Information Sciences 222 (2013) 147–162

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Fast dimension reduction for document classification based on imprecise spectrum analysis

Hu Guan [a,*], Jingyu Zhou [a], Bin Xiao [b], Minyi Guo [a], Tao Yang [c]

[a] Department of Computer Science, SJTU, Dongchuan 800, Shanghai, PR China
[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong
[c] Department of Computer Science, University of California at Santa Barbara, USA

ARTICLE INFO

*Article history:*
Received 25 March 2011
Received in revised form 20 July 2012
Accepted 24 July 2012
Available online 31 July 2012

*Keywords:*
Feature selection
Latent Semantic Indexing
Singular Value Decomposition
Dimension reduction
Imprecise spectrum analysis

ABSTRACT

Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD) is an effective dimension reduction method for document classification and other information analysis tasks. The computational overhead of SVD is known to be a bottleneck in dealing with large data sets, and faster dimension reduction with competitive accuracy is desired in such a setting.

This paper presents Imprecise Spectrum Analysis (ISA) to carry out fast dimension reduction for document classification. ISA follows the one-sided Jacobi method for computing SVD and simplifies its intensive orthogonality computation. It uses a representative matrix composed of top-$k$ column vectors derived from the original feature vector space and reduces the dimension of a feature vector by computing its product with this representative matrix. The paper provides an analysis to show the approximation error and the rationale behind such a dimension reduction method. To further improve classification accuracy, this paper also presents a feature selection method in building the initial feature matrix and augments the representative matrix by including centroid vectors. Our extensive experimental results show that ISA is fast in handling large term-document feature matrices while delivering better or competitive classification accuracy for the tested benchmarks compared to LSI with SVD.

## 1. Introduction

Latent Semantic Indexing (LSI) [12] based on Singular Value Decomposition (SVD) [9,20] is a powerful tool for the linear spectral analysis of documents [49], and is commonly used in dimension reduction for document classification [11,5], query filtering [43], and content indexing [3].

The classical methods for calculating SVD on dense matrices include QR-based algorithms [42] and Jacobi-like algorithms [17,48,41,34,39]. When a term-document matrix becomes sparse, subspace iterative methods can deliver faster results [42]. SVDPACKC [8] is a popular package that implements those fast methods. Random projection [32] presents another way for LSI to cope with a sparse matrix and FJLT [2] has an $O(nlogn)$ complexity. Jacobi method has been shown to be more accurate numerically than the QR method [13], but is much slower. In [16], new optimization for the one-sided Jacobi method is developed for computing SVD with relatively higher accuracy while being faster or competitive to the QR-iteration-based approach. Even with the advancement of these computational methods, recent surveys [33,37] reveal that it is still quite

* Corresponding author. Tel./fax: +86 02134206679.
*E-mail addresses:* guanhu@sjtu.edu.cn (H. Guan), zhou-jy@cs.sjtu.edu.cn (J. Zhou), csbxiao@polyu.edu.hk (B. Xiao), guo-my@cs.sjtu.edu.cn (M. Guo), tyang@cs.ucsb.edu (T. Yang).

0020-0255/$ - see front matter Crown Copyright © 2012 Published by Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.07.032

expensive to compute SVD in LSI applications when very large matrices are involved. Sparsification [18] has been proposed to introduce more zero elements to the reduction matrix in SVD. There is a good speed improvement, yet it takes time to complete SVD process with a sparsified matrix.

In this paper, we propose a scheme called ISA with imprecise spectrum analysis to conduct a fast dimension reduction while retaining the classification accuracy competitive to SVD. The key idea of this approach is to simplify the one-sided Jacobi method and avoid the most time-consuming step in this method for orthogonality computation. The simplified method essentially selects top-*k* vectors from the original feature matrix. We call this matrix a representative matrix. In carrying out the dimension reduction, we multiply this representative matrix with a given feature vector. Selecting top-*k* vectors has a small approximation error in representing the original feature matrix, and using the multiplication of this representative matrix with a feature vector is equivalent to using another known mapping method in SVD scheme. We provide an analysis to explain these two points in supporting the design of ISA.

