Regularized Latent Semantic Indexing: A New Approach to Large Scale Topic Modeling

Quan Wang, MOE-Microsoft Key Laboratory of Statistics & Information Technology, Peking University, China
Jun Xu, Microsoft Research Asia, No. 5 Danling Street, Beijing, China[1]
Hang Li, Microsoft Research Asia, No. 5 Danling Street, Beijing, China[2]
Nick Craswell, Microsoft Cooperation, Bellevue, Washington, USA

Topic modeling provides a powerful way to analyze the content of a collection of documents. It has become a popular tool in research areas such as text mining, information retrieval, natural language processing, and other related fields. In real-world applications, however, the usefulness of topic modeling is limited due to scalability issues. Scaling to larger document collections via parallelization is an active area of research, but most solutions require drastic steps such as vastly reducing input vocabulary. In this paper we introduce Regularized Latent Semantic Indexing (RLSI) including a batch version and an online version, referred to as batch RLSI and online RLSI. Batch RLSI and online RLSI are as effective as existing topic modeling techniques, and can scale to larger datasets without reducing input vocabulary. Moreover, online RLSI can be applied to stream data and capture the dynamic evolution of topics. Both versions of RLSI formalize topic modeling as a problem of minimizing a quadratic loss function regularized by ℓ_1 and/or ℓ_2 norm. This formulation allows the learning process to be decomposed into multiple sub-optimization problems which can be optimized in parallel, for example via MapReduce. We particularly propose adopting ℓ_1 norm on topics and ℓ_2 norm on document representations, to create a model with compact and readable topics and useful for retrieval. In learning, batch RLSI processes all the documents in the collection as a whole, while online RLSI processes the documents in the collection one by one. We also prove the convergence of the learning of online RLSI. Relevance ranking experiments on three TREC datasets show that batch RLSI and online RLSI perform better than LSI, PLSI, LDA, and NMF, and the improvements are sometimes statistically significant. Experiments on a web dataset, containing about 1.6 million documents and 7 million terms, demonstrate a similar boost in performance.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Topic Modeling, Regularization, Sparse Methods, Distributed Learning, Online Learning

1. INTRODUCTION

Topic modeling refers to a suite of algorithms whose aim is to discover the hidden semantic structure in large archives of documents. Recent years have seen significant progress on topic modeling technologies in text mining, information retrieval, natural language processing, and other related fields. Given a collection of text documents each represented as a term vector, a topic model represents the relationship between terms and documents through latent topics. A topic is defined as a probability distribution over terms or a cluster of weighted terms. A document is viewed as a bag of terms generated from a mixture of latent topics[3]. Various topic modeling methods, such as Latent Semantic Indexing (LSI) [Deerwester et al. 1990], Probabilistic Latent Semantic Indexing (PLSI) [Hofmann 1999], and Latent Dirichlet Allocation (LDA) [Blei et al. 2003] have been proposed and successfully applied to different problems.

When applied to real-world tasks especially web applications, the usefulness of topic modeling is often limited due to scalability issues. For probabilistic topic modeling methods like LDA and PLSI, the scalability challenge mainly comes from the necessity of simultaneously updating the term-topic matrix to meet the probability distribution assumptions. When the number of terms is large, which is inevitable in real-world applications, this problem becomes particularly severe. For LSI, the challenge is due to the orthogonality assumption in the formulation, and as a result the problem needs to be solved by Singular Value Decomposition (SVD) and thus is hard to be parallelized. A

[1]Current affiliation: Noah's Ark Lab, Huawei Technologies Co. Ltd.
[2]Current affiliation: Noah's Ark Lab, Huawei Technologies Co. Ltd.
[3]We could train a topic model with phrases. In this paper, we take words as terms and adopt the bag of words assumption.

typical approach is to approximate the learning process of an existing topic model, but often tends to affect the quality of the learned topics.

In this work, instead of modifying existing methods, we introduce two new topic modeling methods that are intrinsically scalable: batch Regularized Latent Semantic Indexing (batch RLSI or bRLSI) for batch learning of topic models and online Regularized Latent Semantic Indexing (online RLSI or oRLSI) for online learning of topic models. In both versions of RLSI, topic modeling is formalized as minimization of a quadratic loss function regularized by ℓ_1 and/or ℓ_2 norm. Specifically, the text collection is represented as a term-document matrix, where each entry represents the occurrence (or tf-idf score) of a term in a document. The term-document matrix is then approximated by the product of two matrices: a term-topic matrix which represents the latent topics with terms and a topic-document matrix which represents the documents with topics. Finally, the quadratic loss function is defined as the squared Frobenius norm of the difference between the term-document matrix and the output of the topic model. Both ℓ_1 norm and ℓ_2 norm may be used for regularization. We particularly propose using ℓ_1 norm on topics and ℓ_2 norm on document representations, which can result in a model with compact and readable topics and useful for retrieval. Note that we call our new approach RLSI because it makes use of the same quadratic loss function as LSI. RLSI differs from LSI in that it uses regularization rather than orthogonality to constrain the solutions.

In batch RLSI, the whole document collection is represented in the term-document matrix and a topic model is learned from the matrix data. The algorithm iteratively updates the term-topic matrix with the topic-document matrix fixed, and updates the topic-document matrix with the term-topic matrix fixed. The formulation of batch RLSI makes it possible to conduct learning in parallel. This is achieved by decomposing the updates for both the term-topic matrix and the topic-document matrix into many sub-optimization problems. Running these in parallel is the main reason that batch RLSI can scale to large collections while retaining a large input vocabulary. We also propose an implementation of batch RLSI on MapReduce [Dean et al. 2004]. The MapReduce system maps the sub-optimization problems over multiple processors and then reduces the results from the processors. During this process, the documents and terms are automatically distributed and processed.

In online RLSI, the documents are input in a data stream and processed in a serial fashion. Online RLSI is a stochastic approximation of batch RLSI. It incrementally builds the topic model when new documents keep coming and thus is capable of capturing the evolution of the topics. Given a new document (or a set of new documents), online RLSI predicts the topic vector(s) of the new document(s) given the previously learned term-topic matrix, and then updates the term-topic matrix based on the new document(s) and the predicted topic vector(s). The formulation of online RLSI makes it possible to decompose the learning problem into multiple sub-optimization problems as well. Furthermore, online learning can make the algorithm scale up to larger datasets with limited storage. In that sense, online RLSI has an even better scalability than batch RLSI.

Regularization is a well-known technique in machine learning that penalizes complexity. In our setting, if we employ ℓ_2 norm on topics and ℓ_1 norm on document representations, batch RLSI becomes (batch) Sparse Coding (SC) [Lee et al. 2007; Olshausen and Fieldt 1997] and online RLSI becomes online SC [Mairal et al. 2010], which are methods used in computer vision and other related fields. However, regularization for topic modeling has not been widely studied, in terms of the performance of different norms or their scalability advantages. As far as we know, this is the first comprehensive study of regularization for topic modeling of text data.

We also show the relationships between RLSI and existing topic modeling techniques. From the viewpoint of optimization, RLSI and existing methods such as LSI, SC, and Non-negative Matrix Factorization (NMF) [Lee and Seung 1999; 2001] are algorithms that optimize different loss functions which can all be represented as specifications of a general loss function. RLSI does not have an explicit probabilistic formulation, like PLSI and LDA. However, we show that RLSI can be implicitly represented as a probabilistic model, like LSI, SC, and NMF.

Experimental results on a large web dataset show that 1) RLSI can scale up well and help improve relevance ranking accuracy. Specifically, we show that batch RLSI and online RLSI can efficiently run on *1.6 million documents and 7 million terms* on 16 distributed machines. In contrast, existing methods on parallelizing LDA were only able to work on far fewer documents and/or far fewer terms. Experiments on three TREC datasets show that 2) the readability of RLSI topics is equal to or better than the readability of those learned by LDA, PLSI, LSI, and NMF; 3) RLSI topics can be used in retrieval with better performance than LDA, PLSI, LSI, and NMF (sometimes statistically significant); 4) the best choice of regularization is ℓ_1 norm on topics and ℓ_2 norm on document representations in terms of topic readability and retrieval performance; 5) online RLSI can effectively capture the evolution of the topics and is useful for topic tracking.

Our main contributions in this paper are 1) we have first replaced the orthogonality constraint in LSI with ℓ_1 and/or ℓ_2 regularization, showing that the regularized LSI (RLSI) scales up better than existing topic modeling techniques such as LSI, PLSI, and LDA; 2) we have first examined the performance of different norms, showing that ℓ_1 norm on topics and ℓ_2 norm on document representations performs best. This paper is an extension of our previous conference paper [Wang et al. 2011]. Additional contributions of the paper include 1) the online RLSI algorithm is proposed and its theoretical properties are studied; 2) the capability of online RLSI on dynamic topic modeling is empirically verified; 3) a theoretical comparison of batch RLSI and online RLSI is given.

The rest of the paper is organized as follows. After a summary of related work in Section 2, we discuss the scalability problem of topic modeling on large scale text data in Section 3. In Section 4 and Section 5, we propose batch RLSI and online RLSI, two new approaches to scalable topic modeling, respectively. Their properties are discussed in Section 6. Section 7 introduces how to apply RLSI to relevance ranking and Section 8 presents the experimental results. Finally, we draw our conclusions in Section 9.

2. RELATED WORK

2.1. Topic Modeling

The goal of topic modeling is to automatically discover the hidden semantic structure of a document collection. Studies on topic modeling fall into two categories: probabilistic approaches and non-probabilistic approaches.

In the probabilistic approaches, a topic is defined as a probability distribution over a vocabulary and documents are defined as data generated from mixtures of topics. To generate a document, one chooses a distribution over topics. Then, for each term in that document, one chooses a topic according to the topic distribution, and draws a term from the topic according to its term distribution. PLSI [Hofmann 1999] and LDA [Blei et al. 2003] are two widely-used probabilistic approaches to topic modeling. One of the advantages of the probabilistic approaches is that the models can easily be extended. Many extensions of LDA have been developed. For a survey on the probabilistic topic models, please refer to [Blei 2011] and [Blei and Lafferty 2009].

In the non-probabilistic approaches, each document is represented as a vector of terms, and the term-document matrix is approximated as the product of a term-topic matrix and a topic-document matrix under some constraints. One interpretation of these approaches is to project the term vectors of documents (the term-document matrix) into a K-dimensional topic space in which each axis corresponds to a topic. LSI [Deerwester et al. 1990] is the best-known model. It decomposes the term-document matrix under the assumption that topic vectors are orthogonal and SVD is employed to solve the problem. NMF [Lee and Seung 1999; 2001] is an approach similar to LSI. In NMF, the term-document matrix is factorized under the constraint that all entries in the matrices are equal to or greater than zero. Sparse Coding (SC) [Lee et al. 2007; Olshausen and Fieldt 1997], which is used in computer vision and other related fields, is a technique similar to RLSI, but with ℓ_2 norm on the topics and ℓ_1 norm on the document representations.

It has been demonstrated that topic modeling is useful for knowledge discovery, relevance ranking in search, and document classification [Mimno and McCallum 2007; Wei and Croft 2006; Yi and Allan 2009; Lu et al. 2011]. In fact, topic modeling is becoming one of the important technologies in text mining, information retrieval, natural language processing, and other related fields.

One important issue of applying topic modeling to real-world problems is to scale up the algorithms to large document collections. Most efforts to improve topic modeling scalability have modified existing learning methods such as LDA. Newman, et al. proposed Approximate Distributed LDA (AD-LDA) [Newman et al. 2008], in which each processor performs a local Gibbs sampling followed by a global update. Two recent papers implemented AD-LDA as PLDA [Wang et al. 2009] and modified AD-LDA as PLDA+ [Liu et al. 2011], using MPI [Thakur and Rabenseifner 2005] and MapReduce [Dean et al. 2004]. In [Asuncion et al. 2011], the authors proposed purely asynchronous distributed LDA algorithms based on Gibbs sampling or Bayesian inference, called Async-CGB or Async-CVB, respectively. In Async-CGB and Async-CVB, each processor performs a local computation followed by a communication with other processors. In all the methods, the local processors need to maintain and update a dense term-topic matrix, usually in memory, which becomes a bottleneck for improving scalability. In [AlSumait et al. 2008; Hoffman et al. 2010; Mimno et al. 2012], online versions of stochastic LDA were proposed. For other related work, please refer to [Mimno and McCallum 2007; Smola and Narayanamurthy 2010; Yan et al. 2009].

In this paper, we propose a new topic modeling method which can scale up to large text corpora. The key ingredient of our method is to make the formulation of learning decomposable and thus make the process of learning parallelizable.

2.2. Regularization and Sparsity

Regularization is a common technique in machine learning to prevent over-fitting. Typical examples of regularization add a penalty based on the ℓ_1 or ℓ_2 norm of the model parameters.

Regularization via ℓ_2 norm uses the sum of squares of parameters and thus can make the model smooth and effectively deal with over-fitting. Regularization via ℓ_1 norm, on the other hand, uses the sum of absolute values of parameters and thus has the effect of causing many parameters to be zero and selecting a sparse model [Tibshirani 1996; Fu 1998; Osborne et al. 2000].

Sparse methods using ℓ_1 regularization, which aim to learn sparse representations (simple models) from the data, have received a lot of attention in machine learning, particularly in image processing (e.g., [Rubinstein et al. 2008]). Sparse Coding (SC) algorithms [Lee et al. 2007; Olshausen and Fieldt 1997], for example, are proposed to discover basis functions that capture high-level features in the data and find succinct representations of the data at the same time. Similar sparse mechanism has been observed in biological neurons of human brains, and thus SC is a plausible model of visual cortex as well. When SC is applied to natural images, the learned bases resemble the receptive fields of neurons in the visual cortex [Olshausen and Fieldt 1997].

In this paper we propose using sparse methods in topic modeling, particularly using ℓ_1 regularization to make the learned topics sparse. One notable advantage of making topics sparse is its ability of automatically selecting the most relevant terms for each topic. Moreover, sparsity leads to less memory usage for storing the topics. Such advantages make it an appealing choice for topic modeling. Wang and Blei [Wang and Blei 2009] suggested discovering sparse topics with a modified version of LDA, where a Bernoulli variable is introduced for each term-topic pair to determine whether or not the term appears in the topic. In [Shashanka et al. 2007], the authors adopted the PLSI framework and used an entropic prior in a Maximum A Posterior formulation to enforce sparsity. Two recent papers chose non-probabilistic formulations. One is based on LSI [Chen et al. 2010] and the other is based on a two-layer sparse coding model [Zhu and Xing 2011], which can directly control the sparsity of learned topics by using the sparsity-inducing ℓ_1 regularizer. However, none of these sparse topic models scales up well to large document collections. [Wang and Blei 2009] and [Shashanka et al. 2007] are based on the probabilistic topic models of LDA and PLSI respectively, whose scalability is limited due to the necessity of maintaining the probability distribution constraints. [Chen et al. 2010] is based on LSI, whose scalability is limited due to the orthogonality assumption. [Zhu and Xing 2011] learns a topic representation for each document as well as each term in the document, and thus the computational cost is high.

3. SCALABILITY OF TOPIC MODELS

One of the main challenges in topic modeling is to scale up to millions of documents or even more. As collection size increases, so does vocabulary size, rather than a maximum vocabulary being reached. For example, in the 1.6 million web documents in our experiment, there are more than 7 million unique terms even after pruning those with low frequency (e.g., with term frequency in the whole collection less than 2).

LSI needs to be solved by SVD due to the orthogonality assumption. The time complexity of computing SVD is normally $O(\min\{MN^2, NM^2\})$, where M denotes the number of rows of the input matrix and N denotes the number of columns. Thus, it appears to be very difficult to make LSI scalable and efficient.

For PLSI and LDA, it is necessary to maintain the probability distribution constraints of the term-topic matrix. When the matrix is large, there is a cost for maintaining the probabilistic framework. One possible solution is to reduce the number of terms, but the negative consequence is that it can sacrifice learning accuracy.

How to make existing topic modeling methods scalable is still a challenging problem. In this paper, we adopt a novel approach called RLSI, which can work equally well or even better than existing topic modeling methods, but is scalable by design. We propose two versions of RLSI: one is batch learning and the other online learning.

4. BATCH REGULARIZED LATENT SEMANTIC INDEXING

4.1. Problem Formulation

Suppose we are given a set of documents $\mathcal{D}$ with size N, containing terms from a vocabulary $\mathcal{V}$ with size M. A document is simply represented as an M-dimensional vector $\boldsymbol{d}$, where the m^{th} entry denotes the weight of the m^{th} term, for example, a Boolean value indicating occurrence, term frequency, tf-idf, or joint probability of the term and document. The N documents in $\mathcal{D}$ are then represented as an $M \times N$ term-document matrix $\mathbf{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_N]$, in which each row corresponds to a term and each column corresponds to a document.

A topic is defined over terms in the vocabulary and is also represented as an M-dimensional vector $\boldsymbol{u}$, where the m^{th} entry denotes the weight of the m^{th} term in the topic. Intuitively, the terms with larger weights are more indicative to the topic. Suppose that there are K topics in the collection. The K topics can be summarized into an $M \times K$ term-topic matrix $\mathbf{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K]$, in which each column corresponds to a topic.

Topic modeling means discovering the latent topics in the document collection as well as modeling the documents by representing them as mixtures of the topics. More precisely, given topics $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$, document $\boldsymbol{d}_n$ is succinctly represented as $\boldsymbol{d}_n \approx \sum_{k=1}^{K} v_{kn}\boldsymbol{u}_k = \mathbf{U}\boldsymbol{v}_n$, where v_{kn} denotes the weight of the k^{th} topic $\boldsymbol{u}_k$ in document $\boldsymbol{d}_n$. The larger value of v_{kn}, the more important role topic $\boldsymbol{u}_k$ plays in the document. Let $\mathbf{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$ be the topic-document matrix, where column $\boldsymbol{v}_n$ stands for the representation of document $\boldsymbol{d}_n$ in the latent topic space.Table I gives a summary of notations.

Different topic modeling techniques choose different schemas to model matrices $\mathbf{U}$ and $\mathbf{V}$ and impose different constraints on them. For example, in the generative topic models such as PLSI and LDA, topics $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$ are probability distributions so that $\sum_{m=1}^{M} u_{mk} = 1$ for $k = 1, \cdots, K$; document representations $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N$ are also probability distributions so that $\sum_{k=1}^{K} v_{kn} = 1$ for $n = 1, \cdots, N$. In LSI, topics $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$ are assumed to be orthogonal. Please note that in LSI, the input matrix $\mathbf{D}$ is approximated as $\mathbf{U\Sigma V}$, where $\mathbf{\Sigma}$ is a $K \times K$ diagonal matrix, as shown in Figure 1.

Regularized Latent Semantic Indexing (RLSI) learns latent topics as well as representations of documents from the given text collection. Document $\boldsymbol{d}_n$ is approximated as $\mathbf{U}\boldsymbol{v}_n$ where $\mathbf{U}$ is the term-topic matrix and $\boldsymbol{v}_n$ is the representation of $\boldsymbol{d}_n$ in the latent topic space. The goodness of the approximation is measured by the squared ℓ_2 norm of the difference between $\boldsymbol{d}_n$ and $\mathbf{U}\boldsymbol{v}_n$: $\|\boldsymbol{d}_n - \mathbf{U}\boldsymbol{v}_n\|_2^2$. Furthermore, topics and document representations are regularized. Specifically, we suggest ℓ_1 regularization on term-topic matrix $\mathbf{U}$ (i.e., topics $\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K$) and ℓ_2 on topic-document

Table I. Table of notations.

