Group Matrix Factorization for Scalable Topic Modeling

Quan Wang
MOE-Microsoft Key Laboratory of
Statistics&Information Technology
Peking University
v-quwan@microsoft.com

Zheng Cao
Dept. of Computer
Science & Engineering
Shanghai JiaoTong University
mozeda@sjtu.edu.cn

Jun Xu, Hang Li
Microsoft Research Asia
No. 5 Danling Street
Beijing, China
{junxu,hangli}@microsoft.com

ABSTRACT

Topic modeling can reveal the latent structure of text data and is useful for knowledge discovery, search relevance ranking, document classification, and so on. One of the major challenges in topic modeling is to deal with large datasets and large numbers of topics in real-world applications. In this paper, we investigate techniques for scaling up the non-probabilistic topic modeling approaches such as RLSI and NMF. We propose a general topic modeling method, referred to as Group Matrix Factorization (GMF), to enhance the scalability and efficiency of the non-probabilistic approaches. GMF assumes that the text documents have already been categorized into multiple semantic classes, and there exist class-specific topics for each of the classes as well as shared topics across all classes. Topic modeling is then formalized as a problem of minimizing a general objective function with regularizations and/or constraints on the class-specific topics and shared topics. In this way, the learning of class-specific topics can be conducted in parallel, and thus the scalability and efficiency can be greatly improved. We apply GMF to RLSI and NMF, obtaining Group RLSI (GRLSI) and Group NMF (GNMF) respectively. Experiments on a Wikipedia dataset and a real-world web dataset, each containing about 3 million documents, show that GRLSI and GNMF can greatly improve RLSI and NMF in terms of scalability and efficiency. The topics discovered by GRLSI and GNMF are coherent and have good readability. Further experiments on a search relevance dataset, containing 30,000 labeled queries, show that the use of topics learned by GRLSI and GNMF can significantly improve search relevance.

Categories and Subject Descriptors: H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing

General Terms: Algorithms, Experimentation

Keywords: Matrix Factorization, Topic Modeling, Large Scale

1. INTRODUCTION

Topic modeling refers to machine learning technologies whose aim is to discover the hidden semantic structure existing in a large collection of text documents. Given a collection of text documents, a topic model represents the relationship between the terms and the 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. Many topic modeling methods, such as Latent Semantic Indexing (LSI) [7], Probabilistic Latent Semantic Indexing (PLSI) [11], Latent Dirichlet Allocation (LDA) [5], Regularized Latent Semantic Indexing (RLSI) [26], and Non-negative Matrix Factorization (NMF) [13, 14] have been proposed and successfully applied to different applications in text mining, information retrieval, natural language processing, and other related fields.

One of the main challenges in topic modeling is to handle large numbers of documents and create large numbers of topics. For the probabilistic topic models 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 applications, this problem becomes particularly severe. For the non-probabilistic methods of NMF and RLSI, the formulation makes it possible to decompose the learning problem into multiple sub-problems and conduct learning in parallel, and hence in general they have better scalability than the probabilistic methods[1]. Refer to [26] for detailed discussions.

The high scalability of non-probabilistic methods makes them easier to be employed in practice. However, to handle millions or even billions of documents, it is still necessary to further improve their scalability and efficiency. In this paper, we investigate the possibilities of further enhancing the scalability and efficiency of non-probabilistic methods such as RLSI and NMF.

The method, called Group Matrix Factorization (GMF), assumes that the documents have already been categorized into multiple classes in a predefined taxonomy. This assumption is practical and common in many real-world applications. For example, Wikipedia data contains a hierarchical taxonomy with 25 classes at the first layer. Each Wikipedia article falls into at least one of the classes. The ODP project[2] provides a taxonomy of semantic classes and about 4 million web pages manually classified into the classes. The data can be used for training a classifier and other webpages can be classified into the classes by the classifier [2]. GMF further assumes that there exists a set of class-specific topics for each of the classes, and there also exists a set of shared topics for all of the classes. Each document in the collection is specified by its classes, class-specific topics, as well as shared topics. In this way, the large-scale learning problem can be decomposed into small-scale subproblems. We refer to the strategy as the divide-and-conquer technique.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
Copyright 2012 ACM 978-1-4503-1472-5/12/08 ...$15.00.

[1] Note that LSI needs to be solved by SVD due to its orthogonality assumption and thus it is hard to be scaled up.

[2] http://www.dmoz.org/

In GMF, the documents in each of the classes are represented as a term-document matrix. The term-document matrix is then approximated as the product of two matrices: one matrix represents the shared topics as well as the class-specific topics, and the other matrix represents the document representations based on the topics. An objective function is defined to measure the goodness of prediction of the data with the model. Optimization of the objective function leads to the automatic discovery of topics as well as topic representations of the documents.

We show that GMF can be used to improve the efficiency and scalability of non-probabilistic topic models, using RLSI and NMF as examples. Specifically, we apply GMF to RLSI [26] and NMF [13, 14], obtaining the Group RLSI (GRLSI) and Group NMF (GNMF), respectively. Like in RLSI, the objective function of GRLSI consists of squared Frobenius norm as loss function, ℓ_1-regularization on topics, and ℓ_2-regularization on document representations. Similarly to NMF, GNMF also uses squared Frobenius norm as loss function and non-negative constraints on the topics and document representations. Algorithms for optimizing the loss functions of GRLSI and GNMF are given and theoretical justification of the algorithms is shown. Time complexity analysis show that GRLSI and GNMF can achieve P times of speedup on RLSI and NMF respectively, where P is the number of classes.

Experiments on two large datasets containing about 3 million documents have verified the following points. (1) Both GRLSI and GNMF can efficiently handle the documents on a single machine, and the number is larger than those which can be processed by most existing topic modeling methods. (2) GRLSI and GNMF are more scalable and efficient than RLSI and NMF respectively, especially when the number of topics is large. (3) In GRLSI and GNMF, the shared topics as well as the class-specific topics are coherent and meaningful. (4) Experiments on another relevance dataset show that GRLSI and GNMF can help significantly improve search relevance.

Exploiting the divide and conquer strategy in the non-probabilistic methods has been investigated in computer vision [16, 25]. However, it was not clear whether it works for text data. As far as we know, this is the first work on large scale text data. Our main contributions in this paper lie in that we have empirically verified the effectiveness of the divide and conquer strategy on text data, by specifically implementing and testing the GRLSI and GNMF methods in large scale experiments.

2. RELATED WORK

The goal of topic modeling is to automatically discover the latent topics in a document collection as well as model the documents by representing them with the topics. Methods for 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 terms and documents are viewed as data generated from mixtures of topics. To generate a document, one first chooses a topic distribution. 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 [11] and LDA [5] are two widely-used probabilistic approaches to topic modeling. Please refer to [3] for a survey on probabilistic topic models. In the non-probabilistic approaches, the term vectors of documents (term-document matrix) are projected into a topic space in which each axis corresponds to a topic. A document is then represented as a vector of topics in the space. These approaches are realized as factorization of the term-document matrix such that the matrix is approximately equal to the product of a term-topic matrix and a topic-document matrix under certain constraints. LSI [7] is a representative method, which performs the factorization under the assumption that the topic vectors are orthonormal. In NMF [13, 14], the factor matrices are assumed to be nonnegative, while in RLSI [26], the factor matrices are regularized with ℓ_1 and/or ℓ_2 norms.

It has been demonstrated that topic modeling is useful for knowledge discovery, search relevance ranking, and document classification (e.g., [19, 27, 26]). Topic modeling is actually becoming one of the most important technologies in text mining, information retrieval, natural language processing, and other related fields.

The topic modeling approaches that we have discussed so far are completely unsupervised. Recently, researchers have also proposed supervised or semi-supervised approaches to topic modeling. For example, Supervised Latent Dirichlet Allocation (SLDA) [4] and Supervised Dictionary Learning (SDL) [18] are methods for incorporating supervision into probabilistic and non-probabilistic topic models. In this paper, we assume that documents have already been classified into classes, and then we conduct topic modeling on the basis of the classification to enhance scalability and efficiency.

Using document classes in topic modeling has been studied in previous literature. For probabilistic approaches, Zhai et al.(2004), for example, proposed incorporating class labels into a multinomial mixture model in order to more accurately discover topics,such that some topics are shared by all classes and other topics are specific to individual classes [28]. The discriminatively training of LDA (DiscLDA) [12] and Partially Labeled Dirichlet Allocation (PLDA) [22] incorporate class labels into LDA to achieve similar goals. For the non-probabilistic approaches, Mairal et al. (2008) [17], Bengio et al. (2009) [1], and Wang et al. (2011) [25] proposed using class labels in Sparse Coding [15, 20], a special case of RLSI, in which a dictionary for each class (i.e., topics specific to each class) is learned first, after that a common dictionary over all classes (i.e., topics shared by all classes) is learned, and finally the common and class-specific dictionaries are learned simultaneously. Group Nonnegative Matrix Factorization (GNMF) [16] extends NMF in a similar way. Both extensions on non-probabilistic methods were conducted in computer vision. As can be seen, all the previous work was not motivated toward enhancing scalability and efficiency. In this paper, we also exploit class information in topic modeling and our goal is to enhance scalability and efficiency. As far as we know, this is the first time such an investigation is conducted on text data. We also note that the formulation of GNMF in this paper is different from that in [16].