We further propose two methods to improve the classification accuracy for ISA. First, in the construction of the initial feature matrix, we exploit the availability of training data by incorporating class-aware weights and making inter-class feature vectors more orthogonal. Second, we include class-based centroid vectors in the representative matrix with an error analysis and the extended scheme called EISA provides an improved representation of the original feature space. As a result, our scheme can deliver good accuracy while reducing the dimension of the feature matrix in a complexity proportional to the number of non-zero elements in the initial matrix.

The remainder of the paper is organized as follows. Section 2 reviews the background, summarizes related work and gives design considerations. Section 3 elaborates the design. Section 4 presents evaluation results. Finally, Section 5 concludes our work with future research directions.

## 2. Background and related work

Comparing with traditional feature selection methods [5], such as information gain [28], mutual information [36,47], Chi-square [45], odds ratio [40,37] and other latest methods [30,29,31], LSI for feature selection performs noise reduction and has a potential benefit to detect synonyms as well as words that refer to the same topic [12,4,1]. Moreover, dimension reduction with SVD can not only greatly reduce the high dimensionality, but also improve the precision and efficiency of document classification systems [5]. Generally, an original vector $d$ can be mapped into a low dimension vector $d_{lsi}$ in LSI space (pseudo-document) with the following standard mapping formula [12,7,19]:

$$d_{lsi}^T = d^T U \Lambda^{-1}, \tag{1}$$

where $U$ is an $m \times r$ matrix, $\Lambda$ is a diagonal matrix of rank $r$ derived through SVD computation. The pseudo-document vector $d_{lsi}$ is of $k$ dimensions, much smaller than the original vector $d$'s dimension. In literature, however, many different mapping formulas are used for various applications. For example, the mapping formula

$$d_{lsi}^T = d^T U, \tag{2}$$

was used in query application [32] and orthogonal centroid [27]. Meanwhile, Yan et al. [43] compared $d_{lsi}^T = d^T U \Lambda^{-1}$ and $d_{lsi}^T = d^T U$ with

$$d_{lsi}^T = d^T U \Lambda, \tag{3}$$

in query comparison, and then suggested $d_{lsi}^T = d^T U \Lambda^{1.35}$. We will develop a mapping method which has a comparable performance as $d_{lsi}^T = d^T U \Lambda$, but with a much faster performance in the matrix derivation process.

We prefer to view LSI for feature selection with one-sided Jacobi theory, and our method is motivated by observing computing steps in the one-sided Jacobi method. Given the original feature matrix $H$, the one-sided Jacobi algorithm obtains the orthogonal matrix $B$ by a series of iterative plane rotation [24,35], which can be stated into the form $HV = B$ for a real matrix $H_{m\times n}$ of rank $r$ and matrix $V$ represents a sequence of the orthogonalization process. Post-multiplying $H$ with $V$ can yield matrix $B_{col} = HV$ whose columns are orthogonal, while pre-multiplying $H$ with $U^T$ can obtain a row-orthogonal matrix $B_{row}^T = U^T H$. The intensive computation arises from the step $B = HV$ or $B_{row}^T = U^T H$.

Among SVD algorithms, although Lanczos [10] in popular SVDPACKC [8], is much faster than traditional Jacobi methods, it is also slow when dealing with large matrices. Although Jacobi SVD is more accurate numerically than QR-based SVD, it is non-scalable due to the bottleneck of convergence speed [15] for applications involving large matrices.

There are many methods to reduce the overhead of computing SVDs. Among them, parallelization has been proposed in many studies [17,6]. Approximation is proposed in [21], which is based on the relaxation on the precision of orthogonal vectors. The Jacobi–Davidson method [38] is a popular technique to compute a few eigenpairs of large sparse matrices. A recent survey [33] reveals that the Jacobi algorithm is not suggested in computing SVD in LSI applications when very large matrices are involved.

Our algorithm is developed for text classification where training data is available. The feature matrix for dimension reduction is constructed using classified training data. Is there any characteristic of training data that can be used for optimizing and speeding up the SVD computation or approximation? When performing a one-sided Jacobi algorithm on a sparse matrix $H$ from the 20-newsgroup corpus to get an orthogonal matrix $B$, we found that before the first sweep [34,39], there were only

5% non-orthogonal pair vectors and this sparse matrix only has 5% of non-zero elements. Then many sweeps of the Jacobi method are spent with a plane rotation on getting a 100%-orthogonal matrix $B$, which is extremely time consuming. This uneconomic practice encouraged us to challenge the fundamental problem: can LSI for feature selection perform well when we relax the rigorously orthogonal constraint on matrix $B$?