Notation	Meaning
M	Number of terms in vocabulary
N	Number of documents in collection
K	Number of topics
$\mathbf{D} \in \mathbb{R}^{M \times N}$	Term-document matrix $[\boldsymbol{d}_1, \cdots, \boldsymbol{d}_N]$
$\boldsymbol{d}_n$	The n^{th} document
d_{mn}	Weight of the m^{th} term in document $\boldsymbol{d}_n$
$\mathbf{U} \in \mathbb{R}^{M \times K}$	Term-topic matrix $[\boldsymbol{u}_1, \cdots, \boldsymbol{u}_K]$
$\boldsymbol{u}_k$	The k^{th} topic
u_{mk}	Weight of the m^{th} term in topic $\boldsymbol{u}_k$
$\mathbf{V} \in \mathbb{R}^{K \times N}$	Topic-document matrix $[\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$
$\boldsymbol{v}_n$	Representation of $\boldsymbol{d}_n$ in the topic space
v_{kn}	Weight of the k^{th} topic in $\boldsymbol{v}_n$

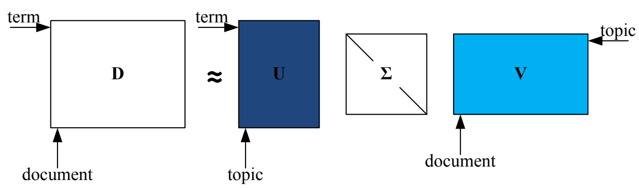


Fig. 1. LSI approximates the input tf-idf matrix $\mathbf{D}$ with $\mathbf{U\Sigma V}$.

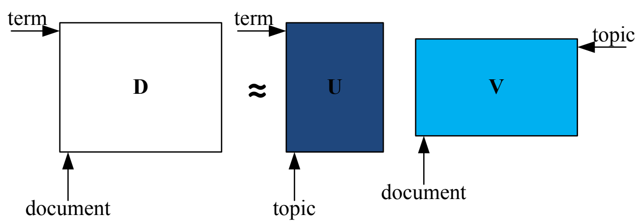


Fig. 2. Batch RLSI approximates the input tf-idf matrix $\mathbf{D}$ with $\mathbf{UV}$.

matrix $\mathbf{V}$ (i.e., document representations $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N$) to favor a model with compact and readable topics and useful for retrieval.

Thus, given a text collection $\mathcal{D} = \{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_N\}$, batch RLSI amounts to solving the following optimization problem:

$$\min_{\mathbf{U}, \{\boldsymbol{v}_n\}} \quad \sum_{n=1}^{N} \|\boldsymbol{d}_n - \mathbf{U}\boldsymbol{v}_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|\boldsymbol{v}_n\|_2^2, \tag{1}$$

where $\lambda_1 \geq 0$ is the parameter controlling the regularization on $\boldsymbol{u}_k$: the larger the value of λ_1, the more sparse is $\boldsymbol{u}_k$; and $\lambda_2 \geq 0$ is the parameter controlling the regularization on $\boldsymbol{v}_n$: the larger the value of λ_2, the larger amount of shrinkage on $\boldsymbol{v}_n$. From the viewpoint of matrix factorization, batch RLSI approximates the input term-document matrix $\mathbf{D}$ with the product of the term-topic matrix $\mathbf{U}$ and the topic-document matrix $\mathbf{V}$, as shown in Figure 2.

In general, the regularization on topics and document representations (the second term and the third term) can be either ℓ_1 norm or ℓ_2 norm. When they are ℓ_2 and ℓ_1 respectively, the method is equivalent to Sparse Coding [Lee et al. 2007; Olshausen and Fieldt 1997]. When both of them are ℓ_1, the model is similar to the double sparse model proposed in [Rubinstein et al. 2008][4].

[4]Note that both Sparse Coding and double sparse model formulate the optimization problems with constraints instead of regularization. The two formulations are equivalent.

Algorithm 1 Batch Regularized Latent Semantic Indexing

Require: $\mathbf{D} \in \mathbb{R}^{M \times N}$
$\mathbf{V}_0 \in \mathbb{R}^{K \times N} \leftarrow$ random matrix
for $t = 1 : T$ **do**
  $\mathbf{U}_t \leftarrow \text{Update}\mathbf{U}(\mathbf{D}, \mathbf{V}_{t-1})$
  $\mathbf{V}_t \leftarrow \text{Update}\mathbf{V}(\mathbf{D}, \mathbf{U}_t)$
end for
return $\mathbf{U}_T, \mathbf{V}_T$

4.2. Regularization Strategy

We propose using the formulation above (i.e., regularization via ℓ_1 norm on topics and ℓ_2 norm on document representations), because according to our experiments this regularization strategy leads to a model with more compact and readable topics and more effective for retrieval.

First, ℓ_1 norm on topics has the effect of making them compact. We do this under the assumption that the essence of a topic can be captured via a small number of terms, which is reasonable in practice. In many applications, small and concise topics are more useful. In learning and utilization of topic models, topic sparsity means that we can efficiently store and process topics. We can also leverage existing techniques on sparse matrix computation [Buluc and Gilbert 2008; Liu et al. 2010], which are efficient and scalable.

Second, ℓ_2 norm on document representations addresses the "term mismatch" problem better than ℓ_1 regularization when applied to relevance ranking. This is because when ℓ_1 regularization is imposed on $\mathbf{V}$, the document and query representations in the topic space will become sparse, and as a result the topic matching scores will not be reliable enough. In contrast, ℓ_2 regularization on $\mathbf{V}$ will make the document and query representations in the topic space "smooth", and thus matching in the topic space can be conducted more effectively.

We test all the four ways of combining ℓ_1 and ℓ_2 norms on topics and document representations on multiple datasets and find that best performance, in terms of topic readability and ranking accuracy, is achieved with ℓ_1 norm on topics and ℓ_2 norm on document representations.

4.3. Optimization

The optimization Eq. (1) is not jointly convex with respect to the two variables $\mathbf{U}$ and $\mathbf{V}$. However, it is convex with respect to one of them, when the other one is fixed. Following the practice in Sparse Coding [Lee et al. 2007], we optimize the function in Eq. (1) by alternately minimizing it with respect to term-topic matrix $\mathbf{U}$ and topic-document matrix $\mathbf{V}$. This procedure is summarized in Algorithm 1, which converges to a local minimum after a certain number of iterations (e.g., 100) according to our experiments. Note that for simplicity we describe the algorithm when ℓ_1 norm is imposed on topics and ℓ_2 norm on document representations. This can easily be extended to other regularization strategies.

4.3.1. Update of Matrix $\mathbf{U}$. Holding $\mathbf{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$ fixed, the update of $\mathbf{U}$ amounts to the following optimization problem:

$$\min_{\mathbf{U}} \quad \|\mathbf{D} - \mathbf{U}\mathbf{V}\|_F^2 + \lambda_1 \sum_{m=1}^{M} \sum_{k=1}^{K} |u_{mk}|,$$

where $\|\cdot\|_F$ is the Frobenius norm and u_{mk} is the $(mk)^{th}$ entry of $\mathbf{U}$. Let $\bar{\boldsymbol{d}}_m = (d_{m1}, \cdots, d_{mN})^T$ and $\bar{\boldsymbol{u}}_m = (u_{m1}, \cdots, u_{mK})^T$ be the column vectors whose entries are those of the m^{th} row of $\mathbf{D}$ and $\mathbf{U}$ respectively. Thus, the previous optimization problem can be rewritten as

$$\min_{\{\bar{\boldsymbol{u}}_m\}} \quad \sum_{m=1}^{M} \left\|\bar{\boldsymbol{d}}_m - \mathbf{V}^T \bar{\boldsymbol{u}}_m\right\|_2^2 + \lambda_1 \sum_{m=1}^{M} \|\bar{\boldsymbol{u}}_m\|_1,$$

Algorithm 2 Update**U**

Require: $\mathbf{D} \in \mathbb{R}^{M\times N}$, $\mathbf{V} \in \mathbb{R}^{K\times N}$

$\mathbf{S} \leftarrow \mathbf{V}\mathbf{V}^T$
$\mathbf{R} \leftarrow \mathbf{D}\mathbf{V}^T$
for $m = 1 : M$ **do**
  $\bar{\boldsymbol{u}}_m \leftarrow \mathbf{0}$
  repeat
    for $k = 1 : K$ **do**
      $w_{mk} \leftarrow r_{mk} - \sum_{l\neq k} s_{kl}u_{ml}$
      $u_{mk} \leftarrow \frac{\left(|w_{mk}| - \frac{1}{2}\lambda_1\right)_+ \mathrm{sign}(w_{mk})}{s_{kk}}$
    end for
  until convergence
end for
return U

which can be decomposed into M optimization problems that can be solved independently, with each corresponding to one row of $\mathbf{U}$:

$$\min_{\bar{\boldsymbol{u}}_m} \quad \left\|\bar{\boldsymbol{d}}_m - \mathbf{V}^T\bar{\boldsymbol{u}}_m\right\|_2^2 + \lambda_1 \|\bar{\boldsymbol{u}}_m\|_1 , \tag{2}$$

for $m = 1, \cdots, M$.

Eq. (2) is an ℓ_1-regularized least squares problem, whose objective function is not differentiable and it is not possible to directly apply gradient-based methods. A number of techniques can be used here, such as interior point methods [Chen et al. 1998], coordinate descent with soft-thresholding [Friedman et al. 2007; Fu 1998], Lars-Lasso algorithm [Efron et al. 2004; Osborne et al. 2000], and feature-sign search [Lee et al. 2007]. Here we choose coordinate descent with soft-thresholding, which is an iterative algorithm that applies soft-thresholding with one entry of the parameter vector (i.e., $\bar{\boldsymbol{u}}_m$) repeatedly until convergence[5]. At each iteration, we take u_{mk} as the variable, and minimize the objective function in Eq. (2) with respect to u_{mk} while keeping all the u_{ml} fixed for which $l \neq k$, $k = 1, \cdots, K$.

Let $\bar{\boldsymbol{v}}_k = (v_{k1}, \cdots, v_{kN})^T$ be the column vector whose entries are those of the k^{th} row of $\mathbf{V}$, $\mathbf{V}^T{}_{\backslash k}$ the matrix of $\mathbf{V}^T$ with the k^{th} column removed, and $\bar{\boldsymbol{u}}_{m\backslash k}$ the vector of $\bar{\boldsymbol{u}}_m$ with the k^{th} entry removed, and we can rewrite the objective in Eq. (2) as a function with respect to u_{mk}:

$$\begin{aligned} L(u_{mk}) &= \left\|\bar{\boldsymbol{d}}_m - \mathbf{V}^T{}_{\backslash k}\bar{\boldsymbol{u}}_{m\backslash k} - u_{mk}\bar{\boldsymbol{v}}_k\right\|_2^2 + \lambda_1 \left\|\bar{\boldsymbol{u}}_{m\backslash k}\right\|_1 + \lambda_1 |u_{mk}| \\ &= \|\bar{\boldsymbol{v}}_k\|_2^2\, u_{mk}^2 - 2\left(\bar{\boldsymbol{d}}_m - \mathbf{V}^T{}_{\backslash k}\bar{\boldsymbol{u}}_{m\backslash k}\right)^T \bar{\boldsymbol{v}}_k u_{mk} + \lambda_1 |u_{mk}| + const \\ &= s_{kk}u_{mk}^2 - 2\left(r_{mk} - \sum_{l\neq k} s_{kl}u_{ml}\right) u_{mk} + \lambda_1 |u_{mk}| + const, \end{aligned}$$

where s_{ij} and r_{ij} are the $(ij)^{th}$ entries of $K \times K$ matrix $\mathbf{S} = \mathbf{V}\mathbf{V}^T$ and $M \times K$ matrix $\mathbf{R} = \mathbf{D}\mathbf{V}^T$, respectively, and $const$ is a constant with respect to u_{mk}. According to Lemma A.1 in Appendix (i.e., Eq. (10)), the optimal u_{mk} is

$$u_{mk} = \frac{\left(\left|r_{mk} - \sum_{l\neq k} s_{kl}u_{ml}\right| - \frac{1}{2}\lambda_1\right)_+ \mathrm{sign}\left(r_{mk} - \sum_{l\neq k} s_{kl}u_{ml}\right)}{s_{kk}},$$

where $(\cdot)_+$ denotes the hinge function. The algorithm for updating $\mathbf{U}$ is summarized in Algorithm 2.

[5]The convergence of coordinate descent with soft-thresholding is shown in [Friedman et al. 2007].

Algorithm 3 Update$\mathbf{V}$

Require: $\mathbf{D} \in \mathbb{R}^{M \times N}$, $\mathbf{U} \in \mathbb{R}^{M \times K}$
1: $\mathbf{\Sigma} \leftarrow \left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-1}$
2: $\mathbf{\Phi} \leftarrow \mathbf{U}^T\mathbf{D}$
3: **for** $n = 1 : N$ **do**
4: $\quad \boldsymbol{v}_n \leftarrow \mathbf{\Sigma}\boldsymbol{\phi}_n$, where $\boldsymbol{\phi}_n$ is the n^{th} column of $\mathbf{\Phi}$
5: **end for**
6: **return** $\mathbf{V}$

4.3.2. Update of Matrix $\mathbf{V}$. The update of $\mathbf{V}$ with $\mathbf{U}$ fixed is a least squares problem with ℓ_2 regularization. It can also be decomposed into N optimization problems, with each corresponding to one $\boldsymbol{v}_n$ and can be solved in parallel:

$$\min_{\boldsymbol{v}_n} \quad \|\boldsymbol{d}_n - \mathbf{U}\boldsymbol{v}_n\|_2^2 + \lambda_2 \|\boldsymbol{v}_n\|_2^2 , \tag{3}$$

for $n = 1, \cdots, N$. It is a standard ℓ_2-regularized least squares problem (also known as Ridge Regression in statistics) and the solution is:

$$\boldsymbol{v}_n^* = \left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-1} \mathbf{U}^T\boldsymbol{d}_n.$$

Algorithm 3 shows the procedure[6].

4.4. Distributed RLSI

The formulation of batch RLSI makes it possible to decompose the learning problem into multiple sub-optimization problems and conduct learning in parallel or distributed manner. Specifically, for both the term-topic matrix and the topic-document matrix, the update in each iteration is decomposed into many sub-optimization problems that can be solved in parallel, for example via MapReduce [Dean et al. 2004], which makes batch RLSI scalable.

MapReduce is a computing model that supports distributed computing on large datasets. MapReduce expresses a computing task as a series of Map and Reduce operations and performs the task by executing the operations in a distributed computing environment. In this section, we describe the implementation of batch RLSI on MapReduce, referred to as distributed RLSI, as shown in Figure 3[7]. At each iteration the algorithm updates $\mathbf{U}$ and $\mathbf{V}$ using the following MapReduce operations:

Map-1. Broadcast $\mathbf{S} = \mathbf{V}\mathbf{V}^T$ and map $\mathbf{R} = \mathbf{D}\mathbf{V}^T$ on m ($m = 1, \cdots, M$) such that all of the entries in the m^{th} row of $\mathbf{R}$ are shuffled to the same machine in the form of $\langle m, \bar{\boldsymbol{r}}_m, \mathbf{S}\rangle$, where $\bar{\boldsymbol{r}}_m$ is the column vector whose entries are those of the m^{th} row of $\mathbf{R}$.

Reduce-1. Take $\langle m, \bar{\boldsymbol{r}}_m, \mathbf{S}\rangle$ and emit $\langle m, \bar{\boldsymbol{u}}_m\rangle$, where $\bar{\boldsymbol{u}}_m$ is the optimal solution for the m^{th} optimization problem (Eq. (2)). We have $\mathbf{U} = [\bar{\boldsymbol{u}}_1, \cdots, \bar{\boldsymbol{u}}_M]^T$.

Map-2. Broadcast $\mathbf{\Sigma} = \left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-1}$ and map $\mathbf{\Phi} = \mathbf{U}^T\mathbf{D}$ on n ($n = 1, \cdots, N$) such that the entries in the n^{th} column of $\mathbf{\Phi}$ are shuffled to the same machine in the form of $\langle n, \boldsymbol{\phi}_n, \mathbf{\Sigma}\rangle$, where $\boldsymbol{\phi}_n$ is the n^{th} column of $\mathbf{\Phi}$.

Reduce-2. Take $\langle n, \boldsymbol{\phi}_n, \mathbf{\Sigma}\rangle$ and emit $\langle n, \boldsymbol{v}_n = \mathbf{\Sigma}\boldsymbol{\phi}_n\rangle$. We have $\mathbf{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N]$.

Note that the data partitioning schemas for $\mathbf{R}$ in Map-1 and for $\mathbf{\Phi}$ in Map-2 are different. $\mathbf{R}$ is split such that entries in the same row (corresponding to one term) are shuffled to the same machine while $\mathbf{\Phi}$ is split such that entries in the same column (corresponding to one document) are shuffled to the same machine.

[6]If K is large such that the matrix inversion $\left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-1}$ is hard, we can employ gradient descent in the update of $\boldsymbol{v}_n$.

[7]Here we only discuss the parallelization for RLSI in the batch mode; in principle the technique can also be applied to the online mode.

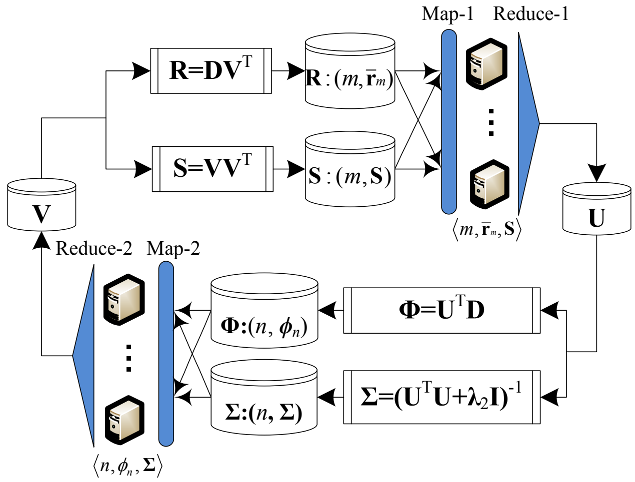


Fig. 3. Update of **U** and **V** on MapReduce.

There are a number of large scale matrix multiplication operations in operation Map-1 ($\mathbf{DV}^T$ and $\mathbf{VV}^T$) and Map-2 ($\mathbf{U}^T\mathbf{D}$ and $\mathbf{U}^T\mathbf{U}$). These matrix multiplication operations can also be conducted on MapReduce infrastructure efficiently. As example, $\mathbf{DV}^T$ can be calculated as $\sum_{n=1}^{N} \boldsymbol{d}_n \boldsymbol{v}_n^T$ and thus fully parallelized. For details please refer to [Buluc and Gilbert 2008; Liu et al. 2010].

5. ONLINE REGULARIZED LATENT SEMANTIC INDEXING

In many applications, documents are provided in a data stream, and the topics covered in newer documents may differ from those in older documents. Examples of such data streams are journal articles, email messages, news articles, and queries from search logs. In this setting, we want to sequentially construct the topic model from documents, and learn the dynamics of topics over time. Dynamic topic modeling techniques have been proposed based on the same motivation and have been successfully applied to real-world applications [Allan et al. 1998; Blei and Lafferty 2006; Wang and McCallum 2006].

In this section, we consider online RLSI, which incrementally builds a topic model on the basis of the stream data and captures the evolution of the topics. As shown in the experiments, online RLSI is effective for topic tracking. Online RLSI has a similar formulation as batch RLSI. Hereafter, we consider the formulation using ℓ_1 norm regularization on topics and ℓ_2 norm regularization on document representations. This regularization strategy leads to a model with high topic readability and effectiveness for retrieval, as discussed in Section 4.2.