3. GROUP MATRIX FACTORIZATION

We present the formulation of Group Matrix Factorization and provide a probabilistic interpretation of it.

3.1 Problem Formulation

Suppose that we are given a document collection $\mathcal{D}$ with size N, containing terms from a vocabulary $\mathcal{V}$ with size M. A document is represented as a vector $\boldsymbol{d} \in \mathbb{R}^M$ where each entry denotes the score of the corresponding term, for example, a Boolean value indicating occurrence, term frequency, tf-idf, etc. Each document is associated with a class label $y \in \{1, \cdots, P\}$. The N documents in $\mathcal{D}$ can be classified into P classes according to their class labels and represented as $\mathcal{D} = \{\mathbf{D}_1, \cdots, \mathbf{D}_P\}$. $\mathbf{D}_p = \left[\boldsymbol{d}_1^{(p)}, \cdots, \boldsymbol{d}_{N_p}^{(p)}\right] \in \mathbb{R}^{M \times N_p}$ is the term-document matrix corresponding to class p, in which each row stands for a term and each column stands for a document. N_p is the number of documents in class p such that $\sum_{p=1}^{P} N_p = N$.

A topic is defined as a subset of terms from $\mathcal{V}$ with important weights, and is also represented as a vector $\boldsymbol{u} \in \mathbb{R}^M$ with each entry corresponding to a term. Suppose that there are K_s shared topics, denoted as a term-topic matrix $\mathbf{U}_0 = \left[\boldsymbol{u}_1^{(0)}, \cdots, \boldsymbol{u}_{K_s}^{(0)}\right] \in \mathbb{R}^{M \times K_s}$, in which each column corresponds to a shared topic. Also, for each class p, there are K_c class-specific topics, which can also be represented by a term-topic matrix $\mathbf{U}_p = \left[\boldsymbol{u}_1^{(p)}, \cdots, \boldsymbol{u}_{K_c}^{(p)}\right] \in \mathbb{R}^{M \times K_c}$, where each column stands for a class-specific topic. Then, the total number of topics in the whole collection is $K = K_s + PK_c$ [3].

The documents in each class are then modeled by the shared topics as well as the topics specific to their own class. Specifically, given the shared topics $\mathbf{U}_0$ and the class-specific topics $\mathbf{U}_p$, document $\boldsymbol{d}_n^{(p)}$ in class p is approximately represented as a linear combination of these topics, i.e.,

$$\boldsymbol{d}_n^{(p)} \approx \tilde{\mathbf{U}}_p \boldsymbol{v}_n^{(p)} = \left[\mathbf{U}_0, \mathbf{U}_p\right] \boldsymbol{v}_n^{(p)}, \tag{1}$$

where $\tilde{\mathbf{U}}_p = \left[\mathbf{U}_0, \mathbf{U}_p\right] \in \mathbb{R}^{M \times (K_s + K_c)}$ is the concatenated term-topic matrix corresponding to class p, and $\boldsymbol{v}_n^{(p)} \in \mathbb{R}^{K_s + K_c}$ is the representation of document $\boldsymbol{d}_n^{(p)}$ in latent topic space. Since a document is represented only by the shared topics and the class-specific topics corresponding to its own class, GMF actually decomposes the large-scale matrix operations concerning all the topics into multiple small-scale ones concerning only subsets of the topics, and thus reduces the computational complexity.

Let $\mathbf{V}_p = \left[\boldsymbol{v}_1^{(p)}, \cdots, \boldsymbol{v}_{N_p}^{(p)}\right] \in \mathbb{R}^{(K_s + K_c) \times N_p}$ be the topic-document matrix corresponding to class p. We denote $\mathbf{V}_p^T = \left[\mathbf{H}_p^T, \mathbf{W}_p^T\right]$ such that $\tilde{\mathbf{U}}_p \mathbf{V}_p = \mathbf{U}_0 \mathbf{H}_p + \mathbf{U}_p \mathbf{W}_p$, where $\mathbf{H}_p \in \mathbb{R}^{K_s \times N_p}$ corresponds to shared topics $\mathbf{U}_0$ and $\mathbf{W}_p \in \mathbb{R}^{K_c \times N_p}$ corresponds to class-specific topics $\mathbf{U}_p$. Table 1 gives a summary of notations.

Thus, given a document collection together with the class labels, represented as $\mathcal{D} = \{\mathbf{D}_1, \cdots, \mathbf{D}_P\}$, GMF amounts to solving the following optimization problem:

$$\begin{aligned}
\min_{\left\{\boldsymbol{u}_k^{(0)}\right\}, \left\{\boldsymbol{u}_k^{(p)}\right\}, \left\{\boldsymbol{v}_n^{(p)}\right\}} \quad & \sum_{p=1}^{P} \sum_{n=1}^{N_p} \mathcal{L}\left(\boldsymbol{d}_n^{(p)} \| \tilde{\mathbf{U}}_p \boldsymbol{v}_n^{(p)}\right) + \theta_1 \sum_{k=1}^{K_s} \mathcal{R}_1\left(\boldsymbol{u}_k^{(0)}\right) \\
& + \theta_2 \sum_{p=1}^{P} \sum_{k=1}^{K_c} \mathcal{R}_2\left(\boldsymbol{u}_k^{(p)}\right) + \theta_3 \sum_{p=1}^{P} \sum_{n=1}^{N_p} \mathcal{R}_3\left(\boldsymbol{v}_n^{(p)}\right), \\
s.t. \quad & \boldsymbol{u}_k^{(0)} \in C_1, \quad k = 1, \cdots, K_s, \\
& \boldsymbol{u}_k^{(p)} \in C_2, \quad k = 1, \cdots, K_c, p = 1, \cdots, P, \\
& \boldsymbol{v}_n^{(p)} \in C_3, \quad n = 1, \cdots, N_p, p = 1, \cdots, P,
\end{aligned} \tag{2}$$

where $\mathcal{L}(\cdot\|\cdot)$ is a loss function that measures the quality of the approximation defined in Eq. (1); $\mathcal{R}_1(\cdot)$, $\mathcal{R}_2(\cdot)$, and $\mathcal{R}_3(\cdot)$ are regularization items on shared topics, class-specific topics, and document representations, respectively; C_1, C_2, and C_3 are feasible sets for shared topics, class-specific topics, and document representations, respectively; θ_1, θ_2, and θ_3 are coefficients.

3.2 Probabilistic Interpretation

We give a probabilistic interpretation of GMF, as shown in Figure 1. In the graphical model, shared topics $\boldsymbol{u}_1^{(0)}, \cdots, \boldsymbol{u}_{K_s}^{(0)}$ and class-specific topics $\boldsymbol{u}_1^{(p)}, \cdots, \boldsymbol{u}_{K_c}^{(p)}, p = 1, \cdots, P$, are parameters. All the shared topics are independent from each other, with prior $p\left(\boldsymbol{u}_k^{(0)}\right) \propto e^{-\theta_1 \mathcal{R}_1\left(\boldsymbol{u}_k^{(0)}\right)}$ and constraint $\boldsymbol{u}_k^{(0)} \in C_1$ on each $\boldsymbol{u}_k^{(0)}$. All the class-specific topics are independent from each other, with prior $p\left(\boldsymbol{u}_k^{(p)}\right) \propto e^{-\theta_2 \mathcal{R}_2\left(\boldsymbol{u}_k^{(p)}\right)}$ and constraint $\boldsymbol{u}_k^{(p)} \in C_2$ on each $\boldsymbol{u}_k^{(p)}$. Document representations $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_N$ are regarded as latent variables, with prior $p(\boldsymbol{v}_n) \propto e^{-\theta_3 \mathcal{R}_3(\boldsymbol{v}_n)}$ and constraint $\boldsymbol{v}_n \in C_3$ on each $\boldsymbol{v}_n$. Class labels $y_1, \cdots, y_N$ are observed variables with a constant prior on each y_n. Documents $\boldsymbol{d}_1, \cdots, \boldsymbol{d}_N$ are also observed variables. Each document is generated according to a probability distribution conditioned on the shared topics, the class-specific topics, the corresponding class label, and the corresponding latent variable, i.e., $p\left(\boldsymbol{d}_n \middle| \left\{\boldsymbol{u}_k^{(0)}\right\}, \left\{\boldsymbol{u}_k^{(p)}\right\}, y_n, \boldsymbol{v}_n\right) = p\left(\boldsymbol{d}_n \middle| \left\{\boldsymbol{u}_k^{(0)}\right\}, \left\{\boldsymbol{u}_k^{(y_n)}\right\}, \boldsymbol{v}_n\right) \propto e^{-\mathcal{L}\left(\boldsymbol{d}_n \| \tilde{\mathbf{U}}_{y_n} \boldsymbol{v}_n\right)}$. Moreover, all triplets $(y_n, \boldsymbol{d}_n, \boldsymbol{v}_n)$ are independent given $\boldsymbol{u}_1^{(0)}, \cdots \boldsymbol{u}_{K_s}^{(0)}$ and $\boldsymbol{u}_1^{(p)}, \cdots, \boldsymbol{u}_{K_c}^{(p)}, p = 1, \cdots, P$. It can be easily shown that GMF formulation Eq. (2) can be obtained with Maximum A Posteriori approximation.