## 3. Design

Our approach is to relax the orthogonal constraint of $B$ in one-sided Jacobi algorithm. Meanwhile, we only make those pair-vectors from different classes to be orthogonal as much as possible by constructing the initial matrix $H$ with a special score. Because an analysis in the LSI space [32] shows that those pair-document-vectors from different topics should be nearly orthogonal, and then LSI does a particularly good dimension reduction for the classification.

With this in mind, we can consider key pair-vectors in $H$ are nearly orthogonal, and thus we remove the time-consuming matrix-rotation-process in the Jacobi computation. Then the one-sided Jacobi computation can be simplified as the following steps:

- Construct matrix $H$ and then use matrix $H$ as an imprecise matrix $B'$ in one-sided Jacobi algorithm.
- Compute the $l_2$ norm of all column vectors in $B'$, sort $B'$ vectors according to norm, and form into the diag-norm-matrix $N$.
- Get an imprecise decomposition by $B' = U'N$ for dimension reduction, similar to the last step $B = U\Lambda$ in one-sided Jacobi algorithm.

To carry out dimension reduction with $U'$ and $N$, we can select top-$k$ norms from $N$ and take these norms as top-$k$ scalar values in the diagonal matrix $\Lambda$, and use corresponding normalized vectors $U'_k$ as the matrix $U_k$ in one-sided Jacobi algorithm. These imitating steps can be stated into $d_{isa}^T = d^T U'_k N_k = d^T B'_k = d^T H_k$, and such an approach is called Imprecise Spectrum Analysis (ISA) in this paper because of imitation of LSI based on an imprecise decomposition.

The rest of this section will first present a method to construct the initial feature matrix $H$ based on a given training data set. Then it provides an analysis to show our simplified mapping method for dimension reduction has an equivalent effect for SVM-based classification compared to $d_{lsi}^T = d^T U\Lambda$. Finally it presents an extension to ISA to further improve classification accuracy.

### 3.1. Construct initial matrix H

Different applications may use a different scoring scheme to build matrix $H$ for their classification tasks. While ISA can support and apply the mapping process to a given $H$, we offer a scheme to construct an initial matrix to improve classification accuracy. The basic idea is that two document vectors in different classes have an orthogonal trend. To do this, we extend CFC score [23] by exploiting summarized class information extracted from training data.

The terms used in this construction is defined in Table 1. Our scoring formula to construct a term-document matrix $H$ is CFC-E [22], where a weight $h_{i,j}$ for a term $t_i$ and a document $j$ in the matrix $H$ is defined as $e_i \times cf_i$ where

$$\begin{aligned} cf_i &= log\left(\frac{|C|}{CF_i}\right), \\ e_i &= 1 + \frac{1}{\log n}\sum_{j=1}^{n}\frac{c_{i,j}}{c_i}\log\frac{c_{i,j}}{c_i}. \end{aligned} \tag{4}$$

In CFC-E, $cf_i$ is a class-based score measuring the number of times that the term $t_i$ has appeared in different classes in the training data, and entropy $e_i$ for the $i$th term is representing its distinctness in the corpus. Both global scores prefer those rare terms in corpus, and the combination of $cf_i$ and $e_i$ increases the inter-class-orthogonal tendency for those pair-vectors from different topics. The inter-class-orthogonal tendency can be easily verified by counting the number of sweeps in obtaining orthogonal centroid-vectors with one-sided Jacobi algorithm. If centroid vectors are constructed with CFC-E score in our benchmarks, the number of sweeps is reduced to be less than 25 in general. Without integration of CFC-E scoring, the number of sweeps would be over 50, which demonstrates the effectiveness of CFC-E scoring.