5.1. Formulation

Suppose that we are given a set of documents $\mathcal{D}$ with size N, in batch RLSI the regularized loss function Eq. (1) is optimized. Equivalently, Eq. (1) can be written as:

$$\min_{\mathbf{U},\{\boldsymbol{v}_n\}} \quad \frac{1}{N}\sum_{n=1}^{N}\left[\|\boldsymbol{d}_n - \mathbf{U}\boldsymbol{v}_n\|_2^2 + \lambda_2 \|\boldsymbol{v}_n\|_2^2\right] + \theta \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_1 \tag{4}$$

by dividing the objective function by N, where the first term stands for the“empirical loss” for the N documents, the second term controls the model complexity, and $\theta = \lambda_1/N$ is a trade-off parameter.

In online RLSI, the documents are assumed to be i.i.d. data drawn one by one from the distribution of documents. The algorithm takes one document $\boldsymbol{d}_t$ at a time, projects the document in the topic space, and updates the term-topic matrix.

The projection $\boldsymbol{v}_t$ of document $\boldsymbol{d}_t$ in the topic space is obtained by solving

$$\min_{\boldsymbol{v}} \quad \|\boldsymbol{d}_t - \mathbf{U}_{t-1}\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2 \,, \tag{5}$$

where $\mathbf{U}_{t-1}$ is the term-topic matrix obtained at the previous iteration.

The new term-topic matrix $\mathbf{U}_t$ is obtained by solving

$$\min_{\mathbf{U}} \quad \hat{f}_t(\mathbf{U}) \triangleq \frac{1}{t}\sum_{i=1}^{t}\left[\|\boldsymbol{d}_i - \mathbf{U}\boldsymbol{v}_i\|_2^2 + \lambda_2 \|\boldsymbol{v}_i\|_2^2\right] + \theta \sum_{k=1}^{K} \|\boldsymbol{u}_k\|_1 \,, \tag{6}$$

where $\boldsymbol{v}_i$ (for $i \leq t$) are cumulated in the previous iterations.

The rationale behind online RLSI is as follows. First, it is a stochastic approximation of batch RLSI. At time t, the optimization problem Eq. (5) is an approximation of Eq. (3), and the loss $\hat{f}_t$ defined in Eq. (6) is also an approximation of Eq. (4). Second, both $\boldsymbol{v}_t$ and $\mathbf{U}_t$ are obtained with the information in the previous iterations, namely term-topic matrix $\mathbf{U}_{t-1}$ and document representations $\boldsymbol{v}_i$ for $i \leq t$. Last, the term-topic matrices $\{\mathbf{U}_t\}$ form a time series and thus can capture the evolution of topics.

5.2. Optimization

The optimization in online RLSI can be performed in a similar way as in batch RLSI.

5.2.1. Document Projection. The document projection (Eq. (5)) can be solved as a standard ℓ_2-regularized least squares problem and the solution is:

$$\boldsymbol{v}_t = \left(\mathbf{U}_{t-1}^T\mathbf{U}_{t-1} + \lambda_2\mathbf{I}\right)^{-1}\mathbf{U}_{t-1}^T\boldsymbol{d}_t.$$

5.2.2. Term-topic Matrix Update. The update (Eq. (6)) is equivalent to

$$\min_{\mathbf{U}} \quad \|\mathbf{D}_t - \mathbf{U}\mathbf{V}_t\|_F^2 + \theta t \sum_{m=1}^{M}\sum_{k=1}^{K} |u_{mk}| \,,$$

where $\mathbf{D}_t = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_t]$ and $\mathbf{V}_t = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_t]$ are the term-document matrix and topic-document matrix until time t respectively. Using the techniques described in Section 4.3, we decompose the optimization problem into M subproblems with each corresponding to one row of $\mathbf{U}$:

$$\min_{\bar{\boldsymbol{u}}_m} \quad \left\|\bar{\boldsymbol{d}}_m^{(t)} - \mathbf{V}_t^T\bar{\boldsymbol{u}}_m\right\|_2^2 + \theta t \|\bar{\boldsymbol{u}}_m\|_1 \,, \tag{7}$$

for $m = 1, \cdots, M$. Here $\bar{\boldsymbol{u}}_m = (u_{m1}, \cdots, u_{mK})^T$ and $\bar{\boldsymbol{d}}_m^{(t)} = (d_{m1}, \cdots, d_{mt})^T$ are the column vectors whose entries are those of the m^{th} row of $\mathbf{U}$ and $\mathbf{D}_t$ respectively.

The minimum of Eq. (7) can be obtained with the technique presented in Algorithm 2, by setting $\mathbf{S} = \mathbf{S}_t$, $\mathbf{R} = \mathbf{R}_t$, and $\lambda_1 = \theta t$. In online RLSI, $\mathbf{S}_t = \mathbf{V}_t\mathbf{V}_t^T = \sum_{i=1}^{t}\boldsymbol{v}_i\boldsymbol{v}_i^T$ and $\mathbf{R}_t = \mathbf{D}_t\mathbf{V}_t^T = \sum_{i=1}^{t}\boldsymbol{d}_i\boldsymbol{v}_i^T$ can be calculated efficiently in an additive manner:

$$\mathbf{S}_t = \begin{cases} \mathbf{S}_{t-1} + \boldsymbol{v}_t\boldsymbol{v}_t^T, & t \geq 1, \\ \mathbf{0}, & t = 0, \end{cases}$$

and

$$\mathbf{R}_t = \begin{cases} \mathbf{R}_{t-1} + \boldsymbol{d}_t\boldsymbol{v}_t^T, & t \geq 1, \\ \mathbf{0}, & t = 0. \end{cases}$$

Algorithm 4 shows the details of the online RLSI algorithm.

Algorithm 4 Online Regularized Latent Semantic Indexing

Require: $p(\boldsymbol{d})$
$\mathbf{U}_0 \in \mathbb{R}^{M \times K} \leftarrow$ (random matrix or previously learned term-topic matrix)
$\mathbf{S}_0 \in \mathbb{R}^{K \times K} \leftarrow \mathbf{0}$
$\mathbf{R}_0 \in \mathbb{R}^{M \times K} \leftarrow \mathbf{0}$
for $t = 1 : T$ **do**
  Draw $\boldsymbol{d}_t$ from $p(\boldsymbol{d})$
  $\boldsymbol{v}_t \leftarrow \left(\mathbf{U}_{t-1}^T \mathbf{U}_{t-1} + \lambda_2 \mathbf{I}\right)^{-1} \mathbf{U}_{t-1}^T \boldsymbol{d}_t$
  $\mathbf{S}_t \leftarrow \mathbf{S}_{t-1} + \boldsymbol{v}_t \boldsymbol{v}_t^T$
  $\mathbf{R}_t \leftarrow \mathbf{R}_{t-1} + \boldsymbol{d}_t \boldsymbol{v}_t^T$
  $\mathbf{U}_t \leftarrow$ Updated by Algorithm 2 with $\mathbf{S} = \mathbf{S}_t$, $\mathbf{R} = \mathbf{R}_t$, and $\lambda_1 = \theta t$
end for
return $\mathbf{U}_T$

5.3. Convergence Analysis

We prove that the term-topic matrix series $\{\mathbf{U}_t\}$ generated by online RLSI satisfies $\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F = O\left(\frac{1}{t}\right)$, which means that the convergence of the positive sum $\sum_{t=1}^{\infty} \|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F^2$ is guaranteed, although there is no guarantee on the convergence of $\mathbf{U}_t$ itself. This is a property often observed in gradient descent methods [Bertsekas 1999]. Our proof is inspired by the theoretical analysis in [Mairal et al. 2010] on the Lipschitz regularity of solutions to optimization problems [Bonnans and Shapiro 1998].

We first give the assumptions necessary for the analysis, which are reasonable and natural.

Assumption 5.1. *The document collection $\mathcal{D}$ is composed of i.i.d. samples of a distribution of documents $p(\boldsymbol{d})$ with compact support $\mathcal{K} = \left\{\boldsymbol{d} \in \mathbb{R}^M : \|\boldsymbol{d}\|_2 \le \delta_1\right\}$.* The compact support assumption is common in text, image, audio, and video processing.

Assumption 5.2. *The solution to the problem of minimizing $\hat{f}_t$ lies in a bounded convex subset $\mathcal{U} = \left\{\mathbf{U} \in \mathbb{R}^{M \times K} : \|\mathbf{U}\|_F \le \delta_2\right\}$ for every t.* Since $\hat{f}_t$ is convex with respect to $\mathbf{U}$, the set of all possible minima is convex. The bound assumption is also quite natural, especially when the minima are obtained by some specific algorithms such as LARS [Efron et al. 2004] and coordinate descent with soft-thresholding [Fu 1998] which we employ in this paper.

Assumption 5.3. *Starting at any initial point, the optimization problem Eq. (7) reaches a local minimum after at most T rounds of iterative minimization.* Here iterative minimization means minimizing the objective function with respect to one entry of $\bar{\boldsymbol{u}}_m$ while the others are fixed. Note that the achieved local minimum is also global since Eq. (7) is a convex optimization problem.

Assumption 5.4. *The smallest diagonal entry of the positive semi-definite matrix $\frac{1}{t}\mathbf{S}_t$ defined in Algorithm 4 is lager than or equal to some constant $\kappa_1 > 0$.* Note that $\frac{1}{t}\mathbf{S}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{v}_i \boldsymbol{v}_i^T$, whose diagonal entries are $\frac{1}{t}\sum_{i=1}^{t} v_{1i}^2, \cdots, \frac{1}{t}\sum_{i=1}^{t} v_{Ki}^2$, where v_{ki} is the k^{th} entry of $\boldsymbol{v}_i$ for $k = 1, \cdots, K$. This hypothesis is experimentally verified to be true after a small number of iterations given that the initial term-topic matrix $\mathbf{U}_0$ is learned in the previous round or is set randomly.

Given Assumption 5.1 - Assumption 5.4, we can obtain the result as follows, whose proof can be found in Appendix.

Proposition 5.5. *Let $\mathbf{U}_t$ be the solution to Eq. (6). Under Assumptions 5.1 - 5.4, the following inequality holds almost surely for all t:*

$$\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F \le \frac{T}{(t+1)\kappa_1} \left(\frac{\delta_1^2 \delta_2}{\lambda_2} + \frac{2\delta_1^2}{\sqrt{\lambda_2}} \right). \tag{8}$$

5.4. Algorithm Improvements

We have presented the basic version of online RLSI and proved a convergence property of it. This section discusses several simple improvements that significantly enhance the performance of basic online RLSI. Note that the convergence analysis in Section 5.3 can be easily extended to the improved versions.

5.4.1. Re-scaling. In Algorithm 4 (line 7 and line 8), at each iteration, the "new" information (i.e., $\boldsymbol{v}_t\boldsymbol{v}_t^T$ and $\boldsymbol{d}_t\boldsymbol{v}_t^T$) added to the matrices $\mathbf{S}_t$ and $\mathbf{R}_t$ has the same weight as the "old" information (i.e., $\mathbf{S}_{t-1}$ and $\mathbf{R}_{t-1}$). One modification is to re-scale the old information so that the new information has higher weight [Neal and Hinton 1998; Mairal et al. 2010]. We can follow the idea in [Mairal et al. 2010] and replace line 7 and line 8 in Algorithm 4 by

$$\mathbf{S}_t \leftarrow \left(\frac{t-1}{t}\right)^{\rho} \mathbf{S}_{t-1} + \boldsymbol{v}_t\boldsymbol{v}_t^T,$$
$$\mathbf{R}_t \leftarrow \left(\frac{t-1}{t}\right)^{\rho} \mathbf{R}_{t-1} + \boldsymbol{d}_t\boldsymbol{v}_t^T,$$

where ρ is a parameter. When $\rho = 0$, we obtain the basic version of online RLSI.

5.4.2. Mini-batch. Mini-batch is a typical heuristic adopted in stochastic learning, which processes multiple data instances in each iteration to reduce noise and speed up convergence [Bottou and Bousquet 2008; Liang and Klein 2009; Hoffman et al. 2010; Mairal et al. 2010]. We can enhance the performance of online RLSI through the mini-batch extension, i.e., processing $\eta \geq 1$ documents at each iteration instead of a single document. Let $\boldsymbol{d}_{t,1}, \cdots, \boldsymbol{d}_{t,\eta}$ denote the documents drawn at iteration t and $\boldsymbol{v}_{t,1}, \cdots, \boldsymbol{v}_{t,\eta}$ denote their representations in the topic space, which can be obtained by the techniques described in Section 5.2. Line 7 and line 8 in Algorithm 4 can then be replaced by

$$\mathbf{S}_t \leftarrow \mathbf{S}_{t-1} + \sum_{i=1}^{\eta} \boldsymbol{v}_{t,i}\boldsymbol{v}_{t,i}^T,$$
$$\mathbf{R}_t \leftarrow \mathbf{R}_{t-1} + \sum_{i=1}^{\eta} \boldsymbol{d}_{t,i}\boldsymbol{v}_{t,i}^T.$$

When $\eta = 1$, we obtain the basic version of online RLSI.

5.4.3. Embedded Iterations. As shown in Algorithm 4 (line 9), the term-topic matrix is updated by Algorithm 2 once per iteration. At each iteration t, no matter what the start point (i.e., $\mathbf{U}_{t-1}$) is, Algorithm 2 forces the term-topic matrix (i.e., $\mathbf{U}_t$) to be zero, before updating it (line 4 in Algorithm 2), which leads to a large deviation in $\mathbf{U}_t$ from the start point $\mathbf{U}_{t-1}$. To deal with this problem, we iterate lines 6-9 in Algorithm 4 for $\xi \geq 1$ times. In practice, such embedded iterations are useful for generating stable term-topic matrix series $\{\mathbf{U}_t\}$. When $\xi = 1$, we obtain the basic version of online RLSI.

6. DISCUSSIONS

We discuss the properties of batch RLSI, online RLSI, and distributed RLSI, with ℓ_1 norm on topics and ℓ_2 norm on document representations as example.

6.1. Relationship with Other Methods

Batch RLSI is closely related to existing topic modeling methods such as LSI, PLSI, NMF and SC. In [Singh and Gordon 2008], the relationship between LSI and PLSI is discussed, from the view point of loss function and regularization. We borrow their framework, and show the relations between batch RLSI and the existing approaches. In the framework, topic modeling is considered

Table II. Optimization framework for different topic modeling methods.

Method	$\mathcal{B}(\mathbf{D}\|\mathbf{UV})$	$\mathcal{R}(\mathbf{U},\mathbf{V})$	Constraint on $\mathbf{U}$	Constraint on $\mathbf{V}$
LSI	$\|\mathbf{D}-\mathbf{UV}\|_F^2$	—	$\mathbf{U}^T\mathbf{U}=\mathbf{I}$	$\mathbf{VV}^T=\mathbf{\Lambda}^2$ ($\mathbf{\Lambda}$ is diagonal)
PLSI	$\sum_{mn}\left(d_{mn}\log\frac{d_{mn}}{(\mathbf{UV})_{mn}}\right)$	—	$\mathbf{U}^T\mathbf{1}=\mathbf{1}$, $u_{mk}\geq 0$	$\mathbf{1}^T\mathbf{V}\mathbf{1}=1$, $v_{kn}\geq 0$
NMF	$\|\mathbf{D}-\mathbf{UV}\|_F^2$	—	$u_{mk}\geq 0$	$v_{kn}\geq 0$
SC	$\|\mathbf{D}-\mathbf{UV}\|_F^2$	$\sum_n\|\boldsymbol{v}_n\|_1$	$\|\boldsymbol{u}_k\|_2^2\leq 1$	—
Batch RLSI	$\|\mathbf{D}-\mathbf{UV}\|_F^2$	$\sum_k\|\boldsymbol{u}_k\|_1$, $\sum_n\|\boldsymbol{v}_n\|_2^2$	—	—

Table III. Priors/constraints in different non-probabilistic methods.

Method	Prior/Constraint on $\boldsymbol{u}_k$	Prior/Constraint on $\boldsymbol{v}_n$
LSI	orthonormality	orthogonality
NMF	$u_{mk}\geq 0$	$v_{kn}\geq 0$
SC	$\|\boldsymbol{u}_k\|_2^2\leq 1$	$p(\boldsymbol{v}_n)\propto\exp(-\lambda\|\boldsymbol{v}_n\|_1)$
Batch RLSI	$p(\boldsymbol{u}_k)\propto\exp(-\lambda_1\|\boldsymbol{u}_k\|_1)$	$p(\boldsymbol{v}_n)\propto\exp\left(-\lambda_2\|\boldsymbol{v}_n\|_2^2\right)$

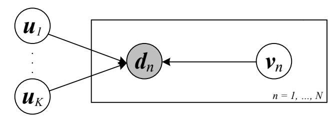


Fig. 4. Probabilistic framework for non-probabilistic methods.

as a problem of optimizing the following general loss function

$$\min_{(\mathbf{U},\mathbf{V})\in C}\quad \mathcal{B}(\mathbf{D}\|\mathbf{UV})+\lambda\mathcal{R}(\mathbf{U},\mathbf{V}),$$

where $\mathcal{B}(\cdot\|\cdot)$ is generalized Bregman divergence with non-negative values and is equal to zero if and only if the two inputs are equivalent; $\mathcal{R}(\cdot,\cdot)\geq 0$ is the regularization on the two inputs; C is the solution space; and λ is a coefficient making trade-off between the divergence and regularization.

Different choices of $\mathcal{B}$, $\mathcal{R}$, and C lead to different topic modeling techniques. Table II shows the relationship between batch RLSI and LSI, PLSI, NMF, and SC. (Suppose that we first conduct normalization $\sum_{m,n}d_{mn}=1$ in PLSI [Ding et al. 2008].) Within this framework, the major question becomes how to conduct regularization as well as optimization to make the learned topics readable.

6.2. Probabilistic and Non-probabilistic Models

Many non-probabilistic topic modeling techniques, such as LSI, NMF, SC, and batch RLSI can be interpreted within a probabilistic framework, as shown in Figure 4.

In the probabilistic framework, columns of the term-topic matrix $\boldsymbol{u}_k$'s are assumed to be independent from each other and columns of the topic-document matrix $\boldsymbol{v}_n$'s are regarded as latent variables. Next, each document $\boldsymbol{d}_n$ is assumed to be generated according to a Gaussian distribution conditioned on $\mathbf{U}$ and $\boldsymbol{v}_n$, i.e., $p(\boldsymbol{d}_n|\mathbf{U},\boldsymbol{v}_n)\propto\exp\left(-\|\boldsymbol{d}_n-\mathbf{U}\boldsymbol{v}_n\|_2^2\right)$. Furthermore, all the pairs $(\boldsymbol{d}_n,\boldsymbol{v}_n)$ are conditionally independent given $\mathbf{U}$.

Different techniques use different priors or constraints on $\boldsymbol{u}_k$'s and $\boldsymbol{v}_n$'s. Table III lists the priors or constraints used in LSI, NMF, SC, and batch RLSI, respectively. It can be shown that LSI, NMF, SC, and batch RLSI can be obtained with Maximum A Posteriori (MAP) Estimation [Mairal et al. 2009]. That is to say, the techniques can be understood in the same framework. In [Ding 2005], the authors propose a probabilistic framework based on document-document and word-word similarities to give an interpretation to LSI, which is very different from the framework here.

6.3. Batch RLSI vs. Online RLSI

Online RLSI is designed for online learning setting. The advantage is that it does not need to use so much storage (memory), while the disadvantage is that it usually requires higher total computation cost. Table IV compares the space and time complexity of batch RLSI and online RLSI, where AvgDL is the average document length in the collection, γ is the sparsity of topics, and T_o and T_i are respectively the numbers of outer and inner iterations in Algorithm 1 and Algorithm 4.

