the computation and the storage. Other closely related models include sparse coding [9], probabilistic LSA [8], Latent Dirichlet Allocation (LDA) [3], and their variants. In Section 3, we compare Sparse LSA and its extension with these popular methods, e.g. LSA [5], Sparse Coding [9] 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 for the topics.

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. When dealing with text data, each latent dimension is also called as a hidden "topic".

Motivated by the latent factor analysis [7], 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 and let $\mathbf{U} = [U_1, \ldots, U_D] \in \mathbb{R}^{N \times D}$. We also assume that each feature vector X_j can be represented as a linear expansion in latent variables $U_1, \ldots, U_D$:

$$X_j = \sum_{d=1}^{D} a_{dj} U_d + \epsilon_j, \tag{1}$$

or simply $\mathbf{X} = \mathbf{U}\mathbf{A} + \boldsymbol{\epsilon}$ where $\mathbf{A} = [a_{dj}] \in \mathbb{R}^{D \times M}$ gives the mapping between the latent space and 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 $\mathbf{U}$:

$$\min_{\mathbf{U},\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 \quad \text{Subject to: } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \tag{2}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm and the constraint $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ guarantees that the latent variables are uncorrelated with the unit length.

At the optimum of Eq. (2), $\mathbf{U}\mathbf{A}$ leads to the best rank-D approximation of the data $\mathbf{X}$. In general, the larger 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: $\widehat{q} = \mathbf{A}q$, i.e. $\mathbf{A}$ could project documents into the vector representations on latent topic space.

2.2 Sparse LSA

Here we propose to add the *sparsity* constraint on the projection matrix $\mathbf{A}$ via the ℓ_1 regularization as in the lasso model [11]. We name this new model as Sparse Latent Semantic Analysis (Sparse LSA). To obtain a sparse $\mathbf{A}$, an entry-wise ℓ_1-norm of $\mathbf{A}$ is added as the regularization term to the loss function and this formulates the Sparse LSA model as:

$$\min_{\mathbf{U},\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_1 \quad \text{Subject to: } \mathbf{U}^T\mathbf{U} = \mathbf{I}, \tag{3}$$

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 3.

One benefit of sparse LSA is to improve LSA with easy interpretability on topic-word relationships. 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 to ease 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 (or meaningful). 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. On the contrary, for Sparse LSA, projecting documents via a highly sparse projection matrix will be much more computationally efficient; and it will take much less memory for storing $\mathbf{A}$ when D is large.

2.3 Optimization Algorithm

In this section, we propose an efficient optimization algorithm to solve Eq. (3). 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. (3) 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 $\mathbf{A}$:

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

can be decomposed in to M independent ones:

$$\min_{A_j} \frac{1}{2}\|X_j - \mathbf{U}A_j\|_2^2 + \lambda\|A_j\|_1; \quad j = 1, \ldots, M. \tag{4}$$

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. (4), we can directly apply the state-of-the-art lasso solver in [6] which is essentially a coordinate descent approach.

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

$$\min_{\mathbf{U}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 \text{ Subject to:} \mathbf{U}^T\mathbf{U} = \mathbf{I}. \tag{5}$$

The objective function in Eq. (5) can be further written as:

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

where the last equality is according to the constraint that $\mathbf{U}^T\mathbf{U} = \mathbf{I}$. By the fact that $\text{tr}(\mathbf{A}^T\mathbf{U}^T\mathbf{X}) \equiv \text{tr}(\mathbf{U}^T\mathbf{X}\mathbf{A}^T)$, the optimization problem in Eq. (5) is equivalent to

$$\max_{\mathbf{U}} \text{tr}(\mathbf{U}^T\mathbf{X}\mathbf{A}^T) \text{ subject to:} \mathbf{U}^T\mathbf{U} = \mathbf{I}. \tag{6}$$

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 $\text{rank}(\mathbf{V}) = D$, Eq. (6) 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.* (6) *is* $\mathbf{U} = \mathbf{P}\mathbf{Q}$.

The proof of the theorem is omitted due to space limit. 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 traditional 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: $\mathbf{U}^0 = \begin{pmatrix} \mathbf{I}_D \\ \mathbf{0} \end{pmatrix}$, where $\mathbf{I}_D$ the D by D identity matrix. It is easy to verify that $(\mathbf{U}^0)^T\mathbf{U}^0 = \mathbf{I}$.

2.4 Extension of Sparse LSA: Nonnegative Sparse LSA

In this section, we propose one important extension of Sparse LSA model. It is natural to assume that each word has a nonnegative contribution to a specific topic, i.e. the projection matrix $\mathbf{A}$ should be nonnegative. 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], this model, named as the nonnegative Sparse LSA, can also provide the most relevant/likely words to a specific topic. More formally, the nonnegative Sparse LSA can be formulated as the following optimization problem:

$$\min_{\mathbf{U},\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{U}\mathbf{A}\|_F^2 + \lambda\|\mathbf{A}\|_1 \quad \text{, Subject to:} \mathbf{U}^T\mathbf{U} = \mathbf{I}, \quad \mathbf{A} \geq 0. \tag{7}$$

The optimization problem in (7) can also be solved by an alternating scheme similar to Section 2.3.

3 Experimental Results

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

3.1 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 [1] and RCV1 [10]. 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] [2]; and convert it into a 53 classes classification task.