**Table 1**
Term definitions.

| Term | Description |
|---|---|
| $n$ | Total # of document vectors |
| $\|C\|$ | Total # of classes |
| $c_{i,j}$ | # Of times that term $t_i$ occurs in the $j$th document |
| $c_i$ | $= \sum_{j=1}^{n} c_{i,j}$, $t_i$'s Occurrences in corpus |
| $CF_i$ | # Of classes containing $t_i$ in training data |

In the implementation, the CFC-E score can be used as a filtering element considering other feature scores, such as term frequency, TF-IDF. For applications owning a small number of classes, a smoothing factor $\mu$ can be added to the class-based weight $cf_i$, i.e., $cf_i = log\left(\frac{|C|+\mu}{CF_i}\right)(0 < \mu < 1)$. After construction of the initial matrix $H$, we use this matrix $H$ as the target matrix $B$ in one-sided Jacobi algorithm to carry out dimension reduction.

### 3.2. Rationale of imprecise spectrum analysis

After construction of the original matrix $H$ using the method described in Section 3.1 or another method the application chooses, the derived computation method can be justified by comparing its classification accuracy to the dimension reduction with $d_{lsi}^T = d^T U\Lambda$. Specifically, we show that $d_{lsi}^T = d^T U\Lambda$ and $d_{isa}^T = d^T H$ have an equivalent effect for classification using the popular Support Vector Machine method (SVM).

Given an SVM classifier, it has the same dual Wolfe optimization problem $L_D$:

$$L_D = \sum \alpha_i - \frac{1}{2}\sum\sum \alpha_i \alpha_j y_i y_j d_i^T d_j, \tag{5}$$

where $y_i$ is the label of the vector $d_i$,$\alpha_i$ is the Lagrange multipliers. If dimension reduction is carried out in LSI space, the part $d_i^T d_j$ will be replaced by $(d_{lsi})_i^T (d_{lsi})_j$. If we use the mapping formula $d_{lsi}^T = d^T U\Lambda$ in dimension reduction, the part $d_i^T d_j$ in Eq. 5 will be substituted by $d_i^T U\Lambda(U\Lambda)^T d_j$. In the Singular Value Decomposition $H = U\Lambda V^T$ for an initial matrix $H$, we have the transformation

$$HH^T = U\Lambda V^T (U\Lambda V^T)^T = U\Lambda V^T V\Lambda U^T.$$

Because $V^T V = I$, we have $HH^T = U\Lambda(U\Lambda)^T$. Consequently, the part $d_i^T d_j$ can be replaced as $d_i^T U\Lambda(U\Lambda)^T d_j = d_i^T HH^T d_j$. In other words, the mapping formula $d_{isa}^T = d^T H$ has an equivalent effect as $d_{lsi}^T = d^T U\Lambda$ for SVM [22].

### 3.3. Error bound of ISA

After construction of the original matrix $H$, ISA $\left(d_{isa}^T = d^T H_k\right)$ selects top-$k$ vectors $H_k$ from $H$ to carry out dimension reduction according to top-$k$ norms in the diag-norm-matrix $N$. In this section, we analyze the bound of errors raised from this approximation $H_k \approx H$ with Frobenius norm [25], which is a popular method for error analysis on a matrix's approximation. Given an $m \times n$ matrix $H$, let $CH_i$ denote the $i$th column of matrix $H$. The $l_2$ norm of this column vector $CH_j$ is

$$\|CH_j\| = \sqrt{\sum_{i=1}^m h_{i,j}^2}.$$

The Frobenius norm of $H$ is defined as:

$$\|H\|_F = \sqrt{\sum_{j=1}^n \sum_{i=1}^m h_{i,j}^2} = \sqrt{\sum_{j=1}^n \|CH_j\|^2}.$$

The relative error ratio of top-$k$ vectors to approximate $H$ can be given as

$$\frac{\|H - H_k\|_F}{\|H\|_F} = \sqrt{1-r} \leqslant \sqrt{1 - \frac{k}{n}}, \tag{6}$$

where $r = \frac{\|H_k\|_F^2}{\|H\|_F^2}$, and our error bound is smaller than the approximation SVD [46], which uses norm to select top-$k$ vectors as $H$. We prove this bound as follows:

**Proof.** Let $l_2$ norms of $\|CH_1\|, \|CH_2\|, \ldots, \|CH_n\|$ be sorted in a non-increasing order as $L_1, L_2, \ldots, L_n$. Thus,

$$\|H - H_k\|_F^2 = \sum_{j=k+1}^n L_j^2,$$

and

$$\|H_k\|_F^2 = \sum_{j=1}^k L_j^2.$$

Because

$$(n-k)\sum_{j=1}^k L_j^2 \geqslant (n-k)kL_{k+1}^2 = k((n-k)L_{k+1}^2) \geqslant k\sum_{j=k+1}^n L_j^2,$$

we have

$$n\|H_k\|_F^2 = n\sum_{j=1}^{k} L_j^2 \geqslant k\sum_{j=1}^{k} L_j^2 + k\sum_{j=k+1}^{n} L_j^2 = k\|H\|_F^2.$$

Then the ratio

$$r = \frac{\|H_k\|_F^2}{\|H\|_F^2} \geqslant \frac{k}{n},$$

and

$$\frac{\|H - H_k\|_F}{\|H\|_F} = \sqrt{1-r} \leqslant \sqrt{1-\frac{k}{n}}. \qquad \square$$

In imprecise spectrum analysis, because the original matrix $H$ is normally a sparse matrix, a pseudo-document $d_{isa}$ will also be a sparse vector. A classification algorithm such as SVM can take advantages of sparsity to reduce computation cost.

### 3.4. Integration of centroid vectors

Our ISA method uses $H_k$ to represent the original feature space and we call this matrix $H_k$ as the representative matrix. We are seeking an improvement of matrix $H_k$ that can better represent vectors in the original data space.

Motivated by centroid-based methods [14,26,27], we add $|C|$ centroid vectors into $H_k$ to enhance $H_k$'s representation under a very small dimensionality. A centroid vector for the $j$th class is a term vector calculated as

$$\overrightarrow{\text{Centroid}}_j = \frac{1}{\ln|C_j|}\sum_{\vec{d}\in C_j}\vec{d}, \tag{7}$$

where $C_j$ is the training vectors in the $j$th class, and $|C_j|$ is the number of those training vectors in $C_j$. In the constructing formula, we select a natural-logarithm-averaging algorithm $\frac{1}{\ln(|C_j|)}$ for term frequency score to overcome the weakness lying in a simple averaging arithmetic method [23], which can introduce noise.

When centroid vectors are integrated into ISA (EISA), we select top $k - |C|$ vectors as follows.

- Compute the $l_2$ norm for column vectors in $H$, sort $H$ with $l_2$ norm, and form into the diag-norm-matrix $N$.
- Select $k - |C|$ vectors from sorted $H$ according to top $k - |C|$ norms in $N$.
- Combine $|C|$ centroid vectors and $k - |C|$ vectors into the matrix $U'$ for dimension reduction.

We now show the relative error introduced when we select top $k$ normalized vectors along the centroid vectors. Table 2 lists the notations used in our error analysis. The sequence vectors from $k - |C| + 1$ to $k$ in $H$ form into the matrix $H_c$. The main difference between $H_k^*$ and $H_k$ is that $H_k^*$ has $|C|$ centroid vectors $H_{centroid}$ without $H_c$, while $H_k$ is the top-$k$ vectors from $H$ containing $H_c$.

In Fig. 1, notations are illustrating the matrix-components in Table 2. Comparing with the initial matrix $H$ of ISA, the initial matrix $H^*$ of EISA is the matrix $H$ including $H_{centroid}$, so the error analysis will be carried on $\frac{\|H_k^*\|_F}{\|H^*\|_F}$ and $\frac{\|H^*-H_k^*\|_F}{\|H^*\|_F}$.