Table 1: Table of notations.

Notation	Meaning
M	Number of terms in vocabulary
N	Number of documents in collection
P	Number of classes
N_p	Number of documents in class p
K_s	Number of shared topics
K_c	Number of class-specific topics for each class
K	Total number of topics
$\mathbf{D}_p \in \mathbb{R}^{M \times N_p}$	Term-document matrix corresponding to class p
$\boldsymbol{d}_n^{(p)} \in \mathbb{R}^M$	The n-th document in class p
$\mathbf{U}_0 \in \mathbb{R}^{M \times K_s}$	Term-topic matrix of shared topics
$\boldsymbol{u}_k^{(0)} \in \mathbb{R}^M$	The k-th shared topic
$\mathbf{U}_p \in \mathbb{R}^{M \times K_c}$	Term-topic matrix of class-specific topics for class p
$\boldsymbol{u}_k^{(p)} \in \mathbb{R}^M$	The k-th class-specific topic in class p
$\tilde{\mathbf{U}}_p = \left[\mathbf{U}_0, \mathbf{U}_p\right]$	Concatenated term-topic matrix corresponding to class p
$\mathbf{V}_p \in \mathbb{R}^{(K_s + K_c) \times N_p}$	Topic-document matrix corresponding to class p
$\boldsymbol{v}_n^{(p)} \in \mathbb{R}^{K_s + K_c}$	Representation of $\boldsymbol{d}_n^{(p)}$ in topic space
$\mathbf{H}_p$ and $\mathbf{W}_p$	Components of $\mathbf{V}_p$: $\mathbf{V}_p^T = \left[\mathbf{H}_p^T, \mathbf{W}_p^T\right]$

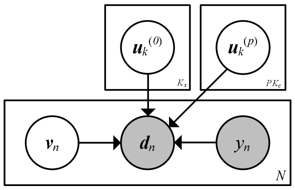

Figure 1: Graphical model of GMF.

GMF can be applied to non-probabilistic methods to further enhance their scalability and efficiency. Next, as examples, we define Group RLSI (GRLSI) and Group NMF (GNMF) under the framework of GMF.

4. GROUP RLSI

GRLSI adopts the squared Euclidean distance to measure the approximation quality and employs the same regularization schema as in RLSI [26], i.e., ℓ_1-regularization on both shared and class-specific topics and ℓ_2-regularization on document representations. The optimization problem of GRLSI is as follows:

$$\begin{aligned}
\min_{\left\{\boldsymbol{u}_k^{(0)}\right\}, \left\{\boldsymbol{u}_k^{(p)}\right\}, \left\{\boldsymbol{v}_n^{(p)}\right\}} \quad & \sum_{p=1}^{P} \sum_{n=1}^{N_p} \left\|\boldsymbol{d}_n^{(p)} - \tilde{\mathbf{U}}_p \boldsymbol{v}_n^{(p)}\right\|_2^2 + \lambda_1 \sum_{k=1}^{K_s} \left\|\boldsymbol{u}_k^{(0)}\right\|_1 \\
& + \lambda_1 \sum_{p=1}^{P} \sum_{k=1}^{K_c} \left\|\boldsymbol{u}_k^{(p)}\right\|_1 + \lambda_2 \sum_{p=1}^{P} \sum_{n=1}^{N_p} \left\|\boldsymbol{v}_n^{(p)}\right\|_2^2,
\end{aligned} \tag{3}$$

where λ_1 is the parameter controlling the ℓ_1-regularization, and λ_2

[3] A more general case is defining different numbers of class-specific topics for different classes, which can also be modeled by GMF.

Algorithm 1 Group RLSI

```
Require: $\mathbf{D}_1, \cdots, \mathbf{D}_P$
for $p = 1 : P$ do
    $\mathbf{U}_p \leftarrow$ zero matrix
    $\mathbf{V}_p \leftarrow$ random matrix
end for
repeat
    $\mathbf{U}_0 \leftarrow \text{Update}U_0\left(\{\mathbf{D}_p\}, \{\mathbf{U}_p\}, \{\mathbf{V}_p\}\right)$
    for $p = 1 : P$ do
        $\mathbf{U}_p \leftarrow \text{Update}U_p\left(\mathbf{D}_p, \mathbf{U}_0, \mathbf{V}_p\right)$
        $\mathbf{V}_p \leftarrow \text{Update}V_p\left(\mathbf{D}_p, \mathbf{U}_0, \mathbf{U}_p\right)$
    end for
until convergence
return $\mathbf{U}_0, \mathbf{U}_1, \cdots, \mathbf{U}_P, \mathbf{V}_1, \cdots, \mathbf{V}_P$
```

is the parameter controlling the ℓ_2-regularization[4]. GRLSI decomposes the large-scale matrix operations in RLSI into multiple small-scale ones and thus can be solved more efficiently.

4.1 Optimization

Optimization Eq. (3) is convex with respect to one of the variables $\mathbf{U}_0, \mathbf{U}_1, \cdots, \mathbf{U}_P, \boldsymbol{V}_1, \cdots, \mathbf{V}_P$ when the others are fixed. Thus we sequentially minimize the objective function with respect to shared topics $\mathbf{U}_0$, class-specific topics $\mathbf{U}_1, \cdots, \mathbf{U}_P$, and document representations $\boldsymbol{V}_1, \cdots, \mathbf{V}_P$. This procedure is summarized in Algorithm 1.

4.1.1 Update of Matrix $\mathbf{U}_0$

Holding $\mathbf{U}_1, \cdots, \mathbf{U}_P, \mathbf{V}_1, \cdots, \mathbf{V}_P$ fixed, the update of $\mathbf{U}_0$ amounts to the following minimization problem:

$$\min_{\mathbf{U}_0} \sum_{p=1}^{P} \left\|\mathbf{D}_p - \mathbf{U}_0\mathbf{H}_p - \mathbf{U}_p\mathbf{W}_p\right\|_F^2 + \lambda_1 \sum_{m=1}^{M}\sum_{k=1}^{K_s} \left|u_{mk}^{(0)}\right|, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm and $u_{mk}^{(0)}$ is the mk-th entry of $\mathbf{U}_0$. Eq. (4) is equivalent to

$$\min_{\mathbf{U}_0} \|\mathbf{D} - \mathbf{U}_0\mathbf{H}\|_F^2 + \lambda_1 \sum_{m=1}^{M}\sum_{k=1}^{K_s} \left|u_{mk}^{(0)}\right|, \tag{5}$$

where $\mathbf{D}$ and $\mathbf{H}$ are defined as $\mathbf{D} = [\mathbf{D}_1 - \mathbf{U}_1\mathbf{W}_1, \cdots, \mathbf{D}_P - \mathbf{U}_P\mathbf{W}_P]$ and $\mathbf{H} = [\mathbf{H}_1, \cdots, \mathbf{H}_P]$, respectively. Let $\bar{\boldsymbol{d}}_m = (d_{m1}, \cdots, d_{mN})^T$ and $\bar{\boldsymbol{u}}_m^{(0)} = \left(u_{m1}^{(0)}, \cdots, u_{mK_s}^{(0)}\right)^T$ be the column vectors whose entries are those of the m^{th} row of $\mathbf{D}$ and $\mathbf{U}_0$, respectively. Eq. (5) can be decomposed into M subproblems that can be solved independently, with each corresponding to one row of $\mathbf{U}_0$:

$$\min_{\bar{\boldsymbol{u}}_m^{(0)}} \left\|\bar{\boldsymbol{d}}_m - \mathbf{H}^T\bar{\boldsymbol{u}}_m^{(0)}\right\|_2^2 + \lambda_1 \left\|\bar{\boldsymbol{u}}_m^{(0)}\right\|_1, \tag{6}$$

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

Eq. (6) 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 method [6], coordinate descent with soft-thresholding [9, 10], Lars-Lasso algorithm [8, 21], and feature-sign search [15]. Here we choose coordinate descent with soft-thresholding. To do so, we calculate $\mathbf{S}_0 = \mathbf{H}\mathbf{H}^T = \sum_{p=1}^{P}\mathbf{H}_p\mathbf{H}_p^T \in \mathbb{R}^{K_s\times K_s}$ and $\mathbf{R}_0 = \mathbf{D}\mathbf{H}^T = \sum_{p=1}^{P}\mathbf{D}_p\mathbf{H}_p^T - \sum_{p=1}^{P}\mathbf{U}_p\mathbf{W}_p\mathbf{H}_p^T \in \mathbb{R}^{M\times K_s}$, and then update $\mathbf{U}_0$ with the following update rule:

$$u_{mk}^{(0)} \leftarrow \frac{\left(\left|r_{mk}^{(0)} - \sum_{l\neq k} s_{kl}^{(0)}u_{ml}^{(0)}\right| - \frac{1}{2}\lambda_1\right)_+ \text{sign}\left(r_{mk}^{(0)} - \sum_{l\neq k} s_{kl}^{(0)}u_{ml}^{(0)}\right)}{s_{kk}^{(0)}},$$

where $s_{ij}^{(0)}$ and $r_{ij}^{(0)}$ are the ij-th entry of $\mathbf{S}_0$ and $\mathbf{R}_0$, respectively, and $(\cdot)_+$ denotes the hinge function. The algorithm for updating $\mathbf{U}_0$ is summarized in Algorithm 2.