The space complexity of batch RLSI is $\gamma KM+(\text{AvgDL}\times N+KN)+\max\left\{K^2+KM,K^2+KN\right\}$, where the first term is for storing $\mathbf{U}$, the second term is for storing $\mathbf{D}$ and $\mathbf{V}$, and the third term is for storing $\mathbf{S}$ and $\mathbf{R}$ when updating $\mathbf{U}$, or storing $\mathbf{\Sigma}$ and $\mathbf{\Phi}$ when updating $\mathbf{V}$. Online RLSI processes one document at a time, and thus we only need to keep in memory one document as well as its

Table IV. Space and time complexity of batch RLSI and online RLSI.

Method	Space complexity	Time complexity
Batch RLSI	$\gamma KM + (\text{AvgDL} \times N + KN) + \max\left\{K^2 + KM, K^2 + KN\right\}$	$O\left(T_o \max\left\{NK^2, \text{AvgDL} \times NK, T_i MK^2\right\}\right)$
Online RLSI	$\gamma KM + (\text{AvgDL} + K) + \left(K^2 + KM\right)$	$O\left(T_o T_i MK^2\right)$

representation in the topic space. Thus the second term reduces to AvgDL + K for online RLSI. This is why we say that online RLSI has better scalability than batch RLSI.

We also compare the time complexity of batch RLSI and online RLSI. For batch RLSI, in each outer iteration, the time for updating $\mathbf{U}$ (i.e., Algorithm 2) dominates, and thus its time complexity is of order $T_o \max\left\{NK^2, \text{AvgDL} \times NK, T_i MK^2\right\}$, where NK^2 is for computing $\mathbf{S}$, AvgDL $\times NK$ is for computing $\mathbf{R}$, and $T_i MK^2$ is for running the inner iterations in each outer iteration. For online RLSI, in the processing of each document, the time for updating $\mathbf{U}$ (i.e., line 9 in Algorithm 4) dominates, and thus its time complexity is of order $T_o T_i MK^2$. In practice, the vocabulary size M is usually larger than the document collection size N, and thus $\max\left\{NK^2, \text{AvgDL} \times NK, T_i MK^2\right\} = T_i MK^2$ holds with some properly chosen K and T_i. Even in that case, online RLSI has higher total time complexity than batch RLSI since the number of outer iterations in Algorithm 4 (i.e., total number of documents) is usually larger than that in Algorithm 1 (i.e., fixed to 100).

The main reason that online RLSI has even higher time complexity than batch RLSI is that stochastic learning can only perform efficient learning of document representations (topic-document matrix $\mathbf{V}$) but not learning of topics (term-topic matrix $\mathbf{U}$), which dominates the total computation cost. Nonetheless, online RLSI is still superior to batch RLSI when processing stream data.

6.4. Scalability of Distributed RLSI

As explained, several methods for improving the efficiency and scalability of existing topic models, especially LDA, have been proposed. Table V shows the space and time complexities of AD-LDA [Newman et al. 2008], Async-CBS, Async-CVB [Asuncion et al. 2011], and distributed RLSI, where AvgDL is the average document length in the collection and γ is the sparsity of topics.

The space complexity of AD-LDA (also Async-CGS and Async-CVB) is of order $\frac{N \times \text{AvgDL} + NK}{P} + MK$, where MK is for storing the term-topic matrix on each processor. For a large text collection, the vocabulary size M will be very large and thus the space complexity will be very high. This will hinder it from being applied to large datasets in real-world applications.

The space complexity of distributed RLSI is $\frac{N \times \text{AvgDL} + \gamma MK + NK + \max\{MK, NK\}}{P} + K^2$, where K^2 is for storing $\mathbf{S}$ or $\mathbf{\Sigma}$, $\frac{\gamma MK + NK}{P}$ is for storing $\mathbf{U}$ and $\mathbf{V}$ in P processors, and $\frac{\max\{MK, NK\}}{P}$ is for storing $\mathbf{R}$ or $\mathbf{\Phi}$ in P processors. Since $K \ll M$, it is clear that distributed RLSI has better scalability. We can reach the same conclusion when comparing distributed RLSI with other parallel/distributed topic modeling methods. The key is that distributed RLSI can distribute both terms and documents over P processors. The sparsity of the term-topic matrix can also help save space in each processor.

The time complexities of different topic modeling methods are also listed. For distributed RLSI, T_i is the number of inner iterations in Algorithm 2; C_U and C_V are for the matrix operations in Algorithms 2 and 3 (e.g., $\mathbf{V}\mathbf{V}^T$, $\mathbf{D}\mathbf{V}^T$, $\mathbf{U}^T\mathbf{U}$, $\mathbf{U}^T\mathbf{D}$, and matrix inversion), respectively:

$$C_U = \max\left\{\frac{\text{AvgDL} \times NK}{P} + \text{nnz}(\mathbf{R}) \log P, \frac{NK^2}{P} + K^2 \log P\right\},$$
$$C_V = \max\left\{\frac{\text{AvgDL} \times \gamma NK}{P} + \text{nnz}(\mathbf{\Phi}) \log P, \frac{M(\gamma K)^2}{P} + K^2 \log P + K^3\right\},$$

where nnz$(\cdot)$ is the number of nonzero entries in the input matrix. For details please refer to [Liu et al. 2010]. Note that the time complexities of these methods are comparable.

Table V. Complexities of parallel/distributed topic models.

Method	Space complexity	Time complexity (per iteration)
AD-LDA	$\frac{N \times \text{AvgDL} + NK}{P} + MK$	$\frac{NK \times \text{AvgDL}}{P} + MK \log P$
Async-CGS	$\frac{N \times \text{AvgDL} + NK}{P} + 2MK$	$\frac{NK \times \text{AvgDL}}{P} + MK \log P$
Async-CVB	$\frac{N \times \text{AvgDL} + 2NK}{P} + 4MK$	$\frac{MK}{P} + MK \log P$
Distributed RLSI	$\frac{N \times \text{AvgDL} + \gamma MK + NK + \max\{MK, NK\}}{P} + K^2$	$\frac{T_i MK^2 + NK^2}{P} + C_U + C_V$

7. RELEVANCE RANKING

Topic models can be used in a wide variety of applications. We apply RLSI to relevance ranking in information retrieval (IR) and evaluate its performance in comparison to existing topic modeling methods. The use of topic modeling techniques such as LSI was proposed in IR many years ago [Deerwester et al. 1990]. Some recent work [Wei and Croft 2006; Yi and Allan 2009; Lu et al. 2011] showed improvements in relevance ranking by applying probabilistic topic models such as LDA and PLSI.

The advantage of incorporating topic modeling in relevance ranking is to reduce "term mismatch". Traditional relevance models, such as VSM [Salton et al. 1975] and BM25 [Robertson et al. 1994], are all based on term matching. The term mismatch problem arises when the authors of documents and the users of search systems use different terms to describe the same concepts. As a result relevant documents may get low relevance scores. For example, if the query contains the term "airplane" and the document contains the term "aircraft", then there is a mismatch between the two and the document may not be retrieved. However, if the two terms are included in the same topic the use of matching score in the topic space can help solve the mismatch problem. In practice it is beneficial to combine topic matching scores with term matching scores, to leverage both broad topic matching and specific term matching.

A simple and effective approach for combining the two is to use a linear combination, which was first proposed in [Hofmann 1999] and then adopted in [Kontostathis 2007; Atreya and Elkan 2010]. The final relevance ranking score $s(q, d)$ is:

$$s(q, d) = \alpha s_{topic}(q, d) + (1 - \alpha) s_{term}(q, d), \tag{9}$$

where $\alpha \in [0, 1]$ is the interpolation coefficient. $s_{term}(q, d)$ can be calculated with any of the conventional relevance models such as VSM and BM25. Another combination approach is to incorporate the topic matching score as a feature in a learning to rank model, e.g., LambdaRank [Burges et al. 2007]. In this paper, we use both approaches in our experiments.

For the probabilistic approaches, the combination can also be realized by smoothing the document language models or query language models with the topic models [Wei and Croft 2006; Yi and Allan 2009; Lu et al. 2011]. In this paper, we use linear combinations for the probabilistic approaches as well, and our experimental results show that they are still quite effective.

We next describe how to calculate the topic matching score between query and document, with RLSI as an example. Given a query and document, we first calculate their matching scores in both term space and topic space. For query q, we represent it in the topic space:

$$\boldsymbol{v}_q = \arg\min_{\boldsymbol{v}} \|\boldsymbol{q} - \mathbf{U}\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2,$$

where vector $\boldsymbol{q}$ is the tf-idf representation of query q in the term space[8]. Similarly, for document d (and its tf-idf representation $\boldsymbol{d}$ in the term space) we represent it in the topic space as $\boldsymbol{v}_d$. The matching score between the query and the document in the topic space is, then, calculated as the cosine similarity between $\boldsymbol{v}_q$ and $\boldsymbol{v}_d$:

$$s_{topic}(q, d) = \frac{\langle \boldsymbol{v}_q, \boldsymbol{v}_d \rangle}{\|\boldsymbol{v}_q\|_2 \cdot \|\boldsymbol{v}_d\|_2}.$$

[8]Using $\boldsymbol{v}_q = \arg\min_{\boldsymbol{v}} \|\boldsymbol{q} - \mathbf{U}\boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_1$ if ℓ_1 norm is imposed on $\mathbf{V}$

The topic matching score $s_{topic}(q, d)$ is then combined with the term matching score $s_{term}(q, d)$ in relevance ranking.

8. EXPERIMENTS

Our experiments compare different RLSI regularization strategies, compare RLSI with existing topic modeling methods, test the capability of online RLSI for dynamic topic modeling, compare online RLSI with batch RLSI, and test the scalability of distributed RLSI.

8.1. Experimental Settings

Our three TREC datasets were AP, WSJ, and OHSUMED, which are widely used in relevance ranking experiments. AP consists of the Associated Press articles from February to December 1988. WSJ consists of the Wall Street Journal articles from April 1990 to March 1992. OHSUMED consists of MEDLINE documents from 1987 to 1991. In AP, WSJ, and OHSUMED, the documents are time stamped. For AP and WSJ, we used the titles of TREC topics 51 - 300[9] as queries. For OHSUMED, there are 106 queries associated[10]. We also used a large real-world web dataset from a commercial web search engine, containing about 1.6 million documents and 10 thousand queries. There is no time information for the web dataset, and the documents are randomly ordered.

Besides documents and queries, each dataset has relevance judgments on some documents with respect to each query. For all four datasets, only the judged documents were included and the titles and bodies were taken as the contents of the documents[11]. From the four datasets, stop words in a standard list were removed[12]. From the web dataset, the terms whose frequencies are less than two were further discarded. Table VI gives some statistics on the datasets. We utilized tf-idf to represent the weight of a term in a document given a document collection. The formula for calculating tf-idf which we employed is

$$\text{tf-idf}\,(t, d, \mathcal{D}) = \frac{n\,(t, d)}{|d|} \times \log \frac{|\mathcal{D}|}{|\{d \in \mathcal{D} : t \in d\}|},$$

where t refers to a term, d refers to a document, $\mathcal{D}$ refers to a document collection, $n\,(t, d)$ is the number of times that term t appears in document d, $|d|$ is the length of document d, $|\mathcal{D}|$ is the total number of documents in the collection, and $|\{d \in \mathcal{D} : t \in d\}|$ is the number of documents in which term t appears.

In AP and WSJ the relevance judgments are at two levels: “relevant” or “irrelevant”. In OHSUMED, the relevance judgments are at three levels: “definitely relevant”, “partially relevant”, and “not relevant”. In the web dataset, there are five levels: “perfect”, “excellent”, “good”, “fair”, and “bad”. In the experiments of relevance ranking, we used MAP and NDCG at the positions of 1, 3, 5, and 10 to evaluate the performance. In calculating MAP, we considered “definitely relevant” and “partially relevant” in OHSUMED, and “perfect”, “excellent”, and “good” in web dataset as “relevant”.

In the experiments on the TREC datasets (Section 8.2), no validation set was used since we only have small query sets. Instead, we chose to evaluate each model in a pre-defined grid of parameters, showing its performance under the best parameter choices. In the experiments on the web dataset (Section 8.3), the queries were randomly split into training/validation/test sets, with 6,000/2,000/2,680 queries, respectively. We trained the ranking models with the training set, selected the best models with the validation set, and evaluated the performances of the methods with the test set. We selected models based on their NDCG@1 values, because NDCG is more suitable as the evaluation measure in web search. The reasons are as follows. First, MAP is based on two-level

---

[9] http://trec.nist.gov/data/intro_eng.html
[10] http://ir.ohsu.edu/ohsumed/ohsumed.html
[11] Note that the whole datasets are too large to handle for the baseline methods such as LDA. Therefore, only the judged documents were used.
[12] http://www.textfixer.com/resources/common-english-words.txt

Table VI. Statistics of datasets.

Dataset	AP	WSJ	OHSUMED	Web
# terms	83,541	106,029	26,457	7,014,881
# documents	29,528	45,305	14,430	1,562,807
# queries	250	250	106	10,680

relevance judgments, while NDCG is based on multi-level relevance judgments, which is more common in web search. Second, MAP takes into account all relevant documents, while NDCG focuses on top-ranked documents, which is more essential in web search.

The experiments on AP, WSJ, and OHSUMED were conducted on a server with Intel Xeon 2.33GHZ CPU, 16GB RAM. The experiments on the web dataset were conducted on a distributed system and the distributed RLSI (both batch and online) was implemented with the SCOPE language [Chaiken et al. 2008].

8.2. Experiments on TREC Datasets

8.2.1. Regularization Strategies. In this experiment, we compared different regularization strategies on (batch) RLSI. Regularization on **U** and **V** via either ℓ_1 or ℓ_2 norm gives us four RLSI variants: RLSI (Uℓ_1-Vℓ_2), RLSI (Uℓ_2-Vℓ_1), RLSI (Uℓ_1-Vℓ_1), and RLSI (Uℓ_2-Vℓ_2), where RLSI (Uℓ_1-Vℓ_2) means, for example, applying ℓ_1 norm on **U** and ℓ_2 norm on **V**. For all the variants, parameters K, λ_1, and λ_2 were respectively set in ranges of $[10, 50]$, $[0.01, 1]$, and $[0.01, 1]$, and interpolation coefficient α was set from 0 to 1 in steps of 0.05. We ran all the methods in 100 iterations (convergence confirmed).

We first compared the RLSI variants in terms of topic readability, by looking at the contents of topics they generated. Note that throughout the paper, topic readability refers to coherence of top weighted terms in a topic. We adopt the terminology "readability" from Stanford Topic Modeling Toolbox[13]. As example, Table VII shows 10 topics (randomly selected) and the average topic compactness (AvgComp) on AP dataset for each of the four RLSI variants, when $K = 20$ and λ_1 and λ_2 are the optimal parameters for the retrieval experiment described below. Here, average topic compactness is defined as average ratio of terms with non-zero weights per topic. For each topic, its top 5 weighted terms are shown[14]. From the results, we have found that 1) if ℓ_1 norm is imposed on either **U** or **V**, RLSI can always discover readable topics; 2) without ℓ_1 regularization (i.e., RLSI(Uℓ_2-Vℓ_2)), many topics are not readable; 3) if ℓ_1 norm is only imposed on **V** (i.e. RLSI (Uℓ_2-Vℓ_1)), the discovered topics are not compact or sparse (e.g., AvgComp = 1). We also conducted the same experiments on WSJ and OHSUMED and observed similar phenomena.

We also compared the RLSI variants in terms of retrieval performance. Specifically, for each of the RLSI variants, we combined topic matching scores with term matching scores given by conventional IR models of VSM or BM25. When calculating BM25 scores, we used the default parameters, i.e., $k_1 = 1.2$ and $b = 0.75$. Since BM25 performs better than VSM on AP and WSJ, and VSM performs better than BM25 on OHSUMED, we combined the topic matching scores with BM25 on AP and WSJ, and with VSM on OHSUMED. The methods we tested are denoted as "BM25+RLSI (Uℓ_1-Vℓ_2)", "BM25+RLSI (Uℓ_2-Vℓ_1)", "BM25+RLSI (Uℓ_1-Vℓ_1)", "BM25+RLSI (Uℓ_2-Vℓ_2)", etc. Tables VIII, IX, and X show the retrieval performance of RLSI variants achieved by the best parameter setting (measured by NDCG@1) on AP, WSJ, and OHSUMED, respectively. Stars indicate significant improvements on the baseline method, i.e., BM25 on AP and WSJ and VSM on OHSUMED, according to the one-sided t-test (p-value < 0.05)[15]. From the results, we can see that 1) all of these methods can improve over the baseline and in some cases the improvements

[13] http://nlp.stanford.edu/software/tmt/tmt-0.4/

[14] In all the results presented in this paper, the terms with the dominating contribution in a topic were used to represent the topic. The dominating contribution will be discussed later in Section 8.4.

[15] Note that in all the experiments, we tested whether the ranking performance of one method (method A) is significantly better than that of the other method (method B). Thus, the alternative hypothesis is that the NDCG/MAP value of method A is larger than that of method B, which is a one-sided significance test.

Table VII. Topics discovered by RLSI variants on AP.

RLSI ($U\ell_1$-$V\ell_2$) AvgComp = 0.0075	bush	yen	student	israeli	opec
	dukakis	trade	school	palestinian	oil
	quayle	dollar	teacher	israel	cent
	bentsen	japan	educate	arab	barrel
	campaign	market	protest	plo	price
	noriega	quake	iran	court	soviet
	panama	earthquake	iranian	prison	nuclear
	panamanian	richter	iraq	sentence	treaty
	delva	scale	iraqi	judge	missile
	canal	damage	gulf	trial	weapon
RLSI ($U\ell_2$-$V\ell_1$) AvgComp = 1	nuclear	court	noriega	africa	cent
	treaty	judge	panama	south	opec
	missile	prison	panamanian	african	oil
	weapon	trial	delval	angola	barrel
	soviet	sentence	canal	apartheid	price
	israeli	dukakis	student	plane	percent
	palestinian	bush	school	crash	billion
	israel	jackson	teacher	flight	rate
	arab	democrat	educate	air	0
	plo	campaign	college	airline	trade
RLSI ($U\ell_1$-$V\ell_1$) AvgComp = 0.0197	court	plane	dukakis	israeli	africa
	prison	crash	bush	palestinian	south
	judge	air	jackson	israel	african
	sentence	flight	democrat	arab	angola
	trial	airline	campaign	plo	apartheid
	soviet	school	yen	cent	noriega
	treaty	student	trade	opec	panama
	missile	teacher	dollar	oil	panamanian
	nuclear	educate	market	barrel	delval
	gorbachev	college	japan	price	canal
RLSI ($U\ell_2$-$V\ell_2$) AvgComp = 1	dukakis	palestinian	soviet	school	africa
	oil	israeli	noriega	student	south
	opec	israel	panama	bakker	iran
	cent	arab	drug	trade	african
	bush	plo	quake	china	dukakis
	dukakis	soviet	drug	percent	soviet
	bush	treaty	cent	billion	israeli
	democrat	student	police	price	missile
	air	nuclear	student	trade	israel
	jackson	missile	percent	cent	treaty

Table VIII. Retrieval performance of RLSI variants on AP.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3918	0.4400	0.4268	0.4298	0.4257
BM25+RLSI ($U\ell_1$-$V\ell_2$)	**0.3998** *	**0.4800** *	**0.4461** *	**0.4498** *	**0.4420** *
BM25+RLSI ($U\ell_2$-$V\ell_1$)	0.3964	0.4640	0.4337	0.4357	0.4379 *
BM25+RLSI ($U\ell_1$-$V\ell_1$)	0.3987 *	0.4640 *	0.4360	0.4375	0.4363 *
BM25+RLSI ($U\ell_2$-$V\ell_2$)	0.3959	0.4520	0.4409	0.4337	0.4314

are statistically significant; 2) among the RLSI variants, RLSI ($U\ell_1$-$V\ell_2$) performs best and its improvements over baseline are significant on all three TREC datasets; 3) any improvement of RLSI ($U\ell_1$-$V\ell_2$) over other RLSI variants, however, is not significant.