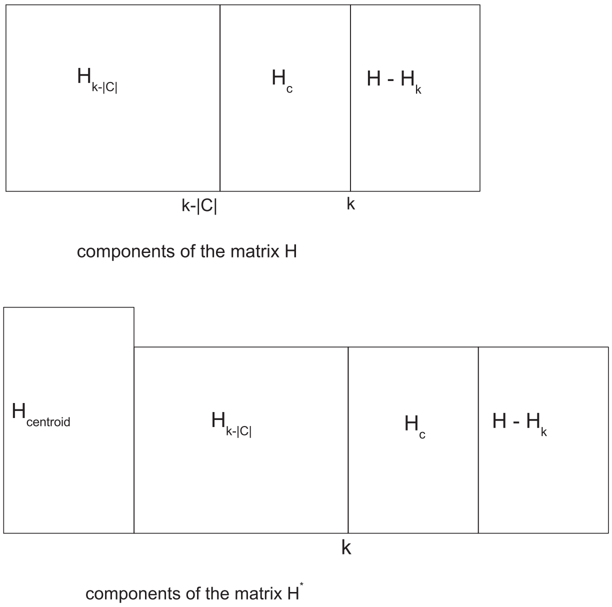
The following analysis shows that for a small $k$, the relative error $\frac{\|H^*-H_k^*\|_F}{\|H^*\|_F} \leqslant \frac{\|H-H_k\|_F}{\|H\|_F}$.

**Proof.** As defined in our aforementioned definitions, let

$$e_{eisa} = \frac{\|H^* - H_k^*\|_F^2}{\|H^*\|_F^2} = \frac{\|H_c + H - H_k\|_F^2}{\|H + H_{\text{centroid}}\|_F^2},$$

**Table 2**
Notations used in error analysis.

| Term | Description |
|---|---|
| $H_{centroid}$ | Centroid vectors |
| $H^*$ | H with centroid vectors |
| $\lvert C_i\rvert$ | # Of document vectors in the $i$th training class |
| $H_{k-\lvert C\rvert}$ | Top $k - \lvert C\rvert$ vectors of $H$ sorted by norms |
| $H_c$ | Sequential vectors from $k - \lvert C\rvert + 1$ to $k$ in $H$ sorted by norms |
| $H_k^*$ | $H_{\text{centroid}}$ combined with top $k - \lvert C\rvert$ sorted vectors ($H_{k-\lvert C\rvert}$) |
| $e_{eisa}$ | Error ratio for EISA |
| $e_{isa}$ | Error ratio for ISA |



components of the matrix H

components of the matrix H*

**Fig. 1.** Matrix-components of the matrix $H$ and $H^*$ in error-bound analysis.

and

$$e_{isa} = \frac{\|H - H_k\|_F^2}{\|H\|_F^2}.$$

Given a comparison between $e_{isa}$ and $e_{eisa}$, we can get

$$\frac{e_{eisa}}{e_{isa}} = \frac{\|H_c + H - H_k\|_F^2}{\|H - H_k\|_F^2} \times \frac{\|H\|_F^2}{\|H + H_{\text{centroid}}\|_F^2}.$$

According to our aforementioned definitions, we have

$$\|H_k\|_F^2 = \sum_{j=1}^{k} L_j^2 \leqslant kL_1^2,$$

$$\|H - H_k\|_F^2 = \sum_{j=k+1}^{n} L_j^2 \geqslant (n-k)L_n^2, \quad \text{and}$$

$$\|H_c\|_F^2 = \sum_{i=k-|C|+1}^{k} L_i^2 \leqslant \sum_{i=k-|C|+1}^{k} L_{k-|C|+1}^2 = |C|L_{k-|C|+1}^2.$$

Let $p = k - |C| + 1$. Thus,

$$\frac{\|H_c + H - H_k\|_F^2}{\|H - H_k\|_F^2} = 1 + \frac{\|H_c\|_F^2}{\|H - H_k\|_F^2} \leqslant 1 + \frac{|C|L_p^2}{(n-k)L_n^2}. \tag{8}$$

At the same time, we have

$$\|H_{\text{centroid}}\|_F^2 = \sum_{i=1}^{|C|}\|\overrightarrow{\text{Centroid}_i}\|^2 = \sum_{i=1}^{|C|}\frac{1}{\ln^2|C_i|}\sum_{j=1}^{|C_i|}\overrightarrow{d_j}\cdot\left(\sum_{j=1}^{|C_i|}\overrightarrow{d_j}\right)^T \geqslant \sum_{i=1}^{|C|}\frac{1}{\ln^2|C_i|}\sum_{j=1}^{|C_i|}\|\overrightarrow{d_j}\|^2 \geqslant \frac{1}{\ln^2\max|C_i|}\sum_{i=1}^{|C|}\sum_{j=1}^{|C_i|}\|\overrightarrow{d_j}\|^2$$
$$\geqslant \frac{1}{\ln^2\max|C_i|}\|H\|_F^2.$$