Algorithm 2 UpdateU_0

```
Require: $\mathbf{D}_1, \cdots, \mathbf{D}_P, \mathbf{U}_1, \cdots, \mathbf{U}_P, \mathbf{V}_1, \cdots, \mathbf{V}_P$
$\mathbf{S}_0 \leftarrow \sum_{p=1}^{P}\mathbf{H}_p\mathbf{H}_p^T$
$\mathbf{R}_0 \leftarrow \sum_{p=1}^{P}\mathbf{D}_p\mathbf{H}_p^T - \sum_{p=1}^{P}\mathbf{U}_p\mathbf{W}_p\mathbf{H}_p^T$
for $m = 1 : M$ do
    $\bar{\boldsymbol{u}}_m^{(0)} \leftarrow \mathbf{0}$
    repeat
        for $k = 1 : K_s$ do
            $x_{mk} \leftarrow r_{mk}^{(0)} - \sum_{l\neq k} s_{kl}^{(0)}u_{ml}^{(0)}$
            $u_{mk}^{(0)} \leftarrow \frac{\left(|x_{mk}| - \frac{1}{2}\lambda_1\right)_+ \text{sign}(x_{mk})}{s_{kk}^{(0)}}$
        end for
    until convergence
end for
return $\mathbf{U}_0$
```

Algorithm 3 UpdateU_p

```
Require: $\mathbf{D}_p, \mathbf{U}_0, \mathbf{V}_p$
$\mathbf{S}_p \leftarrow \mathbf{W}_p\mathbf{W}_p^T$
$\mathbf{R}_p \leftarrow \mathbf{D}_p\mathbf{W}_p^T - \mathbf{U}_0\mathbf{H}_p\mathbf{W}_p^T$
for $m = 1 : M$ do
    $\bar{\boldsymbol{u}}_m^{(p)} \leftarrow \mathbf{0}$
    repeat
        for $k = 1 : K_c$ do
            $x_{mk} \leftarrow r_{mk}^{(p)} - \sum_{l\neq k} s_{kl}^{(p)}u_{ml}^{(p)}$
            $u_{mk}^{(p)} \leftarrow \frac{\left(|x_{mk}| - \frac{1}{2}\lambda_1\right)_+ \text{sign}(x_{mk})}{s_{kk}^{(p)}}$
        end for
    until convergence
end for
return $\mathbf{U}_p$
```

4.1.2 Update of Matrix $\mathbf{U}_p$

Holding the other variables fixed, the update of $\mathbf{U}_p$ amounts to the following optimization problem:

$$\min_{\mathbf{U}_p} \left\|\mathbf{D}_p - \mathbf{U}_0\mathbf{H}_p - \mathbf{U}_p\mathbf{W}_p\right\|_F^2 + \lambda_1 \sum_{m=1}^{M}\sum_{k=1}^{K_c} \left|u_{mk}^{(p)}\right|, \tag{7}$$

where $u_{mk}^{(p)}$ is the mk-th entry of $\mathbf{U}_p$. Eq. (7) can be optimized with the same technique presented for optimizing Eq. (5). We calculate $\mathbf{S}_p = \mathbf{W}_p\mathbf{W}_p^T \in \mathbb{R}^{K_c\times K_c}$ and $\mathbf{R}_p = \mathbf{D}_p\mathbf{W}_p^T - \mathbf{U}_0\mathbf{H}_p\mathbf{W}_p^T \in \mathbb{R}^{M\times K_c}$, and then update $\mathbf{U}_p$ with the following update rule:

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

where $s_{ij}^{(p)}$ and $r_{ij}^{(p)}$ are the ij-th entry of $\mathbf{S}_p$ and $\mathbf{R}_p$, respectively. The algorithm for updating $\mathbf{U}_p$ is summarized in Algorithm 3.

[4] A more general case is setting different regularization parameters for shared topics and class-specific topics, for separately controlling the sparsity of shared topics and class-specific topics.

Algorithm 4 UpdateV_p

```
Require: $\mathbf{D}_p, \mathbf{U}_0, \mathbf{U}_p$
1: $\mathbf{\Sigma}_p \leftarrow \left(\tilde{\mathbf{U}}_p^T \tilde{\mathbf{U}}_p + \lambda_2 \mathbf{I}\right)^{-1}$
2: $\mathbf{\Phi}_p \leftarrow \tilde{\mathbf{U}}_p^T \mathbf{D}_p$
3: for $n = 1 : N_p$ do
4:     $\boldsymbol{v}_n^{(p)} \leftarrow \mathbf{\Sigma}_p \boldsymbol{\phi}_n^{(p)}$, where $\boldsymbol{\phi}_n^{(p)}$ is the $n$-th column of $\mathbf{\Phi}_p$
5: end for
6: return $\mathbf{V}_p$
```

Table 2: Time complexity (per iteration) of RLSI and GRLSI.

	RLSI	GRLSI
Update **U**	$\frac{K^2N+\text{AvgDL}\times KN+IK^2M}{Q}$	$\frac{\frac{K^2N}{P}+\text{AvgDL}\times KN+IK^2M}{PQ}$
Update **V**	$\frac{\gamma^2K^2M+K^3+\text{AvgDL}\times\gamma KN+K^2N}{Q}$	$\frac{\gamma^2K^2M+\frac{K^3}{P}+\text{AvgDL}\times\gamma KN+\frac{K^2N}{P}}{PQ}$

4.1.3 Update of Matrix $\mathbf{V}_p$

The update of $\mathbf{V}_p$ with the other variables fixed is a least squares problem with ℓ_2-regularization. It can also be decomposed into N_p optimization problems, with each corresponding to one $\boldsymbol{v}_n^{(p)}$ and can be solved in parallel:

$$\min_{\boldsymbol{v}_n^{(p)}} \left\|\boldsymbol{d}_n^{(p)} - \tilde{\mathbf{U}}_p \boldsymbol{v}_n^{(p)}\right\|_2^2 + \lambda_2 \left\|\boldsymbol{v}_n^{(p)}\right\|_2^2,$$

for $n = 1, \cdots, N_p$. It is a standard ℓ_2-regularized least squares problem and the solution is:

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

Algorithm 4 shows the procedure.

4.2 Time Complexity

The formulation of learning in GRLSI is decomposable and thus can be processed in parallel. Specifically, the **for**-loops in Algorithm 2 (i.e., line 3 to line 11), Algorithm 3 (i.e., line 3 to line 11), and Algorithm 4 (i.e., line 3 to line 5) can be processed in parallel. In this paper, we implement GRLSI as well as RLSI using multi-threaded programming and compare their time complexities. Table 2 shows the results, where Q is the number of threads, γ the topic sparsity, and AvgDL the average document length. For GRLSI, the "Update **U**" includes the update of $\mathbf{U}_0, \mathbf{U}_1, \cdots, \mathbf{U}_P$ and the "Update **V**" includes the update of $\mathbf{V}_1, \cdots, \mathbf{V}_P$. From the results, we can see that GRLSI is approximately P times faster than RLSI in terms of time complexity. Here we suppose that 1) the documents are evenly distributed to the P classes; 2) the number of class-specific topics in each class is similar to the number of shared topics; and 3) the topic sparsity of GRLSI is similar to the topic sparsity of RLSI.