Table XI summarizes the experimental results in terms of topic readability, topic compactness, and retrieval performance. From the result, we can see that in RLSI, ℓ_1 norm is essential for discovering readable topics, and the discovered topics will also be compact if ℓ_1 norm is imposed on $\mathbf{U}$. Furthermore, between the two RLSI variants with good topic readability and compactness, i.e., RLSI ($U\ell_1$-$V\ell_2$) and RLSI ($U\ell_1$-$V\ell_1$), RLSI ($U\ell_1$-$V\ell_2$) performs better in improving retrieval performance. This is because when ℓ_1 norm is imposed on $\mathbf{V}$, the document and query representa-

Table IX. Retrieval performance of RLSI variants on WSJ.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.2935	0.3720	0.3717	0.3668	0.3593
BM25+RLSI ($U\ell_1$-$V\ell_2$)	0.2968	**0.4040** *	**0.3851** *	0.3791 *	**0.3679** *
BM25+RLSI ($U\ell_2$-$V\ell_1$)	0.2929	0.3960	0.3738	0.3676	0.3627
BM25+RLSI ($U\ell_1$-$V\ell_1$)	**0.2970**	0.3960	0.3827	**0.3798** *	0.3668 *
BM25+RLSI ($U\ell_2$-$V\ell_2$)	0.2969	0.3920	0.3788	0.3708	0.3667 *

Table X. Retrieval performance of RLSI variants on OHSUMED.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
VSM	0.4288	0.4780	0.4159	0.3932	0.3840
VSM+RLSI ($U\ell_1$-$V\ell_2$)	0.4291	**0.5377** *	**0.4383** *	**0.4145** *	**0.4010** *
VSM+RLSI ($U\ell_2$-$V\ell_1$)	0.4282	0.5252	0.4351	0.4018	0.3952
VSM+RLSI ($U\ell_1$-$V\ell_1$)	0.4285	**0.5377** *	0.4291	0.4105	0.3972
VSM+RLSI ($U\ell_2$-$V\ell_2$)	**0.4310**	0.5189 *	0.4279	0.4078 *	0.3928 *

Table XI. Performance of the RLSI variants.

	Topic Readability	Topic Compactness	Retrieval performance
RLSI ($U\ell_1$-$V\ell_2$)	√	√	√
RLSI ($U\ell_2$-$V\ell_1$)	√	×	×
RLSI ($U\ell_1$-$V\ell_1$)	√	√	×
RLSI ($U\ell_2$-$V\ell_2$)	×	×	×

tions in the topic space will also be sparse, and thus the topic matching scores will not be reliable enough. We conclude that it is a better practice to apply ℓ_1 norm on $\mathbf{U}$ and ℓ_2 norm on $\mathbf{V}$ in RLSI, for achieving good topic readability, topic compactness, and retrieval performance.

We will use RLSI ($U\ell_1$-$V\ell_2$) in the following experiments and denote it as RLSI for simplicity.

8.2.2. Comparison of Topic Models. In this experiment, we compared (batch) RLSI with LDA, PLSI, LSI, and NMF.

We first compared RLSI with LDA, PLSI, LSI, and NMF in terms of topic readability, by looking at the topics they generated. We made use of publically available tools when running the baselines[16]. The number of topics K was again set to 20 for all the methods. In RLSI, λ_1 and λ_2 were the optimal parameters used in Section 8.2.1 (i.e., $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$). For LDA, PLSI, LSI, and NMF, there is no additional parameter to tune. Table XII show all the 20 topics discovered by RLSI, LDA, PLSI, LSI, and NMF, and the average topic compactness (AvgComp) on AP dataset. For each topic, its top 5 weighted terms are shown. From the results, we have found 1) RLSI can discover readable and compact (e.g., AvgComp = 0.0075) topics; 2) PLSI, LDA, and NMF can discover readable topics as expected, however the discovered topics are not so compact (e.g., AvgComp = 0.9534, AvgComp = 1, and AvgComp = 0.5488, respectively); 3) the topics discovered by LSI are hard to understand perhaps due to its orthogonality assumption. We also conducted the same experiments on WSJ and OHSUMED and observed similar phenomena.

We further evaluated the quality of the topics discovered by (batch) RLSI, LDA, PLSI, and NMF, in terms of topic representability and topic overlap. Here, topic representability is defined as average contribution of top terms in each topic, where the contribution of top terms in a topic is defined as the sum of absolute weights of top terms divided by the sum of absolute weights of all terms. Topic representability indicates how well the topics can be described by their top terms. The larger the topic representability is, the better the topics can be described by their top terms. Topic overlap is defined as average overlap of the top terms among topic pairs. Topic overlap indicates how distinct the topics are. The smaller the topic overlap is, the more distinct the topics are. Figure 5 and Figure 6 show the representability and overlap of the topics discovered by (batch) RLSI, LDA, PLSI, and

[16]LDA: `http://www.cs.princeton.edu/~blei/lda-c/`; PLSI: `http://www.lemurproject.org/`; LSI: `http://tedlab.mit.edu/~dr/SVDLIBC/`; NMF: `http://cogsys.imm.dtu.dk/toolbox/nmf/`

Table XII. Topics discovered by batch RLSI, LDA, PLSI, LSI, and NMF on AP.

Batch RLSI AvgComp = 0.0075	bush	yen	student	contra	israeli
	dukakis	trade	school	sandinista	palestinian
	quayle	dollar	teacher	rebel	israel
	bentsen	japan	educate	nicaragua	arab
	campaign	market	protest	nicaraguan	plo
	senate	opec	noriega	drug	soviet
	program	oil	panama	test	afghanistan
	house	cent	panamanian	cocain	afghan
	reagan	barrel	delva	aid	gorbachev
	state	price	canal	trafficker	pakistan
	percent	quake	jackson	iran	court
	0	earthquake	dukakis	iranian	prison
	rate	richter	democrat	iraq	sentence
	billion	scale	delegate	iraqi	judge
	increase	damage	party	gulf	trial
	police	firefighter	soviet	hostage	africa
	kill	acr	nuclear	lebanon	south
	crash	forest	treaty	beirut	african
	plane	park	missile	hijack	angola
	air	blaze	weapon	hezbollah	apartheid
LDA AvgComp = 1	soviet	school	dukakis	party	year
	nuclear	student	democrat	govern	new
	union	year	campaign	minister	time
	state	educate	bush	elect	television
	treaty	university	jackson	nation	film
	water	price	court	police	iran
	year	year	charge	south	iranian
	fish	market	case	govern	ship
	animal	trade	judge	kill	iraq
	0	percent	attorney	protest	navy
	people	percent	state	state	president
	0	1	govern	house	reagan
	city	year	unit	senate	bush
	mile	million	military	year	think
	area	0	american	congress	american
	air	company	police	plant	health
	plane	million	year	worker	aid
	flight	bank	death	strike	us
	crash	new	kill	union	test
	airline	year	old	new	research
PLSI AvgComp = 0.9534	company	israeli	bush	year	govern
	million	iran	dukakis	state	military
	share	israel	democrat	new	south
	billion	palestinian	campaign	nation	state
	stock	arab	republican	0	president
	soviet	year	pakistan	mile	year
	treaty	movie	afghan	0	state
	missile	film	guerrilla	people	new
	nuclear	new	afghanistan	area	people
	gorbachev	play	vietnam	year	nation
	percent	year	plane	year	court
	0	state	flight	animal	charge
	10	new	airline	people	attorney
	12	nation	crash	new	judge
	1	govern	air	0	trial
	year	year	percent	year	year
	state	aid	price	state	police
	new	us	market	new	offici
	nation	new	1	nation	report
	govern	study	billion	govern	state

	soviet	567	0	earthquake	drug
	percent	234	yen	quake	school
	police	0	dollar	richter	test
	govern	percent	percent	scale	court
	state	12	tokyo	damage	dukakis
	0	yen	yen	urgent	soviet
	dukakis	police	dukakis	oil	0
	bush	0	bush	opec	test
	jackson	dollar	dollar	dukakis	nuclear
LSI AvgComp = 1	dem	kill	jackson	cent	urgent
	lottery	bakker	israel	south	bakker
	lotto	ptl	israeli	africa	ptl
	weekly	lottery	student	rebel	spe
	pick	lotto	palestinian	african	israeli
	connecticut	soviet	africa	angola	israel
	spe	bakker	noriega	hostage	student
	bc	virus	panama	hamadi	school
	iran	aid	plane	hijack	noriega
	iranian	ptl	drug	africa	panama
	school	infect	contra	south	teacher
	spe	iran	yen	noriega	soviet
	bc	iranian	dollar	panama	nuclear
	car	hostage	tokyo	contra	treaty
	laserphoto	iraq	exchange	sandinista	missile
	mature	lebanon	close	rebel	gorbachev
	lottery	africa	576	urgent	0
	lotto	south	234	caliguiry	percent
	weekly	african	12	allegheny	dem
	connecticut	angola	percent	ercan	uncommitted
NMF AvgComp = 0.5488	pick	mandela	precinct	coron	gop
	bakker	earthquake	plane	police	israeli
	ptl	quake	crash	kill	isra
	ministry	richter	flight	firefighter	palestinian
	benton	scale	air	injure	plo
	bankruptcy	damage	airline	car	arab
	test	court	percent	dukakis	opec
	virus	prison	billion	bush	oil
	school	sentence	company	jackson	cent
	aid	judge	trade	democrat	barrel
	patient	charge	million	campaign	price

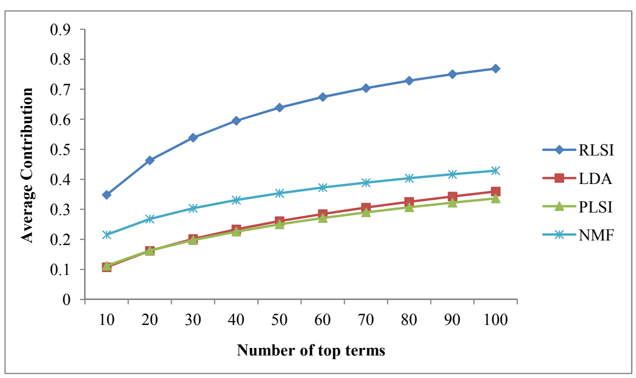


Fig. 5. Topic representability of different methods when number of top terms increases.

NMF when number of top terms increases. The results show that 1) RLSI has much larger topic representability than NMF, LDA, and PLSI, indicating that the topics discovered by RLSI can be described by their top terms better than the topics discovered by the other methods; 2) RLSI and NMF have smaller topic overlap than LDA and PLSI, indicating that the topics discovered by RLSI and NMF are more distinct from each other. We also conducted the same experiments on WSJ and OHSUMED and observed similar trends.

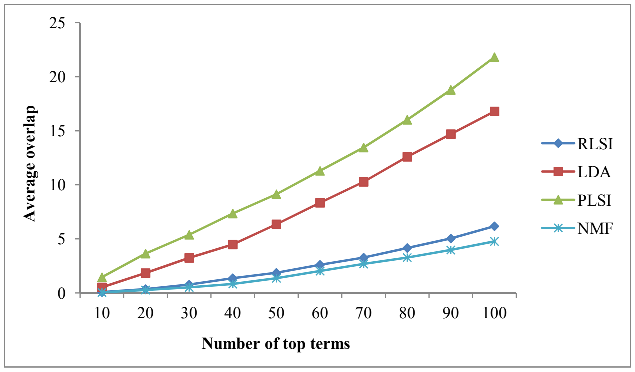


Fig. 6. Topic overlap of different methods when number of top terms increases.

Table XIII. Retrieval performance of topic models on AP.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3918	0.4400	0.4268	0.4298	0.4257
BM25+LSI	0.3952	0.4720	0.4410	0.4360	0.4365
BM25+PLSI	0.3928	0.4680	0.4383	0.4351	0.4291
BM25+LDA	0.3952	0.4760 *	**0.4478** *	0.4332	0.4292
BM25+NMF	0.3985 *	0.4600	0.4445 *	0.4408 *	0.4347 *
BM25+RLSI	**0.3998** *	**0.4800** *	0.4461 *	**0.4498** *	**0.4420** *

Table XIV. Retrieval performance of topic models on WSJ.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.2935	0.3720	0.3717	0.3668	0.3593
BM25+LSI	0.2953	0.3800	0.3765	0.3710	0.3615
BM25+PLSI	0.2976 *	0.3800	0.3815 *	0.3738 *	0.3619
BM25+LDA	**0.2996** *	0.3960	**0.3858** *	0.3777 *	**0.3683** *
BM25+NMF	0.2954	0.3880	0.3772	0.3725	0.3616
BM25+RLSI	0.2968	**0.4040** *	0.3851 *	**0.3791** *	0.3679 *

We also tested the performance of (batch) RLSI in terms of retrieval performance, in comparison to LSI, PLSI, LDA, and NMF. The experimental setting was similar to that in Section 8.2.1. For the five methods, parameter K was set in range of $[10, 50]$, and interpolation coefficient α was set from 0 to 1 in steps of 0.05. For RLSI, parameter λ_2 was fixed to 1 and parameter λ_1 was set in range of $[0.1, 1]$. LSI, PLSI, LDA, and NMF have no additional parameters to tune. Tables XIII, XIV, and XV show retrieval performance achieved by the best parameter setting (measured by NDCG@1) on AP, WSJ, and OHSUMED, respectively. Stars indicate significant improvements on the baseline method, i.e., BM25 on AP and WSJ and VSM on OHSUMED, according to the one-sided t-test (p-value < 0.05). From the results, we can see that 1) RLSI can significantly improve the baseline, going beyond the simple term matching paradigm; 2) among the different topic modeling methods, RLSI and LDA perform slightly better than the other methods, and sometimes the improvements are statistically significant; 3) any improvement of RLSI over LDA, however, is not significant. We conclude that RLSI is a viable choice for improving relevance.

8.2.3. Online RLSI for Topic Tracking. In this experiment, we tested the capability of online RLSI for topic tracking. Here, we adopted online RLSI with ℓ_1 regularization on topics and ℓ_2 regularization

Table XV. Retrieval performance of topic models on OHSUMED.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
VSM	0.4288	0.4780	0.4159	0.3932	0.3840
VSM+LSI	0.4296	0.4969	0.4337	0.4085	0.3948 *
VSM+PLSI	0.4325	0.4843	0.4171	0.3978	0.3820
VSM+LDA	**0.4326**	0.5094 *	**0.4474** *	0.4115 *	0.3906
VSM+NMF	0.4293	0.5000	0.4316 *	0.4087 *	0.3937 *
VSM+RLSI	0.4291	**0.5377** *	0.4383 *	**0.4145** *	**0.4010** *

on document representations[17]. Documents were treated as a stream ordered by their time stamps, and the entire collection was processed in exactly one pass.

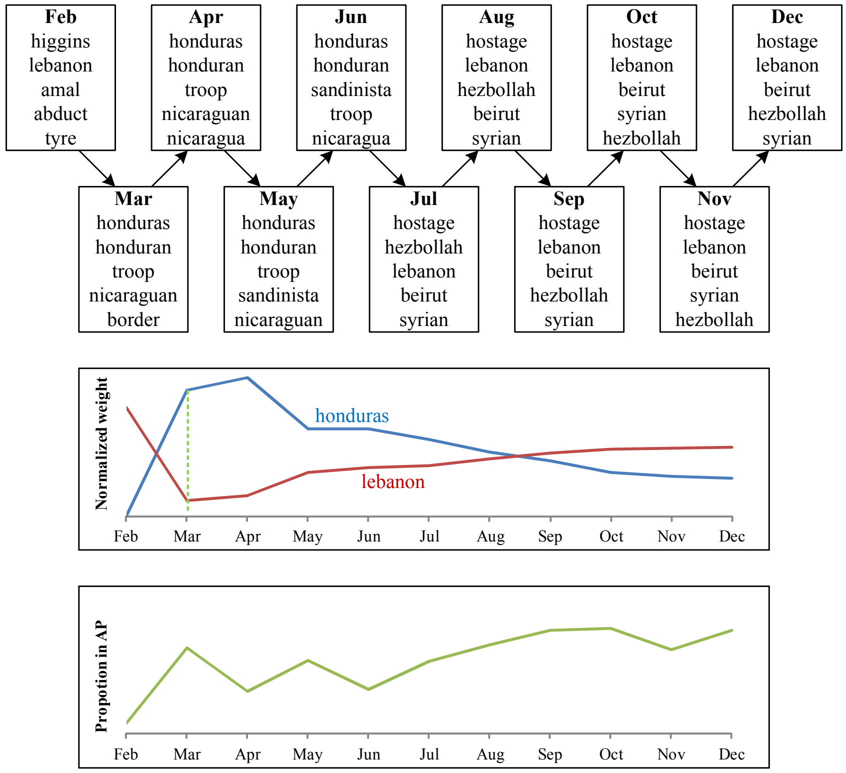
To test the performance of the basic version (described in Section 5.2) and the improved version (described in Section 5.4) of online RLSI, we first decided the ranges of the parameter $\rho \in \{0, 0.1, 0.2, 0.5, 1, 2, 5, 10\}$, $\eta \in \{1, 2, 5, 10, 20, 50, 100\}$, and $\xi \in \{1, 2, 5, 10, 20, 50, 100\}$, and selected the best parameters for the two versions. The basic version of online RLSI was run with $\rho = 0$, $\eta = 1$, and $\xi = 1$. The improved version of online RLSI was run with $\rho = 1$, $\eta = 10$, and $\xi = 10$. This is because we observed that 1) to make online RLSI capable of topic tracking, "re-scaling" (controlled by ρ) and "embedded iterations" (controlled by ξ) are necessary, and the improved version of online RLSI is capable of capturing the evolution of latent topics only when $\rho \geq 1$ and $\xi \geq 10$; 2) "mini-batch" (controlled by η) does not make a critical impact on topic tracking, but it can save execution time when η is large.

Figure 7 and Figure 8 show two example topics discovered by online RLSI on AP dataset, with $K = 20$ and λ_1 and λ_2 set to the optimal parameters for the retrieval experiment described below (i.e., $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$). The figures show the proportion of the two topics in the AP dataset, as well as some example documents talking about the topics, over the time period of the corpus. Here, the proportion of a topic in a document is defined as the absolute weight of the topic in the document normalized by the ℓ_2 norm of the document. The proportion of a topic in a dataset is then defined as the sum over all the documents. For each topic, its top 5 weighted terms in each month are shown. Also shown are the normalized weights of the representative terms in each topic, along the time axis. Here, the normalized weight of a term in a topic is defined as the absolute weight of the term in the topic normalized by the ℓ_1 norm of the topic. The first topic (Figure 7), with top term "honduras", increases sharply in March 1988. This is because President Reagan ordered over 3,000 U.S. troops to Honduras on March 16 that year, claiming that Nicaraguan soldiers had crossed its borders. About 10% of the AP documents in March reported this event and the AP documents later also followed up on the event. The second topic (Figure 8), with top term "hijack", increases sharply in April 1988. This is because on April 5, a Kuwait Airways Boeing 747 was hijacked and diverted to Algiers on its way to Kuwait from Bangkok. About 8% of the AP documents in April reported this event and the AP documents in later months followed up the event. From the results, we conclude that online RLSI is capable of capturing the evolution of the latent topics, and can be used to track the trends of topics.

8.2.4. Online RLSI vs. Batch RLSI. This experiment compares online RLSI (denoted as "oRLSI") and batch RLSI (denoted as "bRLSI").