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 [9] and the regularization parameter is chosen by cross-validation on train set; (3) LDA with the code from [3]; (4) Sparse LSA; (5) Nonnegative Sparse LSA (NN Sparse LSA).

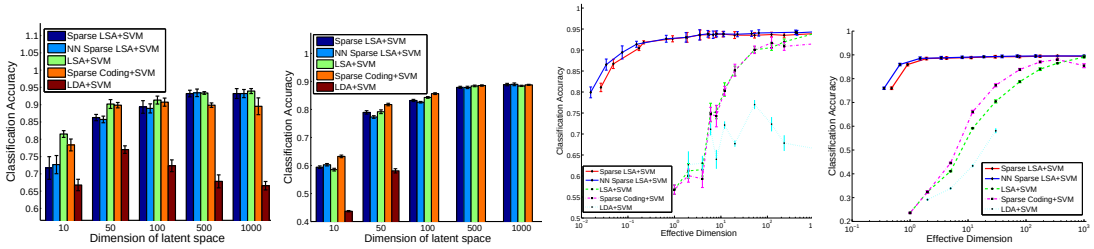
After projecting the documents to the latent space, we randomly split the documents into training/testing set with the ratio $2:1$ and perform the linear SVM using the package LIBSVM [4] with the regularization parameter $C_{\text{svm}} \in \{1e-4, \ldots, 1e+4\}$ selected by 5-fold cross-validation.

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 1 (a) & (b). 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 1.

It can be seen that the performances of LSA and Sparse Coding are comparable. When the dimensionality of latent space is small, Sparse LSA is slightly worse than LSA. This is expectable since when the number of parameters in its projection matrix is very small, the sparse model may miss important topic-word patterns. By contrast, when dealing with more latent topics, Sparse LSA

[1] See `http://people.csail.mit.edu/jrennie/20Newsgroups/`

[2] See `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#rcv1.multiclass`



(a) (b) (c) (d)

Figure 1: Classification accuracy vs the dimensionality of latent space for (a) 20NG; (b) RCV1. Classification Accuracy vs effective dimension for (a) 20NG (b) RCV1.

Table 1: Density of Projection Matrix $\mathbf{A}$ (%)

(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

10	50	100	500	1000
13.52	7.46	7.40	2.71	1.13
11.65	4.97	0.40	1.91	0.79
100	100	100	100	100

Table 2: Computational Efficiency and Storage on 20NG corpus

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

shows its advantage in the sense that it can achieve similar classification performance with a highly sparse model (see Table 1). 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 an even more sparse $\mathbf{A}$. LDA performs not very well for the text classification task, which is also expectable since LDA is a generative model designed for better interpretability instead of better dimension reduction performance.

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 $\frac{\#\text{nz}(\mathbf{A})}{M}$, where $\#\text{nz}(\mathbf{A})$ is the number of nonzero entries of $\mathbf{A}$ and M is the vocabulary size [3]. For other methods, 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.

The result is shown in Figure 1 (c) & (d). 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 $\#\text{nz}(\mathbf{A})$, 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.

3.2 Efficiency and Storage

In this section, we fix the number of latent topics as 1000, the regularization parameter as $\lambda = 0.05$ and report the projection time, storage and the density of the projected documents for different

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

Table 3: Topic-word learned by NN Sparse LSA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
network	learning	network	model	function	input	image
neural	reinforcement	learning	data	functions	output	images
networks	algorithm	data	models	approximation	inputs	recognition
system	function	neural	parameters	linear	chip	visual
neurons	rule	training	mixture	basis	analog	object
neuron	control	set	likelihood	threshold	circuit	system
input	learn	function	distribution	theorem	signal	feature
output	weight	model	gaussian	loss	current	figure
time	action	input	em	time	action	input
systems	policy	networks	variables	systems	policy	networks

Table 4: Topic-word learned by LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7
learning	figure	algorithm	single	rate	algorithms	function
data	model	method	general	unit	set	neural
model	output	networks	sets	data	problem	hidden
training	neurons	process	time	time	weight	networks
information	vector	learning	maximum	estimation	temporal	recognition
number	networks	input	paper	node	prior	output
algorithm	state	based	rates	set	obtain	visual
performance	layer	function	features	input	parameter	noise
linear	system	error	estimated	neural	neural	parameters
input	order	parameter	neural	properties	simulated	references

methods on 20NG corpus in Table 2. 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 $\widehat{q} = \mathbf{A}q$ for a document q. Both quantities are computed for 1000 randomly selected documents in the corpus. Storage means the size of the memory needed for storing the $\mathbf{A}$ matrix.

Using Sparse LSA and NN Sparse LSA, although the classification accuracy is slightly worse (difference $< 1\%$), the time and memory costs during projection are much lower by the orders of magnitudes than LSA and Sparse Coding. This means, if we need to project millions of documents, e.g. web-scale data, into the latent space representation in a timely fashion (e.g. online), Sparse LSA and its extension will be much more efficient. Moreover, given a new document q, under Sparse LSA, the projected document will also be a sparse vector, which is desired for efficient indexing.

3.3 Topic-word Relationship

In this section, we qualitatively compare the topic-word relationship learned by NN Sparse LSA to LDA. We use the benchmark data: NIPS proceeding papers [4] from 1988 to 1999 including 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 [5] are listed in Table 3. 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 4. 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.

4 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 conduct experiments on several real-world datasets to illustrate the advantages of our model from different perspectives.

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

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

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

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

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

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

[10] 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.

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

[12] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15, 2004.