4.3 Folding-in New Documents

Folding-in refers to the problem of computing representations of documents that were not contained in the original training collection. When a new document d, represented as $\boldsymbol{d} \in \mathbb{R}^M$ in the term space, is given, its representation in the topic space can be computed under two different conditions. First, if the class label of the document is given, denoted as y_d, we represent the document in the topic space as

$$\boldsymbol{v}_d = \arg\min_{\boldsymbol{v}} \|\boldsymbol{d} - \tilde{\mathbf{U}}_{y_d} \boldsymbol{v}\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2. \tag{8}$$

Algorithm 5 Group NMF

```
Require: $\mathbf{D}_1, \cdots, \mathbf{D}_P$
1: $\mathbf{U}_0 \leftarrow$ random matrix
2: for $p = 1 : P$ do
3:     $\mathbf{U}_p \leftarrow$ random matrix
4:     $\mathbf{V}_p \leftarrow$ random matrix
5: end for
6: repeat
7:     $\mathbf{U}_0 \leftarrow \mathbf{U}_0 * \frac{\sum_{p=1}^P \mathbf{D}_p \mathbf{H}_p^T}{\sum_{p=1}^P \mathbf{U}_0 \mathbf{H}_p \mathbf{H}_p^T + \sum_{p=1}^P \mathbf{U}_p \mathbf{W}_p \mathbf{H}_p^T}$
8:     for $p = 1 : P$ do
9:         $\mathbf{U}_p \leftarrow \mathbf{U}_p * \frac{\mathbf{D}_p \mathbf{W}_p^T}{\mathbf{U}_p \mathbf{W}_p \mathbf{W}_p^T + \mathbf{U}_0 \mathbf{H}_p \mathbf{W}_p^T}$
10:        $\mathbf{V}_p \leftarrow \mathbf{V}_p * \frac{\tilde{\mathbf{U}}_p^T \mathbf{D}_p}{\tilde{\mathbf{U}}_p^T \tilde{\mathbf{U}}_p \mathbf{V}_p}$
11:    end for
12: until convergence
13: return $\mathbf{U}_0, \mathbf{U}_1, \cdots, \mathbf{U}_P, \mathbf{V}_1, \cdots, \mathbf{V}_P$
```

Second, if the document label is unknown, we first define the error of classifying document d into class p as

$$\mathcal{E}\left(\boldsymbol{d}; \tilde{\mathbf{U}}_p\right) = \min_{\boldsymbol{v}} \left\|\boldsymbol{d} - \tilde{\mathbf{U}}_p \boldsymbol{v}\right\|_2^2 + \lambda_2 \|\boldsymbol{v}\|_2^2,$$

and predict the class label of document d by

$$y_d = \arg\min_p \mathcal{E}\left(\boldsymbol{d}; \tilde{\mathbf{U}}_p\right).$$

We then represent the document in the topic space with Eq. (8).

5. GROUP NMF

Similarly we can define Group NMF (GNMF) by adopting the squared Euclidean distance to measure the approximation quality and employing the nonnegative constraints on shared topics, class-specific topics, and document representations, as in NMF [13, 14]. The optimization problem of GNMF is as follows:

$$\begin{aligned} \min_{\left\{\boldsymbol{u}_k^{(0)}\right\}, \left\{\boldsymbol{u}_k^{(p)}\right\}, \left\{\boldsymbol{v}_n^{(p)}\right\}} \; & \sum_{p=1}^{P} \sum_{n=1}^{N_p} \left\|\boldsymbol{d}_n^{(p)} - \tilde{\mathbf{U}}_p \boldsymbol{v}_n^{(p)}\right\|_2^2 \\ s.t. \quad & \boldsymbol{u}_k^{(0)} \geq 0, \quad k = 1, \cdots, K_s, \\ & \boldsymbol{u}_k^{(p)} \geq 0, \quad k = 1, \cdots, K_c, p = 1, \cdots, P, \\ & \boldsymbol{v}_n^{(p)} \geq 0, \quad n = 1, \cdots, N_p, p = 1, \cdots, P, \end{aligned} \tag{9}$$

which decomposes the large-scale matrix operations in NMF into multiple small-scale ones and thus can be solved more efficiently.

5.1 Optimization

Optimization Eq. (9) is convex with respect to one of the variables $\mathbf{U}_0$, $\mathbf{U}_1, \cdots, \mathbf{U}_P$, $V_1, \cdots, \mathbf{V}_P$ while keeping the others fixed. We again sequentially minimize the objective function with respect to shared topics $\mathbf{U}_0$, class-specific topics $\mathbf{U}_1, \cdots, \mathbf{U}_P$, and document representations $V_1, \cdots, \mathbf{V}_P$. The procedure is summarized in Algorithm 5, where the operator "$*$" represents the entry-wise multiplication, and the division is also entry-wise.

The multiplicative update rules in Algorithm 5 were first proposed in [16] and then applied in [25]. However, neither [16] nor [25] gave sufficient evidence to demonstrate the correctness of them. Here, we theoretically justify Algorithm 5, showing that the objective in Eq. (9) is nonincreasing under the update rules in Algorithm 5. We first proof Proposition 1.

Proposition 1. *Given* $\mathbf{X}, \mathbf{Y} \in \mathbb{R}_+^{M\times N}$ *and* $\mathbf{S} \in \mathbb{R}_+^{K\times N}$*, consider optimization problem* $\min_{\mathbf{A}\geq 0} \|\mathbf{X} - \mathbf{Y} - \mathbf{A}\mathbf{S}\|_F^2$*. The objective is nonincreasing under the update rule*

$$\mathbf{A} \leftarrow \mathbf{A} * \frac{\mathbf{X}\mathbf{S}^T}{\mathbf{A}\mathbf{S}\mathbf{S}^T + \mathbf{Y}\mathbf{S}^T},$$

where the operator "$*$" represents the entry-wise multiplication, and the division is also entry-wise.

A proof sketch of the proposition can be found in Appendix.

5.1.1 Update of Matrix $\mathbf{U}_0$

Holding $\mathbf{U}_1, \cdots, \mathbf{U}_P, \mathbf{V}_1, \cdots, \mathbf{V}_P$ fixed, the update of $\mathbf{U}_0$ amounts to the following minimization problem:

$$\min_{\mathbf{U}_0 \geq 0} \sum_{p=1}^{P} \left\|\mathbf{D}_p - \mathbf{U}_0\mathbf{H}_p - \mathbf{U}_p\mathbf{W}_p\right\|_F^2,$$

which can be rewritten as

$$\min_{\mathbf{U}_0 \geq 0} \|\mathbf{E} - \mathbf{F} - \mathbf{U}_0\mathbf{H}\|_F^2,$$

where $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{H}$ are respectively defined as $\mathbf{E} = [\mathbf{D}_1, \cdots, \mathbf{D}_P]$, $\mathbf{F} = [\mathbf{U}_1\mathbf{W}_1, \cdots, \mathbf{U}_P\mathbf{W}_P]$, and $\mathbf{H} = [\mathbf{H}_1, \cdots, \mathbf{H}_P]$. It is easy to show that the objective is nonincreasing under the update rule

$$\mathbf{U}_0 \leftarrow \mathbf{U}_0 * \frac{\sum_{p=1}^{P} \mathbf{D}_p\mathbf{H}_p^T}{\sum_{p=1}^{P} \mathbf{U}_0\mathbf{H}_p\mathbf{H}_p^T + \sum_{p=1}^{P} \mathbf{U}_p\mathbf{W}_p\mathbf{H}_p^T},$$

according to Proposition 1.

5.1.2 Update of Matrix $\mathbf{U}_p$

Holding the other variables fixed, the update of $\mathbf{U}_p$ amounts to the following optimization problem:

$$\min_{\mathbf{U}_p \geq 0} \left\|\mathbf{D}_p - \mathbf{U}_0\mathbf{H}_p - \mathbf{U}_p\mathbf{W}_p\right\|_F^2.$$

According to Proposition 1 we get the multiplicative update rule:

$$\mathbf{U}_p \leftarrow \mathbf{U}_p * \frac{\mathbf{D}_p\mathbf{W}_p^T}{\mathbf{U}_p\mathbf{W}_p\mathbf{W}_p^T + \mathbf{U}_0\mathbf{H}_p\mathbf{W}_p^T},$$

which keeps the objective nonincreasing.

5.1.3 Update of Matrix $\mathbf{V}_p$

The update of $\mathbf{V}_p$ with the other variables fixed amounts to the following optimization problem:

$$\min_{\mathbf{V}_p \geq 0} \left\|\mathbf{D}_p - \tilde{\mathbf{U}}_p\mathbf{V}_p\right\|_F^2.$$

As demonstrated in [14], $\mathbf{V}_p$ can be updated with the following update rule:

$$\mathbf{V}_p \leftarrow \mathbf{V}_p * \frac{\tilde{\mathbf{U}}_p^T\mathbf{D}_p}{\tilde{\mathbf{U}}_p^T\tilde{\mathbf{U}}_p\mathbf{V}_p},$$

which keeps the objective nonincreasing.

5.2 Time Complexity

The multiplicative update rules of GNMF (i.e., line 7, line 9, and line 10 in Algorithm 5) can be processed in parallel since the multiplication and division are both entry-wise. In this paper, we implement GNMF as well as NMF using multithreaded programming and compare their time complexities. Table 3 shows the results, where Q is the number of threads and AvgDL is the average document length. For GNMF, the "Update $\mathbf{U}$" includes the update of $\mathbf{U}_0, \mathbf{U}_1, \cdots, \mathbf{U}_P$ and the "Update $\mathbf{V}$" includes the update of $\mathbf{V}_1, \cdots, \mathbf{V}_P$. From the results, we can see that GNMF are approximately P times faster than NMF in terms of time complexity. Here we also make the same assumptions as in Section 4.2.

Table 3: Time complexity (per iteration) of NMF and GNMF.