We first compared the performance of online RLSI and batch RLSI in terms of topic readability, by looking at the topics they generated. Table XVI shows all the 20 final topics discovered by online RLSI and the average topic compactness (AvgComp) on AP dataset, with $K = 20$ and λ_1 and λ_2 set to the optimal parameters for the retrieval experiment described below (i.e., $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$). For each topic, its top 5 weighted terms are shown. From the results, we have found that 1) online

[17]This regularization strategy in batch RLSI has been demonstrated to be the best as described in Section 8.2.1. We tested all of the four online RLSI variants, with regularization on topics and document representations via either ℓ_1 or ℓ_2 norm, and found a similar trend as in batch RLSI.



(AP880225-0091) Gunmen Hurl Grenade At Militiamen Searching for Higgins
(AP880317-0167) American Troops Arrive in Honduras
(AP880409-0091) Anti-US Violence in Honduras Unexpected
(AP880502-0157) House Refuses To Ban US Manuevers Along Honduras-Nicaragua Border
(AP880602-0216) Nicaragua Denies Soldiers Crossed Honduran Border
(AP880725-0031) Hezbollah Holy Man Calls For `Happy Ending' To Hostage Crisis
(AP880813-0048) Hezbollah Official Denounces U.N. Resolution on Higgins
(AP880912-0161) Syria Helps Free American Engineer Kidnapped In Lebanon
(AP881019-0214) Christian Leader Denounces Syrian Influence in Lebanon
(AP881105-0140) Amal and Hezbollah Agree To Non-Kidnapping Pact
(AP881210-0063) Shiite Cleric Says Hostages' Issue On Ice

Fig. 7. Example topic discovered by online RLSI on AP.

Table XVI. Topics discovered by online RLSI on AP (AvgComp = 0.0079).

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10
africa	noriega	opec	student	tax	percent	dukakis	hostage	hijack	drug
south	panama	oil	school	budget	billion	bush	lebanon	plane	aid
african	panamanian	cent	teacher	billion	rate	jackson	beirut	hamadi	test
angola	delva	barrel	educate	senate	trade	democrat	hezbollah	crash	virus
apartheid	military	price	college	reagan	price	campaign	syrian	hostage	infect
Topic 11	Topic 12	Topic 13	Topic 14	Topic 15	Topic 16	Topic 17	Topic 18	Topic 19	Topic 20
police	0	contra	iran	palestinian	bush	soviet	gang	yen	bakker
court	party	sandinista	iranian	israel	robertson	treaty	police	dollar	ptl
people	delegate	rebel	iraq	israeli	quayle	nuclear	drug	tokyo	swaggart
prison	percent	nicaragua	iraqi	plo	republican	missile	arrest	trade	ministry
govern	democrat	ortega	gulf	arab	reagan	gorbachev	cocain	market	church

RLSI can discover readable and compact (e.g., AvgComp = 0.0079) topics; 2) the topics discovered by online RLSI are similar to those discovered by batch RLSI, as in Table XII.

We also compared the performance of online RLSI and batch RLSI in terms of retrieval performance. The experimental setting was similar to that in Section 8.2.2. For both cases, parameter K

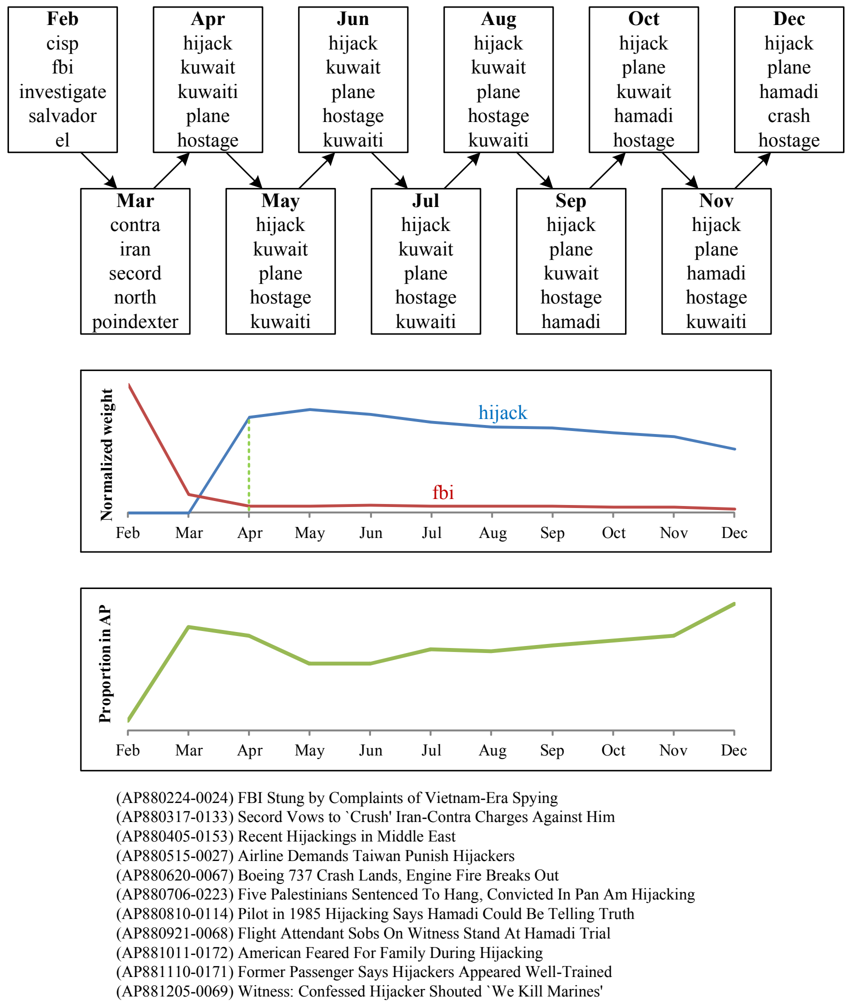


Fig. 8. Example topic discovered by online RLSI on AP.

was set in range of $[10, 50]$, parameter λ_2 was fixed to 1, parameter λ_1 was set in range of $[0.1, 1]$, and interpolation coefficient α was set from 0 to 1 in steps of 0.05. Tables XVII, XVIII, and XIX show the retrieval performances achieved by the best parameter setting (measured by NDCG@1) on AP, WSJ, and OHSUMED, respectively. Stars indicate significant improvement on the baseline method, i.e., BM25 on AP and WSJ and VSM on OHSUMED, according to the one-sided t-test (p-value < 0.05). From the results, we can see that 1) online RLSI can improve the baseline, and in most cases, the improvement is statistically significant; 2) online RLSI performs slightly worse than batch RLSI, however, the improvement of batch RLSI over online RLSI is not statistically significant. This is because online RLSI updates the term-topic matrix as well as the document representation(s) with the documents observed so far, while batch RLSI updates the term-topic matrix as well as the topic-document matrix with the whole document collection.

We conclude that online RLSI can discover readable and compact topics and can achieve high enough accuracy in relevance ranking. More importantly, online RLSI can capture the temporal evolution of the topics, which batch RLSI cannot.

Table XVII. Retrieval performance of online RLSI and batch RLSI on AP.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3918	0.4400	0.4268	0.4298	0.4257
BM25+bRLSI	**0.3998** *	**0.4800** *	**0.4461** *	**0.4498** *	**0.4420** *
BM25+oRLSI	0.3980	0.4720 *	0.4455 *	0.4419	0.4386 *

Table XVIII. Retrieval performance of online RLSI and batch RLSI on WSJ.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.2935	0.3720	0.3717	0.3668	0.3593
BM25+bRLSI	**0.2968**	**0.4040** *	**0.3851** *	**0.3791** *	**0.3679** *
BM25+oRLSI	0.2947	**0.4040** *	0.3836 *	0.3743	0.3646

Table XIX. Retrieval performance of online RLSI and batch RLSI on OHSUMED.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
VSM	0.4288	0.4780	0.4159	0.3932	0.3840
VSM+bRLSI	**0.4291**	**0.5377** *	**0.4383** *	**0.4145** *	0.4010 *
VSM+oRLSI	0.4266	0.5252 *	0.4330	0.4091	**0.4020** *

8.3. Experiments on Web Dataset

We tested the scalability of both batch RLSI and online RLSI using a large real-world web dataset. Table XX lists the sizes of datasets used to evaluate existing distributed/parallel topic models, as well as the size of the Web dataset in this paper. We can see that the number of terms in the Web dataset is much larger. RLSI can handle much larger datasets with a much smaller number of machines than existing models. (Note that it is difficult for us to re-implement existing parallel topic modeling methods, because most of them require special computing infrastructures and the development costs of the methods are high.)

In the experiments, the number of topics K was set to 500, λ_1 and λ_2 were again set to 0.5 and 1.0 respectively, and the mini-batch size in online RLSI was adjusted to $\eta = 10,000$ because the number of documents is large (e.g., $N = 1,562,807$). It took about 1.5 and 0.6 hour for batch and online RLSI to complete an iteration on the MapReduce system with 16 processors. Table XXI shows 10 randomly selected topics discovered by batch RLSI and online RLSI, and the average topic compactness (AvgComp) on the Web dataset. We can see that the topics obtained by both (distributed) batch RLSI and (distributed) online RLSI are compact and readable.

Next, we tested retrieval performance of distributed RLSI. We took LambdaRank [Burges et al. 2007] as the baseline. There are 16 features used in the LambdaRank model, including BM25, PageRank, and so on. The topic matching scores by batch RLSI and online RLSI were respectively used as a new feature in LambdaRank, and the obtained ranking models are denoted as "LambdaRank+bRLSI" and "LambdaRank+oRLSI" respectively. We randomly split the queries into training/validation/test sets, with 6,000/2,000/2,680 queries, respectively. We trained the ranking models with the training set, selected the best models (measured by NDCG@1) with the validation set, and evaluated the performances of the models with the test set. Tables XXII and XXIII show the ranking performance of batch RLSI and online RLSI on the test set, respectively, where stars indicate significant improvements on the baseline method of LambdaRank according to the one-sided t-test (p-value < 0.05). The results indicate that LambdaRank+bRLSI and LambdaRank+oRLSI enriched by batch and online RLSI can significantly outperform LambdaRank in terms of NDCG@1.

Since other papers reduced the input vocabulary size, we tested the effect of reducing the vocabulary size in RLSI. Specifically, we removed the terms whose total term frequency is less than 100 from the Web dataset obtaining a new dataset with 222,904 terms. We applied both batch RLSI and online RLSI on the new dataset with parameters $K = 500, \lambda_1 = 0.5$ and $\lambda_2 = 1.0$. We then created two LambdaRank models with topic matching scores as features, denoted as "LambdaRank+bRLSI (Reduced Vocabulary)" and "LambdaRank+oRLSI (Reduced Vocabulary)" respectively. Tables XXII and XXIII show the retrieval performances of "LambdaRank+bRLSI (Reduced Vocabulary)" and "LambdaRank+oRLSI (Reduced Vocabulary)" on the test set, where stars indi-

Table XX. Sizes of datasets used in distributed/parallel topic models.

Dataset	# docs	# terms	Applied algorithms
NIPS	1,500	12,419	Async-CVB, Async-CGS, PLDA
Wiki-200T	2,122,618	200,000	PLDA+
PubMed	8,200,000	141,043	AD-LDA, Async-CVB, Async-CGS
Web dataset	1,562,807	7,014,881	Distributed RLSI

Table XXI. Topics discovered by batch RLSI and online RLSI on Web dataset.

Batch RLSI AvgComp = 0.0035	casino	mortgage	wheel	cheap	login
	poker	loan	rim	flight	password
	slot	credit	tire	hotel	username
	game	estate	truck	student	registration
	vegas	bank	car	travel	email
	christian	google	obj	spywar	friend
	bible	web	pdf	anti	myspace
	church	yahoo	endobj	sun	music
	god	host	stream	virus	comment
	jesus	domain	xref	adwar	photo
Online RLSI AvgComp = 0.0018	book	estate	god	law	furniture
	science	real	bible	obama	bed
	math	property	church	war	decoration
	write	sale	christian	govern	bedroom
	library	rental	jesus	president	bathroom
	february	cancer	ebay	jewelry	music
	january	health	store	diamond	song
	october	medical	buyer	ring	album
	december	disease	seller	gold	guitar
	april	patient	item	necklace	artist

Table XXII. Retrieval performance of batch RLSI on Web dataset.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
LambdaRank	0.3076	0.4398	0.4432	0.4561	0.4810
LambdaRank+bRLSI	**0.3116** *	**0.4528** *	**0.4494** *	**0.4615** *	0.4860 *
LambdaRank+bRLSI (Reduced Vocabulary)	0.3082	0.4448 *	0.4483 *	0.4608	**0.4861** *

cate significant improvements on the baseline method of LambdaRank according to the one-sided t-test (p-value < 0.05). The results indicate that reducing the vocabulary size will sacrifice accuracy of RLSI in both batch and online versions, and consequently hurt the retrieval performance. This is because after reducing the vocabulary some of the query terms (as well as the document terms) will not be included in the topic models, and hence the topic matching scores will not be as accurate as before. Let us take query "myspacegraphics" as an example. Without reducing the vocabulary, the query term "myspacegraphics" is mapped to the topic containing "myspace" and "graphics", and thus the relevant documents with respect to the query will get high topic matching scores. However, after reducing the vocabulary, the query term "myspacegraphics" is not included in the topic models, and thus the relevant documents with respect to the query will get zero topic matching scores. This will hurt the retrieval performance. We further conducted one-sided t-test on the difference of NDCG@1 between "LambdaRank+bRLSI (Reduced Vocabulary)" and "LambdaRank+bRLSI", as well as "LambdaRank+oRLSI (Reduced Vocabulary)" and "LambdaRank+oRLSI", and found that the differences are statistically significant (p-value < 0.05) in both cases. We observed the same trends on the TREC datasets for RLSI and LDA and omit the details here.

8.4. Discussions

In this section, we discuss the properties of batch RLSI and online RLSI from the experimental results. Without loss of generality, we take our examples from the AP dataset.

Table XXIII. Retrieval performance of online RLSI on Web dataset.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
LambdaRank	0.3076	0.4398	0.4432	0.4561	0.4810
LambdaRank+oRLSI	0.3088	**0.4478** *	**0.4473** *	**0.4592**	**0.4851** *
LambdaRank+oRLSI (Reduced Vocabulary)	**0.3092**	0.4442 *	0.4464	0.4583	0.4842

Table XXIV. Characteristics of topics by batch RLSI.

	PosContri	NegContri	MR (%)
Topic 1	21.76	1.34	94.18
Topic 2	22.96	1.72	93.04
Topic 3	19.13	1.91	90.92
Topic 4	25.92	0.64	97.58
Topic 5	28.13	0.92	96.83
Topic 6	116.83	1.70	98.57
Topic 7	23.58	1.06	95.69
Topic 8	18.24	0.16	99.14
Topic 9	16.26	0.44	97.35
Topic 10	3.17	20.33	86.51
Topic 11	43.35	1.18	97.35
Topic 12	19.17	0.03	99.86
Topic 13	26.43	1.22	95.60
Topic 14	24.12	0.91	96.36
Topic 15	32.82	4.00	89.14
Topic 16	52.61	6.84	88.50
Topic 17	24.82	0.47	98.13
Topic 18	28.19	2.20	92.77
Topic 19	24.63	0.32	98.71
Topic 20	0.33	19.54	98.31
Average	—-	—-	95.23

Table XXV. Characteristics of topics by online RLSI.

	PosContri	NegContri	MR (%)
Topic 1	20.84	0.50	97.66
Topic 2	18.51	0.03	99.84
Topic 3	3.42	18.01	84.02
Topic 4	17.01	1.21	93.36
Topic 5	33.47	9.72	77.50
Topic 6	55.26	2.24	96.10
Topic 7	37.51	1.13	97.08
Topic 8	13.88	10.17	57.71
Topic 9	7.70	14.61	65.48
Topic 10	20.42	2.27	89.99
Topic 11	124.52	1.28	98.98
Topic 12	6.39	11.38	64.05
Topic 13	26.59	1.53	94.55
Topic 14	24.87	1.09	95.79
Topic 15	28.37	0.44	98.48
Topic 16	6.65	4.84	57.89
Topic 17	33.42	2.29	93.60
Topic 18	4.07	11.19	73.36
Topic 19	10.23	6.90	59.70
Topic 20	12.24	0.00	100.00
Average	—-	—-	84.76

8.4.1. Entries with Negative Values in the Term-Topic Matrix. In LDA, PLSI, and NMF, the probabilities or weights of terms are all non-negative. In RLSI, the weights of terms can be either positive or negative. In this experiment, we investigated the distributions of terms with positive weights and negative weights in the topics of RLSI.

We examined the "positive contribution" (PosContri), "negative contribution" (NegContri), and "majority ratio" (MR) of each topic created by batch RLSI and online RLSI. Here, the positive or negative contribution of a topic is defined as the sum of absolute weights of positive or negative terms in the topic, and the majority ratio of a topic is defined as the ratio of the dominant contribution, i.e., MR = max {PosContri, NegContri} / (PosContri + NegContri). A larger MR value reflects a larger gap between positive and negative contributions in the topic, indicating that the topic is "pure". Table XXIV and Table XXV show the results for batch RLSI and online RLSI, with the same parameter settings as in Section 8.2.2 (i.e., $K = 20$, $\lambda_1 = 0.5$ and $\lambda_2 = 1.0$) and Section 8.2.4 (i.e., $K = 20$, $\lambda_1 = 0.4$ and $\lambda_2 = 1.0$). From the results, we can see that 1) almost every RLSI topic is pure and the average MR value of topic is quite high; 2) in a topic, the positive contribution usually dominates; 3) online RLSI has a lower average MR than batch RLSI.

Table XXVI shows four example topics from Table XXIV. Among them, two are dominated by positive contributions (i.e., Topic 9 and Topic 17) and two are dominated by negative contributions (i.e., Topic 10 and Topic 20). For each topic, 20 terms as well as their weights are shown, 10 with the largest weights and the other 10 with the smallest weights. From the result, we can see that all the topics are readable if the dominant parts are taken, whether positive or negative.

8.4.2. Linear Combination of Topic and Term Matching Scores. In this experiment, we investigated how topic models such as RLSI and LDA can address the term mismatch problem when combined with the term-based matching models, e.g., BM25 (with default parameters $k_1 = 1.2$ and $b = 0.75$).

We take query "Weather Related Fatalities" (T-059) as an example. There are two documents, AP880502-0086 and AP880219-0053, associated with the query, and the first one is relevant and

Table XXVI. Example topics discovered by batch RLSI on AP.

Topic 9				Topic 10			
drug	(3.638)	party	(-0.120)	nuclear	(0.313)	**soviet**	(-2.735)
test	(0.942)	tax	(-0.112)	plant	(0.255)	**afghanistan**	(-1.039)
cocain	(0.716)	strike	(-0.085)	senate	(0.161)	**afghan**	(-1.032)
aid	(0.621)	elect	(-0.042)	reactor	(0.134)	**gorbachev**	(-0.705)
trafficker	(0.469)	court	(-0.038)	air	(0.127)	**pakistan**	(-0.680)
virus	(0.411)	opposite	(-0.012)	test	(0.115)	**guerrilla**	(-0.673)
infect	(0.351)	plant	(-0.012)	contra	(0.114)	**kabul**	(-0.582)
enforce	(0.307)	reform	(-0.011)	palestinian	(0.109)	**union**	(-0.512)
disease	(0.274)	polite	(-0.010)	safety	(0.084)	**moscow**	(-0.511)
patient	(0.258)	govern	(-0.002)	pentagon	(0.082)	**troop**	(-0.407)
Topic 17				Topic 20			
firefighter	(1.460)	plane	(-0.057)	soviet	(0.073)	**africa**	(-2.141)
acr	(1.375)	bomb	(-0.053)	crash	(0.057)	**south**	(-1.881)
forest	(1.147)	crash	(-0.051)	contra	(0.041)	**african**	(-1.357)
park	(0.909)	airline	(-0.048)	flight	(0.029)	**angola**	(-1.125)
blaze	(0.865)	party	(-0.043)	sandinista	(0.027)	**apartheid**	(-0.790)
yellowstone	(0.857)	police	(-0.040)	air	(0.026)	**black**	(-0.684)
fire	(0.773)	military	(-0.035)	plane	(0.020)	**botha**	(-0.601)
burn	(0.727)	govern	(-0.032)	investigate	(0.016)	**cuban**	(-0.532)
wind	(0.537)	flight	(-0.027)	program	(0.015)	**mandela**	(-0.493)
evacuate	(0.328)	elect	(-0.020)	airline	(0.010)	**namibia**	(-0.450)

Table XXVII. Judgments and matching scores of example query and documents.