	NMF	GNMF
Update $\mathbf{U}$	$\frac{\text{AvgDL}\times KN+K^2M+K^2N}{Q}$	$\frac{\text{AvgDL}\times KN+K^2M+\frac{K^2N}{P}}{PQ}$
Update $\mathbf{V}$	$\frac{\text{AvgDL}\times KN+K^2M+K^2N}{Q}$	$\frac{\text{AvgDL}\times KN+K^2M+\frac{K^2N}{P}}{PQ}$

5.3 Folding-in New Documents

Given a new document $\boldsymbol{d} \in \mathbb{R}^M$, its representation in the topic space can be computed under two different conditions. First, if the class label y_d is also given, we can represent the document in the topic space as

$$\boldsymbol{v}_d = \arg\min_{\boldsymbol{v}\geq 0} \|\boldsymbol{d} - \tilde{\mathbf{U}}_{y_d}\boldsymbol{v}\|_2^2. \tag{10}$$

Second, if the document label is unknown, we first define the error of classifying document d into class p as

$$\mathcal{E}\left(\boldsymbol{d}; \tilde{\mathbf{U}}_p\right) = \min_{\boldsymbol{v}\geq 0} \left\|\boldsymbol{d} - \tilde{\mathbf{U}}_p\boldsymbol{v}\right\|_2^2,$$

and predict the class label of document d by

$$y_d = \arg\min_p \mathcal{E}\left(\boldsymbol{d}; \tilde{\mathbf{U}}_p\right).$$

We then represent the document in the topic space with Eq. (10).

6. RELEVANCE RANKING

Topic modeling can be used in a wide variety of applications. We apply GRLSI and GNMF to relevance ranking in search and evaluate their performances in comparison to RLSI and NMF respectively. The use of topic modeling techniques such as LSI was proposed in IR many years ago [7]. Two recent works [27, 26] demonstrated that improvements on relevance ranking can be achieved by using topic modeling.

The motivation of incorporating topic modeling into relevance ranking is to reduce "term mismatch". Traditional relevance models, such as VSM [24] and BM25 [23], are all based on term matching. The term mismatch problem arises when the author of a document and the user of a search system use different terms to describe the same concept, and in such a case the search may not be carried out successfully. For example, if the query contains the term "airplane" but the document contains the term "aircraft", then there is a mismatch and the document may not be viewed as relevant. In the topic space, however, it is very likely that the two terms are in the same topic, and thus the use of matching score in the topic space may help improve the relevance ranking. 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 general way of using topic models in IR is as follows. Suppose that there is a pre-learned topic model. Given a query q and a document d, we first represent them in the topic space as $\boldsymbol{v}_q$ and $\boldsymbol{v}_d$ respectively. Then we calculate the matching score between the query and the document in the topic space as the cosine similarity between $\boldsymbol{v}_q$ and $\boldsymbol{v}_d$. The topic matching score $s_{topic}(q, d)$ is then linearly combined with the term matching score $s_{term}(q, d)$ for final relevance ranking. The final relevance ranking score $s(q, d)$ is calculated as:

$$s(q, d) = \alpha s_{topic}(q, d) + (1 - \alpha)s_{term}(q, d), \tag{11}$$

Table 4: Sizes of Wikipedia and Web-I.

Dataset	# terms	# documents	# classes
Wikipedia	610,035	2,807,535	25
Web-I	530,905	3,184,138	204

Table 5: Statistics of Wikipedia and Web-I.

Dataset	Min	Max	R	Mean	STD	CV
Wikipedia	185	991,695	991,510	112301.4	200123.6	1.8
Web-I	226	29,999	29,773	15608.5	11152.0	0.7

where $\alpha \in [0, 1]$ is the coefficient. $s_{term}(q, d)$ can be calculated with any existing term-based model, for example, VSM and BM25.

7. EXPERIMENTS

We have conducted experiments to test the efficiency and effectiveness of GRLSI and GNMF.

7.1 Experimental Settings

We tested the efficiency and effectiveness of GRLSI and GNMF on two datasets[5]: Wikipedia dataset which consists of articles downloaded from the English version of Wikipedia and Web-I dataset which consists of webpages randomly sampled from a crawl of the Internet at a commercial search engine. The Wikipedia dataset contains 2,807,535 articles and the Web-I dataset contains 3,184,138 web documents. For both datasets, the titles and bodies were taken as the contents of the documents. Stop words in a standard list and terms whose total frequencies are less than 10 were removed. Table 4 lists the sizes of Wikipedia and Web-I datasets.

In the Wikipedia dataset, documents are associated with labels representing the categories of them. We adopted the 25 first-level categories in the Wikipedia hierarchy, i.e., each Wikipedia document is categorized into one of the 25 categories. The categories include “agriculture”, “arts”, “business”, “education”, “law”, etc. In the Web-I dataset, similarly, all documents are categorized into one of the ODP categories by a built-in classifier at the search engine. There are 204 categories from the second-level ODP categories, including “arts/music”, “business/management”, “computer/graphics”, “science/chemistry”, “sports/baseball”, etc. Table 5 gives the statistics of both Wikipedia and Web-I, where Min and Max stand for the minimal and maximal class sizes respectively, R is the range of class sizes, i.e., R = Max − Min, Mean and STD represent the mean value and the standard deviation of class sizes respectively, and CV is the coefficient of variance, i.e., CV = STD/Mean. One can see that Web-I has smaller R and CV values, indicating that it has a smaller degree of dispersion in the distribution of class sizes. From the table, we can see that although these two datasets have similar data sizes, the granularities of classes, i.e., number of classes and average number of documents per class, are very different.

We tested RLSI, NMF, GRLSI and GNMF on the Wikipedia dataset and Web-I dataset under different parameter settings. We used single machine implementations of the methods. Specifically, for the Wikipedia dataset, we set the number of class-specific topics per class and the number of shared topics in GRLSI and GNMF as $(K_s, K_c) = (10, 4)/(20, 8)/(50, 20)/(100, 40)$, resulting in $K = 110/220/550/1100$ total number of topics. (Note that the total number of topics in GRLSI and GNMF is $K_s + 25 \times K_c$, where 25 is the number of classes in the Wikipedia dataset.) We set the number of topics in RLSI and NMF as $110/220/550/1100$ for fair comparison. For the Web-I dataset, we decided the number of class-specific topics per class and the number of shared topics in GRLSI and GNMF as $(K_s, K_c) = (10, 5)/(20, 10)/(40, 20)/(100, 50)$, resulting in $K = 1030/2060/4120/10300$ as the total number of topics. (The total number of topics in GRLSI and GNMF is $K_s + 204 \times K_c$, where 204 is the number of classes in the Web-I dataset.) As will be explained later, we found that it is not possible to run RLSI and NMF with such large numbers of topics on a single machine. Thus, we determined the number of topics in RLSI and NMF as $100/200/500/1000$. Parameter λ_1 in GRLSI and RLSI, which controls the sparsity of topics, was selected from $0.01/0.02/0.05/0.1$, for both datasets. Parameter λ_2 in GRLSI and RLSI was fixed to 0.1, following the experimental results in [26].

Table 6: Execution time (per iteration) of RLSI on Wikipedia.

Min.	$K = 110$	$K = 220$	$K = 550$	$K = 1100$
$\lambda_1 = 0.01$	19.49	44.13	110.35	342.59
$\lambda_1 = 0.02$	19.02	43.64	93.47	332.33
$\lambda_1 = 0.05$	16.73	34.63	90.45	318.27
$\lambda_1 = 0.1$	14.91	27.26	89.92	307.67

Table 7: Execution time (per iteration) of GRLSI on Wikipedia.

Min.	$K = 110$	$K = 220$	$K = 550$	$K = 1100$
$\lambda_1 = 0.01$	14.99	23.27	51.95	106.13
$\lambda_1 = 0.02$	14.01	22.88	50.17	104.13
$\lambda_1 = 0.05$	13.95	22.68	48.25	99.03
$\lambda_1 = 0.1$	14.05	22.47	48.07	97.13

We also conducted search relevance experiments to test the effectiveness of GRLSI and GNMF on another dataset, the Web-II dataset, which is obtained from the same web search engine. The dataset consists of 752,365 documents, 30,000 queries, and relevance judgments on the documents with respect to the queries. The relevance judgments are at five levels: “perfect”, “excellent”, “good”, “fair”, and “bad”. There are in total 837,717 judged query-document pairs. The documents in Web-II are classified into 204 ODP categories with the same classifier as in Web-I. We randomly split the queries into validation/test sets, each has 15,000/15,000 queries. We used the validation set for parameter tuning and the test set for evaluation. We adopted MAP and NDCG at the positions of 1, 3, 5, and 10 as evaluation measures for relevance ranking. When calculating MAP, we considered “perfect”, “excellent”, and “good” as “relevant”, and the other two as “irrelevant”.

All of the experiments were conducted on a server with AMD Opteron 2.10GHz multi-core processor (2×12 cores), 96GB RAM. All the methods were implemented using C# multithreaded programming, with the thread number being 24.

7.2 Experiment 1

In this experiment, we evaluated the efficiency improvement of GRLSI and GNMF over RLSI and NMF on the Wikipedia dataset and the Web-I dataset. We ran all the methods in 100 iterations. For each method, the average execution time per iteration was recorded.