QryID/DocID	Title/Head	Judgment	s_{term}	s_{topic}
T-059	Weather Related Fatalities	—	—	—
AP880502-0086	May Snowstorm Hits Rockies	Relevant	0	0.9434
AP880219-0053	Rain Heavy in South; Snow Scattered	Irrelevant	0	0.8438

Table XXVIII. Corresponding topics.

Topic 6	Topic 16	Topic 17
senate	police	firefighter
program	kill	acr
house	crash	forest
reagan	plane	park
state	air	blaze
congress	bomb	yellowstone
tax	attack	fire
budget	flight	burn
govern	army	wind
committee	soldier	evacuate

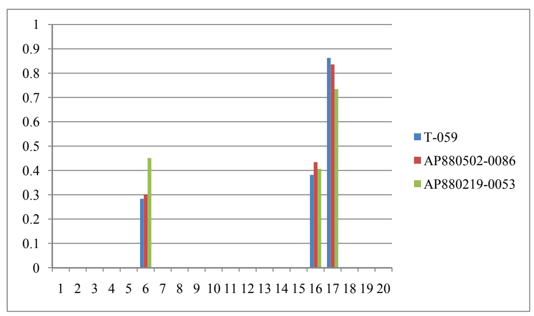


Fig. 9. Representations for sampled query and documents.

the second one is not. Table XXVII shows the titles of the two documents[18]. Neither document shares a term with the query, and thus the term-based matching scores (s_{term}) of them are both zero. In contrast, the matching scores of the two documents based on RLSI are large (i.e., 0.9434 and 0.8438), where parameters $K = 20$, $\lambda_1 = 0.5$, and $\lambda_2 = 1.0$. The topics of the RLSI model are those in Table XII. Figure 9 shows the representations of the query and the documents in the topic space. We can see that the query and the documents are mainly represented by the 6^{th}, 16^{th}, and 17^{th} topics. Table XXVIII shows the details of the three topics about the US government, accidents, and disasters, respectively[19]. We can judge that the representations are reasonable given the contents of the documents.

[18] The whole documents can be found in `http://www.daviddlewis.com/resources/testcollections/trecap/`.
[19] Note that the topics here are identical to those in Table XII, where top 10 instead of 5 terms are shown here.

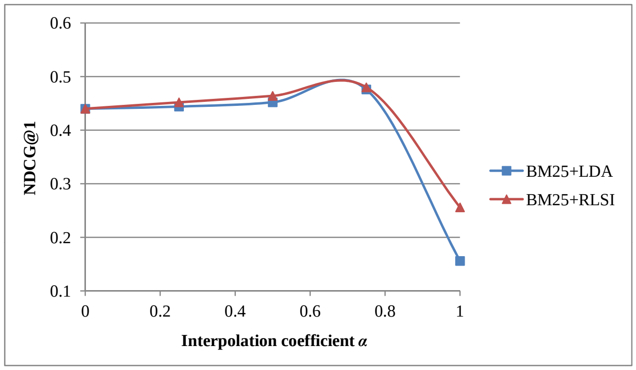


Fig. 10. Retrieval performances of linear combination with different interpolation coefficient values.

This example indicates that relevant documents that do not share terms with the query may still receive large scores through matching in the topic space. That is the reason that RLSI can address the term mismatch problem and improve retrieval performance. On the other hand, irrelevant documents that do not share terms with the query may also get some scores through the matching. That is to say, RLSI may occasionally hurt the retrieval performance because matching in the topic space can be coarse. Therefore, employing a combination of topic-based model and term-based model may leverage the advantages of both and significantly improve the overall retrieval performance. Similar phenomenon was observed in the study of LDA [Wei and Croft 2006], in which the authors suggested a combination of language model and LDA.

We examined how the retrieval performance of RLSI and LDA combined with BM25, denoted as "BM25+RLSI" and "BM25+LDA", changes when the interpolation coefficient α varies from 0 to 1. For both RLSI and LDA, the optimal parameters were used, as in Section 8.2.2 (i.e., $K = 50$, $\lambda_1 = 0.5$, and $\lambda_2 = 1.0$ for RLSI; $K = 50$ for LDA). Figure 10 shows the NDCG@1 scores of BM25+RLSI and BM25+LDA at different α values. Note that BM25+RLSI and BM25+LDA degenerate into RLSI and LDA respectively when $\alpha = 1$, and they degenerate into BM25 when $\alpha = 0$. From the result, we can see that 1) RLSI alone and LDA alone perform worse than BM25; 2) RLSI and LDA can significantly improve the overall retrieval performance when properly combined with BM25, i.e., with proper α values.

We further examined the precisions at position n ($p@n$) of three models, BM25 only (BM25), RLSI only (RLSI), and their linear combination (BM25+RLSI), when n increases from 1 to 50. Here, the optimal parameters of RLSI and the optimal interpolation coefficient were used, as in Section 8.2.2 (i.e., $K = 50$, $\lambda_1 = 0.5$, $\lambda_2 = 1.0$, and $\alpha = 0.75$). Figure 11 shows the precision curves of the three models at different positions. We also conducted the same experiment with BM25 only (BM25), LDA only (LDA), and their linear combination (BM25+LDA). Here, the optimal parameters of LDA and the optimal interpolation coefficient were used, as in Section 8.2.2 (i.e., $K = 50$ and $\alpha = 0.75$). The corresponding result is shown in Figure 12. From the results, we can see that 1) BM25 performs quite well when n is small, and its performance drops rapidly as n increases; 2) neither RLSI alone nor LDA alone performs well when n takes different values; 3) RLSI alone as well as LDA alone perform even worse than BM25; 4) BM25+RLSI outperforms both BM25 and RLSI, and BM25+LDA outperforms both BM25 and LDA, particularly when n is small; 5) BM25+RLSI performs better than BM25+LDA. We can conclude that: 1) term matching and topic matching are complementary; 2) the most relevant documents are relevant (have high scores) from both the viewpoints of term matching and topic matching. That is to say, combining topic-based matching models with term-based matching models is effective for enhancing the overall retrieval performance.

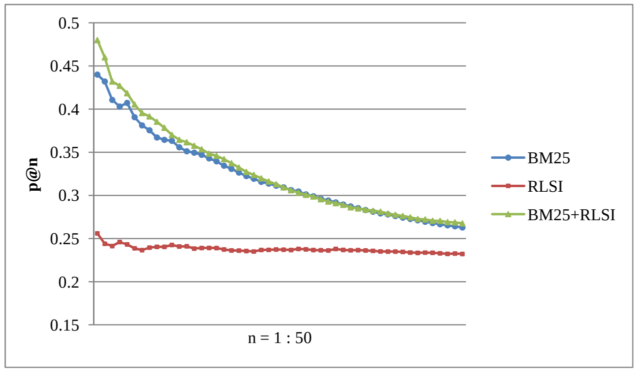


Fig. 11. Precisions at different positions $p@n$.

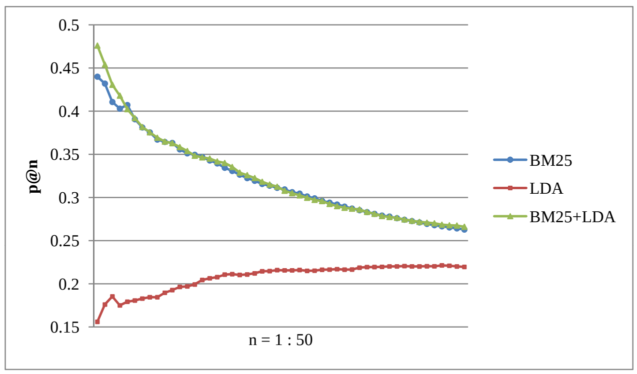


Fig. 12. Precisions at different positions $p@n$.

8.4.3. BM25 with fine-tuned parameters as baseline. In this experiment, we investigated how topic models such as LSI, PLSI, LDA, NMF, and RLSI behave when combined with fine-tuned BM25.

First, to tune the parameters of BM25, we set k_1 from 1.2 to 2.0 in steps of 0.1, and b from 0.5 to 1 in steps of 0.05. We found that BM25 with $k_1 = 1.5$ and $b = 0.5$ performs best (measured by NDCG@1). Then, we combined topic models LSI, PLSI, LDA, NMF, and RLSI with the best-performing BM25 model, denoted as “BM25+LSI”, “BM25+PLSI”, “BM25+LDA”, “BM25+NMF”, and “BM25+RLSI”, and tested their retrieval performances. The experimental setting was the same as that in Section 8.2.2, i.e., parameter K was set in range of $[10, 50]$, interpolation coefficient α was set from 0 to 1 in steps of 0.05, and λ_2 was fixed to 1 and λ_1 was set in range of $[0.1, 1]$ in RLSI. Tables XXIX shows the results achieved by the best parameter setting (measured by NDCG@1) on AP. Stars indicate significant improvements on the baseline method, i.e., the best-performing BM25, according to one-sided t-test (p-value < 0.05). From the results, we can see that 1) when combined with a fine-tuned term-based matching model, topic-based matching models can still significantly improve the retrieval performance; 2) RLSI performs equally well compared with the other topic models, which is the same trend as in Section 8.2.2. We also conducted the same experiments on WSJ and OHSUMED and obtained similar results.

9. CONCLUSIONS

In this paper, we have studied topic modeling from the viewpoint of enhancing scalability. We have proposed a new method for topic modeling, called Regularized Latent Semantic Indexing (RLSI). RLSI formalizes topic modeling as minimization of a quadratic loss function with a regularization (either ℓ_1 or ℓ_2 norm). Two versions of RLSI have been given, namely batch mode and online

Table XXIX. Retrieval performance of topic models combined with fine-tuned BM25.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3983	0.4760	0.4465	0.4391	0.4375
BM25+LSI	0.4005	0.4880	0.4500	0.4430	0.4405
BM25+PLSI	0.4000	0.4880	**0.4599** *	0.4510 *	0.4452 *
BM25+LDA	0.3985	0.4960 *	0.4577 *	0.4484	0.4453
BM25+NMF	**0.4021** *	0.4880	0.4504	0.4465	0.4421
BM25+RLSI	0.4002	**0.5000** *	0.4585 *	**0.4535** *	**0.4502** *

mode. Although similar techniques have been used in other fields, such as sparse coding in computer vision, this is the first comprehensive study of regularization for topic modeling, as far as we know. The formulation of RLSI makes its optimization process decomposable, and thus scalable. Specifically, RLSI replaces the orthogonality constraint or probability distribution constraint with regularization. Therefore, RLSI can be more easily implemented in a parallel and/or distributed computing environment, such as MapReduce.

In our experiments on topic discovery and relevance ranking, we have tested different variants of RLSI and confirmed that the sparse topic regularization and smooth document regularization is the best choice from the viewpoint of overall performance. Specifically the ℓ_1 norm on topics (making topics sparse) and ℓ_2 norm on document representations gave the best readability and retrieval performance. We have also confirmed that both batch RLSI and online RLSI can work almost equally well. In our experiments on topic detection and tracking, we have verified that online RLSI can effectively capture the evolution of the topics over time.

Experimental results on TREC data and large scale web data show that RLSI is better than or comparable with existing methods such as LSI, PLSI, and LDA in terms of readability of topics and accuracy in relevance ranking. We have also demonstrated that RLSI can scale up to large document collection with 1.6 million documents and 7 million terms, which is very difficult for existing methods. Most previous work reduced the input vocabulary size to tens of thousands of terms, which has been demonstrated to hurt the ranking accuracy.

As future work, we plan to further enhance the performance of online RLSI. More specifically, we try to develop better online RLSI algorithms which can not only save memory but also save computation cost. We make comparison of the online RLSI algorithms with other online topic modeling algorithms such as [Hoffman et al. 2010; Mimno et al. 2012]. We also want to enhance the scale of experiments to process even larger datasets, and further study the theoretical properties of RLSI and other applications of RLSI, both batch version and online version.

Acknowledgments

We would like to thank Xinjing Wang of MSRA for helpful discussions and the anonymous reviewers for their valuable comments.

REFERENCES

ALLAN, J., CARBONELL, J., DODDINGTON, G., YAMRON, J., AND YANG, Y. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*.

ALSUMAIT, L., BARBARA, D., AND DOMENICONI, C. 2008. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Proceedings of the IEEE International Conference on Data Mining*.

ASUNCION, A., SMYTH, P., AND WELLING, M. 2011. Asynchronous distributed estimation of topic models for document analysis. *Statistical Methodology*.

ATREYA, A. AND ELKAN, C. 2010. Latent semantic indexing (lsi) fails for trec collections. *ACM SIGKDD Explorations Newsletter 12*.

BERTSEKAS, D. P. 1999. *Nonlinear Programming*. Athena Scientific Belmont.

BLEI, D. 2011. Introduction to probabilistic topic models. *Communications of the ACM*, to appear.

BLEI, D. AND LAFFERTY, J. 2009. Topic models. *Text Mining: Classification, Clustering, and Applications*.

BLEI, D., NG, A. Y., AND JORDAN, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research 3*.

BLEI, D. M. AND LAFFERTY, J. D. 2006. Dynamic topic models. In *Proceedings of the International Conference on Machine Learning*.

BONNANS, J. F. AND SHAPIRO, A. 1998. Optimization problems with perturbations: A guided tour. *SIAM Review 40*.

BOTTOU, L. AND BOUSQUET, O. 2008. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*.

BULUC, A. AND GILBERT, J. R. 2008. Challenges and advances in parallel sparse matrix-matrix multiplication. In *Proceedings of the International Conference on Parallel Processing*.

BURGES, C. J., RAGNO, R., AND LE, Q. V. 2007. Learning to rank with nonsmooth cost functions. In *Advances in Neural Information Processing Systems*.

CHAIKEN, R., JENKINS, B., LARSON, P.-A., RAMSEY, B., SHAKIB, D., WEAVER, S., AND ZHOU, J. 2008. Scope: Easy and efficient parallel processing of massive data sets. *Very Large Data Base Endowment 1*.

CHEN, S. S., DONOHO, D. L., AND SAUNDERS, M. A. 1998. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing 20*.

CHEN, X., BAI, B., QI, Y., LIN, Q., AND CARBONELL, J. 2010. Sparse latent semantic analysis. In *Workshop on Neural Information Processing Systems*.

DEAN, J., GHEMAWAT, S., AND INC, G. 2004. Mapreduce: simplified data processing on large clusters. In *Proceedings of USENIX Symposium on Operating Systems Design and Implementation*.

DEERWESTER, S., DUMAIS, S. T., FURNAS, G. W., LANDAUER, T. K., AND HARSHMAN, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science 41*.

DING, C., LI, T., AND PENG, W. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing semantic indexing. *Computational Statistics and Data Analysis 52*.

DING, C. H. Q. 2005. A probabilistic model for latent semantic indexing. *Journal of the American Society for Information Science and Technology 56*.

EFRON, B., HASTIE, T., JOHNSTONE, I., AND TIBSHIRANI, R. 2004. Least angle regression. *Annals of Statistics 32*.

FRIEDMAN, J., HASTIE, T., HOFLING, H., AND TIBSHIRANI, R. 2007. Pathwise coordinate optimization. *Annals of Applied Statistics 1*.

FU, W. J. 1998. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics 7*.

HOFFMAN, M. D., BLEI, D. M., AND BACH, F. 2010. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*.

HOFMANN, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the international ACM SIGIR conference on Research and Development in Information Retrieval*.

KONTOSTATHIS, A. 2007. Essential dimensions of latent semantic indexing (lsi). In *Proceedings of the 40th Hawaii International International Conference on Systems Science*.

LEE, D. D. AND SEUNG, H. S. 1999. Learning the parts of objects with nonnegative matrix factorization. *Nature 401*.

LEE, D. D. AND SEUNG, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*.

LEE, H., BATTLE, A., RAINA, R., AND NG, A. Y. 2007. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*.

LIANG, P. AND KLEIN, D. 2009. Online em for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

LIU, C., CHIH YANG, H., FAN, J., HE, L.-W., AND WANG, Y.-M. 2010. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce. In *Proceedings of International World Wide Web Conference*.

LIU, Z., ZHANG, Y., AND CHANG, E. Y. 2011. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing. *ACM Transactions on Intelligent Systems and Technology 2*.

LU, Y., MEI, Q., AND ZHAI, C. 2011. Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. *Information Retrieval 14*.

MAIRAL, J., BACH, F., PONCE, J., SAPIRO, G., AND ZISSERMAN, A. 2009. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*.

MAIRAL, J., BACH, F., SUPRIEURE, E. N., AND SAPIRO, G. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research 11*.

MIMNO, D., HOFFMAN, M. D., AND BLEI, D. M. 2012. Sparse stochastic inference for latent dirichlet allocation. In *Proceedings of the 29th International Conference on Machine Learning*.

MIMNO, D. M. AND MCCALLUM. 2007. Organizing the oca: Learning faceted subjects from a library of digital books. In *Proceedings of the Joint Conference on Digital Libraries*.

NEAL, R. M. AND HINTON, G. E. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models 89*.

Newman, D., Asuncion, A., Smyth, P., and Welling, M. 2008. Distributed inference for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*.

Olshausen, B. A. and Fieldt, D. J. 1997. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research 37*.

Osborne, M., Presnell, B., and Turlach, B. 2000. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*.

Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M., and Gatford, M. 1994. Okapi at trec-3. In *Proceedigns of the 3rd Text REtrieval Conference*.

Rubinstein, R., Zibulevsky, M., and Elad, M. 2008. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*.

Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM 18*.

Shashanka, M., Raj, B., and Smaragdis, P. 2007. Sparse overcomplete latent variable decomposition of counts data. In *Advances in Neural Information Processing Systems*.

Singh, A. P. and Gordon, G. J. 2008. A unified view of matrix factorization models. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*.

Smola, A. and Narayanamurthy, S. 2010. An architecture for parallel topic models. *Proceedings of the VLDB Endowment 3*.

Thakur, R. and Rabenseifner, R. 2005. Optimization of collective communication operations in mpich. *International Journal of High Performance Computing 19*.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*.

Wang, C. and Blei, D. M. 2009. Decoupling sparsity and smoothness in the discrete hierachical dirichlet process. In *Advances in Neural Information Processing Systems*.

Wang, Q., Xu, J., Li, H., and Craswell, N. 2011. Regularized latent semantic indexing. In *Proceedings of the international ACM SIGIR conference on Research and Development in Information Retrieval*.

Wang, X. and McCallum, A. 2006. Topics over time: A non-markov continuous-time model of topical trends. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Wang, Y., Bai, H., Stanton, M., yen Chen, W., and Chang, E. Y. 2009. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Proceedings of the International Conference on Algorithmic Aspects of Information and Management*.

Wei, X. and Croft, B. W. 2006. Lda-based document models for ad-hoc retrieval. In *Proceedings of the international ACM SIGIR conference on Research and Development in Information Retrieval*.