Table 6 and Table 7 report the average execution time per iteration for RLSI and GRLSI on Wikipedia, under different settings of topic numbers and λ_1 values. Figure 2 further shows average time per iteration of GRLSI and RLSI versus numbers of topics when $\lambda_1 = 0.01$. Figure 3 shows the average time per iteration of GNMF over NMF on Wikipedia, versus numbers of topics. From these results, we can conclude that GRLSI and GNMF consistently outperform RLSI and NMF, respectively, in terms of efficiency. More speedup can be achieved when total number of topics increases.

[5] We plan to release the two datasets to the research communities.

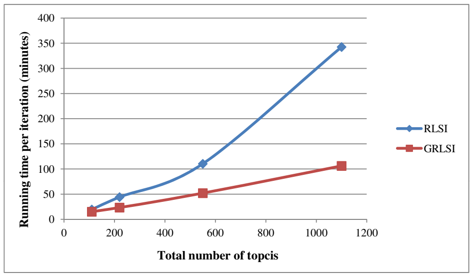

Figure 2: Execution time of RLSI and GRLSI on Wikipedia.

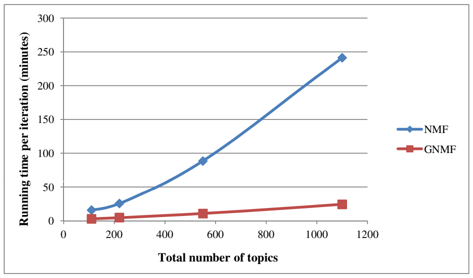

Figure 3: Execution time of NMF and GNMF on Wikipedia.

The results indicate that GRLSI and GNMF are superior to RLSI and NMF in terms of efficiency.

Table 8 and Table 9 report the average execution time per iteration for RLSI and GRLSI on Web-I with respect to different settings of topic numbers and λ_1 values. Figure 4 shows the average execution time per iteration of GRLSI and RLSI when λ_1 equals 0.01. Figure 5 shows the results of GNMF and NMF. In fact we were not able to run RLSI and NMF on the single machine, when the number of topics is larger than 1,000. The results indicate that GRLSI and GNMF have better efficiency and scalability, particularly when the number of topics gets large.

From the experimental results reported above, we can conclude that applying GMF to non-probabilistic methods of RLSI and NMF can significantly improve the efficiency and scalability of them. The proposed GRLSI and GNMF methods can handle much larger numbers of topics and much larger datasets.

Next, we evaluated the effectiveness of GRLSI and GNMF by checking the readability of the topics generated by them. As example, we show the topics generated by GRLSI and GNMF in the setting of $(K_s = 20, K_c = 8(10), \lambda_1 = 0.01, \lambda_2 = 0.1)$ for both Wikipedia and Web-I. Table 10 and Table 11 present example topics randomly selected from the topics discovered by GRLSI and GNMF on Wikipedia and Web-I. For each of the datasets and each of the methods, 3 shared topics and 9 class-specific topics are presented. The corresponding class labels are also shown for the class-specific topics. Top 6 weighted terms are shown for each topic. From all the results (including the results in other parameter settings), we found that (1) GRLSI and GNMF can discover readable topics. Both of the shared topics and the class-specific topics are coherent and easy to understand. (2) For each class, GRLSI and GNMF can discover class-specific topics that characterize the class. (3) GRLSI discovers compact topics (the average topic compactness AvgComp = 0.0032 for Wikipedia topics and AvgComp = 0.0018 for Web-I topics) [6] as expected.

Table 8: Execution time (per iteration) of RLSI on Web-I.

Min.	$K = 100$	$K = 200$	$K = 500$	$K = 1000$
$\lambda_1 = 0.01$	26.57	49.79	123.45	324.58
$\lambda_1 = 0.02$	26.79	39.24	117.54	313.49
$\lambda_1 = 0.05$	23.23	34.64	110.67	303.24
$\lambda_1 = 0.1$	14.19	32.68	100.25	301.74

Table 9: Execution time (per iteration) of GRLSI on Web-I.

Min.	$K = 1030$	$K = 2060$	$K = 4120$	$K = 10300$
$\lambda_1 = 0.01$	35.48	57.50	104.37	438.29
$\lambda_1 = 0.02$	35.30	55.60	99.46	427.22
$\lambda_1 = 0.05$	35.36	52.63	94.78	414.37
$\lambda_1 = 0.1$	34.86	50.44	92.15	409.25

We further evaluated the shared topics discovered from Wikipedia (Table 10) and Web-I (Table 11). In the Web-I dataset, the shared topics seem to characterize general information. In the Wikipedia dataset some of the shared topics are similar to the class-specific topics in category "geography". We checked the Wikipedia dataset and found that this is because more than one third of Wikipedia articles fall into category "geography", and some geography related topics appear to be general in the document collection.

From the experimental results reported above, we can conclude that applying GMF to non-probabilistic methods of RLSI and NMF can maintain the same level of readability while significantly improving the efficiency and scalability. The resulting methods of GRLSI and GNMF can really find coherent and meaningful topics. This is true for not only class-specific topics, but also shared topics.

7.3 Experiment 2

In this experiment, we tested the effectiveness of GRLSI and GNMF by using the topics generated by them with the Web-I dataset in search relevance ranking on the Web-II dataset[7]. Specifically, for GRLSI, we combined the topic matching scores with the term matching scores given by BM25, denoted as "BM25+GRLSI". We took RLSI and CRLSI as baselines, denoted as "BM25+RLSI" and "BM25+CRLSI", respectively. In the former an RLSI model is trained for the whole Web-I dataset and in the latter an RLSI model is trained for each class. Similarly, for GNMF, we combined the topic matching scores with the term matching scores by BM25, denoted as "BM25+GNMF". We took NMF and CNMF as baselines, denoted as "BM25+NMF" and "BM25+CNMF", respectively. In the former an NMF model is trained for the whole Web-I dataset and in the latter an NMF model is trained for each class.

GRLSI, RLSI, GNMF, and NMF were trained on Web-I dataset with the same parameter settings in Section 7.1. For CRLSI and CNMF, we also trained the models on Web-I dataset under the same parameter settings in Section 7.1, except parameter K_s, as there exists no shared topic in CRLSI and CNMF.

To evaluate the relevance performance of these topic models on Web-II, we took a heuristic method for relevance ranking. Given a query q and a document d (and its label y_d), the method assigns the query into the same class that the document belongs to, i.e., class y_d, and then calculates the matching score between the query and the document in the topic space using the techniques described above for GRLSI and CRLSI (also GNMF and CNMF). The method then ranks the documents based on their relevance scores. The relevance score of a document is calculated as a linear combination of the BM25 score and the topic matching score

[6] Average topic compactness is defined as average ratio of terms with non-zero weights per topic.

[7] We did not try to use the topics generated with Wikipedia, because the categories are not consistent with the categories in Web-II.

Table 10: Topics discovered by GRLSI (top) and GNMF (bottom) on Wikipedia.

	Shared topics			Arts			Geography			Politics		
GRLSI	commune	state	political	album	rock	groups	province	municipality	communes	elections	states	kingdom
	communes	highways	party	albums	american	musical	state	municipalities	commune	election	congressional	political
	department	route	colour	singers	musicians	music	village	gmina	department	weapon	delegations	parties
	places	highway	india	musicians	singers	rappers	villages	voivodeship	france	party	elections	country
	france	india	canada	track	country	metal	highways	population	departments	parties	united	party
	populated	brazil	australia	listing	english	heavy	united	germany	places	political	senate	fascism
GNMF	places	new	language	album	groups	rappers	village	district	department	elections	war	military
	populated	york	japanese	albums	rock	musicians	villages	germany	commune	election	world	country
	village	city	films	track	american	american	england	districts	communes	results	poland	units
	azerbaijan	zealand	cast	listing	musical	singers	india	town	france	members	weapons	formations
	population	jersey	chinese	released	metal	singles	population	administrative	departments	parties	conflict	army
	municipality	routes	english	band	musicians	wiley	central	towns	home	held	union	infantry

Table 11: Topics discovered by GRLSI (top) and GNMF (bottom) on Web-I.

	Shared topics			Arts/literature			Business/healthcare			Computers/internet		
GRLSI	video	business	games	poems	harry	book	dental	healthcare	care	chat	facebook	web
	phone	services	game	poetry	potter	chapter	dentist	practice	medical	teen	people	hosting
	mobile	company	cheats	poem	books	summary	care	test	health	online	connect	design
	tv	service	xbox	poets	rowling	books	dentistry	management	equipment	people	sign	website
	cell	products	ign	love	series	analysis	dentists	exam	ppo	friends	web	domain
	phones	management	pc	poet	children	author	health	patient	supplies	join	password	internet
GNMF	www	products	day	poems	harry	books	dentist	healthcare	medical	google	facebook	design
	http	product	october	quotes	potter	children	dentists	management	equipment	maps	people	web
	org	quality	september	shakespear	rowling	read	dentistry	patient	supplies	blog	connect	website
	website	buy	july	william	series	reading	dr	hospital	surgical	gmail	sign	development
	net	accessories	june	poetry	deathly	list	dental	solutions	patient	map	friends	marketing
	html	store	august	poets	hallows	readers	cosmetic	nursing	hospital	engine	password	graphic

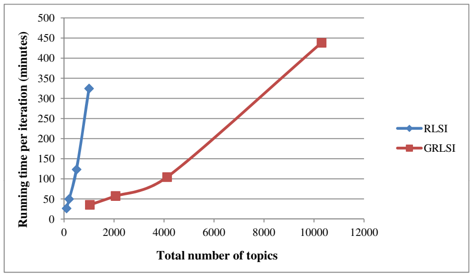

Figure 4: Execution time of RLSI and GRLSI on Web-I.