Yan, F., Xu, N., and Qi, Y. A. 2009. Parallel inference for latent dirichlet allocation on graphics processing units. In *Advances in Neural Information Processing Systems*.

Yi, X. and Allan, J. 2009. A comparative study of utilizing topic models for information retrieval. In *Proceedings of the 31st European Conference on IR Research*.

Zhu, J. and Xing, E. P. 2011. Sparse topical coding. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

APPENDIX

In this section we provide the proof of Proposition 5.5. Before that, we give and prove several lemmas.

Lemma A.1. *Let $f : \mathbb{R} \to \mathbb{R}, f(x) = ax^2 - 2bx + \lambda|x|$ with $a > 0$ and $\lambda > 0$. Let x^* denote the minimum of $f(x)$. Then,*

$$x^* = \frac{\left(|b| - \frac{1}{2}\lambda\right)_+ sign(b)}{a}, \tag{10}$$

where $(\cdot)_+$ denotes the hinge function. Moreover, $f(x) \geq f(x^*) + a\,(x - x^*)^2$ holds for all $x \in \mathbb{R}$.

Proof. Note that

$$f(x) = \begin{cases} ax^2 - (2b - \lambda)x, & \text{if } x \geq 0, \\ ax^2 - (2b + \lambda)x, & \text{if } x \leq 0, \end{cases}$$

which can be minimized in the following three cases. First, if $b > \frac{1}{2}\lambda$, we obtain

$$x^* = \left(b - \tfrac{1}{2}\lambda\right)\Big/a, \quad f\left(x^*\right) = -\left(b - \tfrac{1}{2}\lambda\right)^2\Big/a$$

by using $\min_{x \geq 0} f(x) = f(x^*) \leq 0$ and $\min_{x \leq 0} f(x) = f(0) = 0$. Second, if $b < -\frac{1}{2}\lambda$, we obtain

$$x^* = \left(b + \tfrac{1}{2}\lambda\right) \Big/ a, \quad f(x^*) = -\left(b + \tfrac{1}{2}\lambda\right)^2 \Big/ a$$

by using $\min_{x \geq 0} f(x) = f(0) = 0$ and $\min_{x \leq 0} f(x) = f(x^*) \leq 0$. Finally, we can easily get $f(x^*) = 0$ with $x^* = 0$, if $|b| \leq \frac{1}{2}\lambda$, since $\min_{x \geq 0} f(x) = f(0) = 0$ and $\min_{x \leq 0} f(x) = f(0) = 0$. To conclude, we have

$$x^* = \begin{cases} \frac{b - \frac{1}{2}\lambda}{a}, & \text{if } b > \frac{1}{2}\lambda, \\ \frac{b + \frac{1}{2}\lambda}{a}, & \text{if } b < -\frac{1}{2}\lambda, \\ 0, & \text{if } |b| \leq \frac{1}{2}\lambda, \end{cases}$$

which is equivalent to Eq. (10). Moreover,

$$f(x^*) = \begin{cases} -\frac{\left(b - \frac{1}{2}\lambda\right)^2}{a}, & \text{if } b > \frac{1}{2}\lambda, \\ -\frac{\left(b + \frac{1}{2}\lambda\right)^2}{a}, & \text{if } b < -\frac{1}{2}\lambda, \\ 0, & \text{if } |b| \leq \frac{1}{2}\lambda. \end{cases}$$

Next, we consider function $\Delta(x) = f(x) - f(x^*) - a(x - x^*)^2$. A short calculation shows that

$$\Delta(x) = \begin{cases} \lambda|x| - \lambda x, & \text{if } b > \frac{1}{2}\lambda, \\ \lambda|x| + \lambda x, & \text{if } b < -\frac{1}{2}\lambda, \\ \lambda|x| - 2bx, & \text{if } |b| \leq \frac{1}{2}\lambda. \end{cases}$$

Note that $|x| \geq x$, $|x| \geq -x$, and $\lambda \geq 2b$ when $|b| \leq \frac{1}{2}\lambda$. Thus, we obtain $\Delta(x) \geq 0$ for all $x \in \mathbb{R}$, which gives us the desired result. □

LEMMA A.2. *Consider the following optimization problem:*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^K} f(\boldsymbol{\beta}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1,$$

where $\boldsymbol{y} \in \mathbb{R}^N$ *is a real vector,* $\mathbf{X} \in \mathbb{R}^{N \times K}$ *is an* $N \times K$ *real matrix such that all the diagonal entries of matrix* $\mathbf{X}^T\mathbf{X}$ *are larger than zero, and* $\lambda > 0$ *is a parameter. For any* $\boldsymbol{\beta}^{(0)} \in \mathbb{R}^K$*, take* $\boldsymbol{\beta}^{(0)}$ *as the initial value and minimize* $f(\boldsymbol{\beta})$ *with respect to one entry of* $\boldsymbol{\beta}$ *while keep the others fixed (i.e., minimizing with respect to* $\beta_1, \cdots, \beta_K$ *in turn). After one round of such iterative minimization, we obtain* $\boldsymbol{\beta}^{(1)} \in \mathbb{R}^K$ *such that*

$$f\left(\boldsymbol{\beta}^{(0)}\right) - f\left(\boldsymbol{\beta}^{(1)}\right) \geq \kappa_2 \left\|\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta}^{(1)}\right\|_2^2 \tag{11}$$

with a constant $\kappa_2 > 0$*. Moreover, we obtain* $\boldsymbol{\beta}^{(T)} \in \mathbb{R}^K$ *such that*

$$f\left(\boldsymbol{\beta}^{(0)}\right) - f\left(\boldsymbol{\beta}^{(T)}\right) \geq \frac{\kappa_2}{T} \left\|\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta}^{(T)}\right\|_2^2 \tag{12}$$

after T *rounds of such iterative minimization.*

PROOF. Define $\boldsymbol{\beta}_j^{(0)} \in \mathbb{R}^K$ as $\boldsymbol{\beta}_j^{(0)} = \left(\beta_1^{(1)}, \cdots, \beta_j^{(1)}, \beta_{j+1}^{(0)}, \cdots, \beta_K^{(0)}\right)^T$ for $j = 1, \cdots, K-1$, where $\beta_j^{(0)}$ is the j^{th} entry of $\boldsymbol{\beta}^{(0)}$ and $\beta_j^{(1)}$ is the j^{th} entry of $\boldsymbol{\beta}^{(1)}$. By defining $\boldsymbol{\beta}_0^{(0)} = \boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\beta}_K^{(0)} = \boldsymbol{\beta}^{(1)}$, it is easy to see that starting from $\boldsymbol{\beta}_{j-1}^{(0)}$, minimizing $f(\boldsymbol{\beta})$ with respect to β_j (i.e., the j^{th} entry of $\boldsymbol{\beta}$) leads us to $\boldsymbol{\beta}_j^{(0)}$ for $j = 1, \cdots, K$. After one round of such iterative minimization, we move from $\boldsymbol{\beta}^{(0)}$ to $\boldsymbol{\beta}^{(1)}$.

Consider minimizing $f(\boldsymbol{\beta})$ with respect to β_j. Let $\boldsymbol{\beta}_{\setminus j}$ denote the vector of $\boldsymbol{\beta}$ with the j^{th} entry removed, $\boldsymbol{x}_j$ denote the j^{th} column of $\mathbf{X}$, and $\mathbf{X}_{\setminus j}$ denote the matrix of $\mathbf{X}$ with the j^{th} column removed.

Rewrite $f(\boldsymbol{\beta})$ as a function respect to β_j, and we obtain

$$f(\boldsymbol{\beta}) = \left\|\boldsymbol{x}_j\right\|_2^2 \beta_j^2 - 2\boldsymbol{x}_j^T \left(\boldsymbol{y} - \mathbf{X}_{\backslash j}\boldsymbol{\beta}_{\backslash j}\right)\beta_j + \lambda \left|\beta_j\right| + const,$$

where *const* is a constant with respect to β_j. Let $\kappa_2 = \min\left\{\|\boldsymbol{x}_1\|_2^2, \cdots, \|\boldsymbol{x}_K\|_2^2\right\}$. The second conclusion of Lemma A.1 indicates that

$$f\left(\boldsymbol{\beta}_{j-1}^{(0)}\right) - f\left(\boldsymbol{\beta}_j^{(0)}\right) \geq \left\|\boldsymbol{x}_j\right\|_2^2 \left(\beta_j^{(0)} - \beta_j^{(1)}\right)^2 \geq \kappa_2 \left(\beta_j^{(0)} - \beta_j^{(1)}\right)^2$$

for $j = 1, \cdots, K$. Summing over the K inequalities, we obtain the first part of the theorem Eq. (11) by noting that $\boldsymbol{\beta}_0^{(0)} = \boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\beta}_K^{(0)} = \boldsymbol{\beta}^{(1)}$. Here $\kappa_2 > 0$ holds since all the diagonal entries of matrix $\mathbf{X}^T\mathbf{X}$ are larger than zero.

The second part is easy to prove. First, the first part indicates that

$$f\left(\boldsymbol{\beta}^{(0)}\right) - f\left(\boldsymbol{\beta}^{(T)}\right) = \sum_{t=1}^{T} f\left(\boldsymbol{\beta}^{(t-1)}\right) - f\left(\boldsymbol{\beta}^{(t)}\right) \geq \kappa_2 \sum_{t=1}^{T} \left\|\boldsymbol{\beta}^{(t-1)} - \boldsymbol{\beta}^{(t)}\right\|_2^2.$$

Furthermore, the triangle inequality of Euclidean distance (ℓ_2-norm distance) leads to

$$\begin{aligned}
\left\|\boldsymbol{\beta}^{(0)} - \boldsymbol{\beta}^{(T)}\right\|_2^2 =& \sum_{i=1}^{T}\sum_{j=1}^{T} \left(\boldsymbol{\beta}^{(i-1)} - \boldsymbol{\beta}^{(i)}\right)^T \left(\boldsymbol{\beta}^{(j-1)} - \boldsymbol{\beta}^{(j)}\right) \\
\leq& \frac{1}{2}\sum_{i=1}^{T}\sum_{j=1}^{T} \left(\left\|\boldsymbol{\beta}^{(i-1)} - \boldsymbol{\beta}^{(i)}\right\|_2^2 + \left\|\boldsymbol{\beta}^{(j-1)} - \boldsymbol{\beta}^{(j)}\right\|_2^2\right) \\
=& T\sum_{t=1}^{T} \left\|\boldsymbol{\beta}^{(t-1)} - \boldsymbol{\beta}^{(t)}\right\|_2^2.
\end{aligned}$$

From these two inequalities, we obtain the second part Eq. (12). □

LEMMA A.3. *Let* $\boldsymbol{v}^* = \left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-1}\mathbf{U}^T\boldsymbol{d}$, *and Assumptions 5.1 and 5.2 hold. Then,* $\|\boldsymbol{v}^*\|_2^2 \leq \delta_1^2/4\lambda_2$ *holds for all* $\boldsymbol{d} \in \mathcal{K}$ *and* $\mathbf{U} \in \mathcal{U}$.

PROOF. Without loss of generality, we suppose that $M \geq K$. Suppose that the SVD of $\mathbf{U}$ has the form $\mathbf{U} = \mathbf{P\Omega Q}^T$, where $\mathbf{P} \in \mathbb{R}^{M\times M}$ and $\mathbf{Q} \in \mathbb{R}^{K\times K}$ are orthogonal matrices, and $\mathbf{\Omega} \in \mathbb{R}^{M\times K}$ is a diagonal matrix with diagonal entries $\omega_{11} \geq \omega_{22} \geq \cdots \geq \omega_{KK} \geq 0$. Computing the squared ℓ_2-norm of $\boldsymbol{v}^*$, we get

$$\begin{aligned}
\|\boldsymbol{v}^*\|_2^2 =& \boldsymbol{d}^T\mathbf{U}\left(\mathbf{U}^T\mathbf{U} + \lambda_2\mathbf{I}\right)^{-2}\mathbf{U}^T\boldsymbol{d} \\
=& \boldsymbol{d}^T\mathbf{P\Omega}\left(\mathbf{\Omega}^T\mathbf{\Omega} + \lambda_2\mathbf{I}\right)^{-2}\mathbf{\Omega}^T\mathbf{P}^T\boldsymbol{d} \\
=& \sum_{k=1}^{K} \boldsymbol{d}^T\boldsymbol{p}_k \frac{\omega_{kk}^2}{\left(\omega_{kk}^2 + \lambda_2\right)^2}\boldsymbol{p}_k^T\boldsymbol{d},
\end{aligned}$$

where $\boldsymbol{p}_k \in \mathbb{R}^M$ is the k^{th} column of $\mathbf{P}$. By noting that $\omega_{kk}^2 \Big/ \left(\omega_{kk}^2 + \lambda_2\right)^2 \leq 1/4\lambda_2$ holds for $k = 1, \cdots, K$, it is easy to show that

$$\|\boldsymbol{v}^*\|_2^2 \leq \frac{1}{4\lambda_2}\boldsymbol{d}^T\left(\sum_{k=1}^{K}\boldsymbol{p}_k\boldsymbol{p}_k^T\right)\boldsymbol{d} = \frac{1}{4\lambda_2}\|\boldsymbol{d}\|_2^2 - \frac{1}{4\lambda_2}\sum_{i=K+1}^{M}\left(\boldsymbol{d}^T\boldsymbol{p}_i\right)^2 \leq \frac{\delta_1^2}{4\lambda_2},$$

where we use the fact that $\mathbf{I} = \mathbf{PP}^T = \sum_{m=1}^{M}\boldsymbol{p}_m\boldsymbol{p}_m^T$. □

LEMMA A.4. *Let $\hat{f}_t$ denote the loss defined in Eq. (6), and Assumptions 5.1 and 5.2 hold. Then, $\hat{f}_t - \hat{f}_{t+1}$ is Lipschitz with constant $L_t = \frac{1}{t+1}\left(\frac{\delta_1^2\delta_2}{\lambda_2} + \frac{2\delta_1^2}{\sqrt{\lambda_2}}\right)$.*

PROOF. A short calculation shows that

$$\hat{f}_t - \hat{f}_{t+1} = \frac{1}{t+1}\left[\frac{1}{t}\sum_{i=1}^{t}\left(\|\boldsymbol{d}_i - \mathbf{U}\boldsymbol{v}_i\|_2^2 + \lambda_2\|\boldsymbol{v}_i\|_2^2\right) - \left(\|\boldsymbol{d}_{t+1} - \mathbf{U}\boldsymbol{v}_{t+1}\|_2^2 + \lambda_2\|\boldsymbol{v}_{t+1}\|_2^2\right)\right],$$

whose gradient can be calculated as

$$\nabla_{\mathbf{U}}\left(\hat{f}_t - \hat{f}_{t+1}\right) = \frac{2}{t+1}\left[\mathbf{U}\left(\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{v}_i\boldsymbol{v}_i^T - \boldsymbol{v}_{t+1}\boldsymbol{v}_{t+1}^T\right) - \left(\frac{1}{t}\sum_{i=1}^{t}\boldsymbol{d}_i\boldsymbol{v}_i^T - \boldsymbol{d}_{t+1}\boldsymbol{v}_{t+1}^T\right)\right].$$

To prove Lipschitz continuity, we consider the Frobenius norm of the gradient, obtaining the following bound:

$$\begin{aligned}\left\|\nabla_{\mathbf{U}}\left(\hat{f}_t - \hat{f}_{t+1}\right)\right\|_F \leq & \frac{2}{t+1}\left[\|\mathbf{U}\|_F\left(\frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{v}_i\|_2^2 + \|\boldsymbol{v}_{t+1}\|_2^2\right) + \left(\frac{1}{t}\sum_{i=1}^{t}\|\boldsymbol{d}_i\|_2\|\boldsymbol{v}_i\|_2 + \|\boldsymbol{d}_{t+1}\|_2\|\boldsymbol{v}_{t+1}\|_2\right)\right] \\ \leq & \frac{1}{t+1}\left(\frac{\delta_1^2\delta_2}{\lambda_2} + \frac{2\delta_1^2}{\sqrt{\lambda_2}}\right),\end{aligned}$$

where we use Assumption 5.1, Assumption 5.2, and Lemma A.3. Then, the mean value theorem gives the desired results. □

PROOF OF PROPOSITION 5.5. This proof is partially inspired by [Bonnans and Shapiro 1998; Mairal et al. 2010]. Let

$$g_m(\bar{\boldsymbol{u}}) = \left\|\bar{\boldsymbol{d}}_m^{(t)} - \mathbf{V}_t^T\bar{\boldsymbol{u}}\right\|_2^2 + \theta t\|\bar{\boldsymbol{u}}\|_1$$

denote the objective function in Eq. (7). With Assumption 5.3, starting from $\bar{\boldsymbol{u}}_m^{(t+1)}$, optimization problem Eq. (7) reaches its minimum $\bar{\boldsymbol{u}}_m^{(t)}$ after at most T rounds of iterative minimization, where $\bar{\boldsymbol{u}}_m^{(t)}$ and $\bar{\boldsymbol{u}}_m^{(t+1)}$ are the column vectors whose entries are those of the m^{th} row of $\mathbf{U}_t$ and $\mathbf{U}_{t+1}$ respectively. Lemma A.2 applies, and

$$g_m\left(\bar{\boldsymbol{u}}_m^{(t+1)}\right) - g_m\left(\bar{\boldsymbol{u}}_m^{(t)}\right) \geq \frac{\kappa_3}{T}\left\|\bar{\boldsymbol{u}}_m^{(t+1)} - \bar{\boldsymbol{u}}_m^{(t)}\right\|_2^2$$

for $m = 1, \cdots, M$, where κ_3 is the smallest diagonal entry of $\mathbf{S}_t$. Summing over the M inequalities and using Assumption 5.4, we obtain

$$\hat{f}_t(\mathbf{U}_{t+1}) - \hat{f}_t(\mathbf{U}_t) \geq \frac{\kappa_1}{T}\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F^2. \tag{13}$$

Moreover,

$$\begin{aligned}\hat{f}_t(\mathbf{U}_{t+1}) - \hat{f}_t(\mathbf{U}_t) =& \hat{f}_t(\mathbf{U}_{t+1}) - \hat{f}_{t+1}(\mathbf{U}_{t+1}) + \hat{f}_{t+1}(\mathbf{U}_{t+1}) - \hat{f}_{t+1}(\mathbf{U}_t) + \hat{f}_{t+1}(\mathbf{U}_t) - \hat{f}_t(\mathbf{U}_t) \\ \leq& \hat{f}_t(\mathbf{U}_{t+1}) - \hat{f}_{t+1}(\mathbf{U}_{t+1}) + \hat{f}_{t+1}(\mathbf{U}_t) - \hat{f}_t(\mathbf{U}_t),\end{aligned}$$

where $\hat{f}_{t+1}(\mathbf{U}_{t+1}) - \hat{f}_{t+1}(\mathbf{U}_t) \leq 0$ since $\mathbf{U}_{t+1}$ minimizes $\hat{f}_{t+1}$. Given Assumptions 5.1 and 5.2, Lemma A.4 indicates that $\hat{f}_t - \hat{f}_{t+1}$ is Lipschitz with constant $L_t = \frac{1}{t+1}\left(\frac{\delta_1^2\delta_2}{\lambda_2} + \frac{2\delta_1^2}{\sqrt{\lambda_2}}\right)$, which leads to

$$\hat{f}_t(\mathbf{U}_{t+1}) - \hat{f}_t(\mathbf{U}_t) \leq \frac{1}{t+1}\left(\frac{\delta_1^2\delta_2}{\lambda_2} + \frac{2\delta_1^2}{\sqrt{\lambda_2}}\right)\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F. \tag{14}$$

From Eq. (13) and (14), we get the desired result, i.e., Eq. (8). □