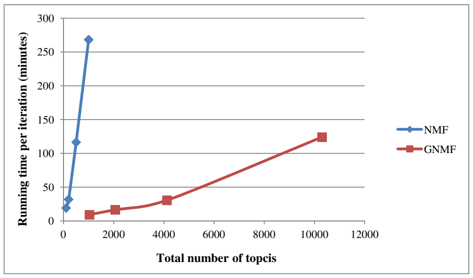

Figure 5: Execution time of NMF and GNMF on Web-I.

between the document and the query. For RLSI (also NMF), neither document labels nor query labels were needed. We directly calculated the matching score between a query and a document in the topic space using the techniques described in [26]. The trading-off parameter α in the linear combination was set from 0 to 1 in steps of 0.1 for all methods. The heuristic method of automatic assignment of a query into a class has the advantage of better efficiency in online prediction, given that usually the number of classes is large. Even though this is heuristic, our experimental results show that it is effective.

Table 12 and Table 13 show the retrieval performance of RLSI families and NMF families on the test set of Web-II respectively, obtained with the best parameter setting determined by the validation set. From the results, we can see that (1) all of these methods can *significantly* improve the baseline BM25 (t-test, p-value < 0.05). (2) GRLSI and GNMF perform *significantly* better than CRLSI and CNMF respectively (t-test, p-value < 0.05), indicating the effectiveness of Group Matrix Factorization, specifically, the use of shared topics. (3) GRLSI and GNMF perform slightly worse than RLSI and NMF, but they can achieve much higher efficiency and scalability, as described in Section 7.2. The decreases of accuracy by GRLSI and GNMF are very small, e.g., NDCG@1 drops only 0.0010 for GRLSI and 0.0011 for GNMF. (4) The NMF families perform better than the RLSI families. This is because we did not further tune the parameters for the RLSI families. The results in [26] show that with fine tuning RLSI can achieve high performances, and we anticipate that this is also the case for the other RLSI methods. We conclude that both GRLSI and GNMF are useful for relevance ranking with high accuracies.

8. CONCLUSIONS

In this paper, we have investigated the possibilities of further enhancing the scalability and efficiency of non-probabilistic topic modeling methods. We have proposed a general topic modeling technique, referred to as Group Matrix Factorization (GMF), which conducts topic modeling on the basis of existing classes of documents. Thus the learning of a large number of topics (i.e.,class-specific topics) can be performed in parallel. Although the strategy has been tried in computer vision, this is the first compre-

Table 12: Relevance performance of RLSI families on Web-II.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3006	0.3043	0.3490	0.3910	0.4805
BM25+RLSI	0.3050	0.3076	0.3539	0.3943	0.4858
BM25+CRLSI	0.3027	0.3051	0.3509	0.3927	0.4840
BM25+GRLSI	0.3039	0.3066	0.3520	0.3934	0.4855

Table 13: Relevance performance of NMF families on Web-II.

Method	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3006	0.3043	0.3490	0.3910	0.4805
BM25+NMF	0.3057	0.3091	0.3546	0.3960	0.4895
BM25+CNMF	0.3033	0.3055	0.3512	0.3934	0.4869
BM25+GNMF	0.3046	0.3080	0.3530	0.3955	0.4887

hensive study of it on text data, as far as we know. The GMF technique can be further specified in individual non-probabilistic methods. We have applied GMF to RLSI and NMF, obtaining Group RLSI (GRLSI) and Group NMF (GNMF), and theoretically demonstrated that GRLSI and GNMF are much more efficient and scalable than RLSI and NMF in terms of time complexity.

We have conducted experiments on two large datasets to test the performances of GRLSI and GNMF. Both datasets contain about 3 million documents. Experimental results show that GRLSI and GNMF are much faster and scalable than existing methods such as RLSI and NMF, especially when the number of topics is large. We have also verified that GMF can discover meaningful topics and the topics can be used to improve search relevance. As future work, we plan to implement GMF on distributed systems and perform experiments on even larger datasets.

9. REFERENCES

[1] S. Bengio, F. Pereira, and Y. Singer. Group sparse coding. In *NIPS*, pages 82–89, 2009.

[2] P. N. Bennett, K. M. Svore, and S. T. Dumais. Classification-enhanced ranking. In *WWW*, pages 111–120, 2010.

[3] D. Blei. Introduction to probabilistic topic models. *COMMUN ACM*, to appear, 2011.

[4] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, pages 121–128, 2008.

[5] D. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SISC*, 20:33–61, 1998.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J AM SOC INFORM SCI*, 41:391–407, 1990.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *ANN STAT*, 32:407–499, 2004.

[9] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *ANN APPL STAT*, 1:302–332, 2007.

[10] W. J. Fu. Penalized regressions: The bridge versus the lasso. *J COMPUT GRAPH STAT*, 7:397–416, 1998.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57, 1999.

[12] S. Lacoste-Julien, F. Sha, and M. I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *NIPS*, pages 897–904, 2008.

[13] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:391–407, 1999.

[14] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS 13*, pages 556–562. 2001.

[15] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808. 2007.

[16] H. Lee and S. Choi. Group nonnegative matrix factorization for eeg classification. In *AISTATS*, pages 320–327, 2009.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *CVPR*, 2008.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, pages 1033–1040. 2009.

[19] D. M. Mimno and McCallum. Organizing the oca: Learning faceted subjects from a library of digital books. In *JCDL*, pages 376–385, 2007.

[20] B. A. Olshausen and D. J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *VISION RES*, 37:3311–3325, 1997.

[21] M. Osborne, B. Presnell, and B. Turlach. A new approach to variable selection in least squares problems. *IMA J NUMER ANAL*, 2000.

[22] D. Ramage, C. D. Manning, and S. Dumais. Partially labeled topic models for interpretable text mining. In *SIGKDD*, pages 457–465, 2011.

[23] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *TREC'3*, 1994.

[24] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, 1975.

[25] F. Wang, N. Lee, J. Sun, J. Hu, and S. Ebadollahi. Automatic group sparse coding. In *AAAI*, pages 495–500, 2011.

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

[27] X. Wei and B. W. Croft. Lda-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185, 2006.

[28] C. Zhai, A. Velivelli, and B. Yu. A crosscollection mixture model for comparative text mining. In *SIGKDD*, pages 743–748, 2004.

Appendix

Proof sketch of Proposition 1. The proof will follow closely the proof given in [14] for the case $\mathbf{Y} = \mathbf{0}$. First note that the objective is decomposable in the rows of $\mathbf{A}$. Considering the case of a single row, denoted as $\bar{\boldsymbol{a}}$, leads to the objective

$$F\left(\bar{\boldsymbol{a}}\right) = \left\|\bar{\boldsymbol{x}} - \bar{\boldsymbol{y}} - \mathbf{S}^T \bar{\boldsymbol{a}}\right\|_2^2,$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{y}}$ are the corresponding rows of $\mathbf{X}$ and $\mathbf{Y}$ respectively. Define the auxiliary function $G\left(\bar{\boldsymbol{a}}, \bar{\boldsymbol{a}}^t\right)$ as

$$G\left(\bar{\boldsymbol{a}}, \bar{\boldsymbol{a}}^t\right) = F\left(\bar{\boldsymbol{a}}\right) + \left(\bar{\boldsymbol{a}} - \bar{\boldsymbol{a}}^t\right)^T \nabla_{\bar{\boldsymbol{a}}} F\left(\bar{\boldsymbol{a}}^t\right) + \left(\bar{\boldsymbol{a}} - \bar{\boldsymbol{a}}^t\right)^T \boldsymbol{\Omega}\left(\bar{\boldsymbol{a}}^t\right)\left(\bar{\boldsymbol{a}} - \bar{\boldsymbol{a}}^t\right),$$

where $\boldsymbol{\Omega}\left(\bar{\boldsymbol{a}}^t\right)$ is a diagonal matrix defined as

$$\omega_{ij}^t = \delta_{ij} \frac{\left(\mathbf{S}\mathbf{S}^T \bar{\boldsymbol{a}}^t\right)_i + \left(\mathbf{S}\bar{\boldsymbol{y}}\right)_i}{\left(\bar{\boldsymbol{a}}^t\right)_i}.$$

Here, δ_{ij} is equal to 1 if $i = j$ and 0 otherwise. Then the update rule can be derived using the methods in [14]. □