In the above process, $\sum_{j=1}^{|C_i|}\overrightarrow{d_j}\cdot\sum_{j=1}^{|C_i|}\overrightarrow{d_j}^T = \sum_{j=1}^{|C_i|}\|\overrightarrow{d_j}\|^2 + \sum_{i\neq j}\overrightarrow{d_i}\cdot\overrightarrow{d_j}^T$. Because document vectors are constructed with positive TF-IDF or CFC-E scores, $\overrightarrow{d_i}\cdot\overrightarrow{d_j}^T \geqslant 0$. Thus, we have $\sum_{j=1}^{|C_i|}\overrightarrow{d_j}\cdot\sum_{j=1}^{|C_i|}\overrightarrow{d_j}^T \geqslant \sum_{j=1}^{|C_i|}\|\overrightarrow{d_j}\|^2$. For the training data that contains multi-label documents, the same document vector appears in a number of categories. Thus, $\sum_{i=1}^{|C|}\sum_{j=1}^{|C_i|}\|\overrightarrow{d_j}\|^2 \geqslant \|H\|_F^2$, because $\|H\|_F^2$ only counts a multi-label document vector once. Then,

$$\frac{\|H+H_{\text{centroid}}\|_F^2}{\|H\|_F^2} = 1+\frac{\|H_{\text{centroid}}\|_F^2}{\|H\|_F^2} \geqslant 1+\frac{1}{\ln^2\max|C_i|} \geqslant 1+\frac{1}{\ln^2 n}. \tag{9}$$

Combining Eqs. (8) and (9), we can get

$$\frac{e_{eisa}}{e_{isa}} = \frac{\|H_c+H-H_k\|_F^2}{\|H-H_k\|_F^2}\times\frac{\|H\|_F^2}{\|H+H_{\text{centroid}}\|_F^2} \leqslant \frac{|C|L_p^2+(n-k)L_n^2}{(n-k)L_n^2}\times\frac{\|H\|_F^2}{\|H+H_{\text{centroid}}\|_F^2} \leqslant \frac{|C|L_p^2+(n-k)L_n^2}{(n-k)L_n^2}\times\frac{\ln^2 n}{1+\ln^2 n} = \frac{\frac{\ln^2 n}{1+\ln^2 n}}{\frac{\frac{(n-k)L_n^2}{|C|L_p^2}}{1+\frac{(n-k)L_n^2}{|C|L_p^2}}}.$$

Because $\frac{x}{1+x}$ is a monotonously increasing function when $x \geqslant 0, \frac{e_{eisa}}{e_{isa}} \leqslant 1$ only if $\frac{(n-k)L_n^2}{|C|L_p^2} \geqslant \ln^2 n$. Thus, when $k \leqslant n - \frac{L_p^2}{L_n^2}|C|\ln^2 n, e_{eisa} \leqslant e_{isa}$. □

Although EISA's error is also bounded by $\sqrt{1-\frac{k}{n}}$, EISA has an advantage for a small $k$ compared to ISA. We will validate this in our experiments.

## 4. Evaluation

In this section, we present our evaluation with following objectives:

- Demonstrate the effectiveness of ISA and EISA for dimension reduction for document classification.
- Substantiate the effectiveness of the mapping formula $d_{lsi}^T = d^T U\Lambda$ and show ISA has comparable performance.
- Show that ISA and EISA generate sparse vectors that can be exploited for very fast computation. We also evaluate the impact of integrating centroid vectors in EISA and show performance improvement over ISA.
- Illustrate CFC-E feature scoring is an effective technique for ISA.

To study the performance of different mapping formulas in classifying tasks, we use an SVM classifier to carry out classifying tasks, and use standard macro-averaging F1 (MacroF1) and micro-averaging F1 (MicroF1) as the accuracy metrics to evaluate the performance. The following Table 3 shows the needed elements in the definition.

While MacroF1 and MicroF1 [44,37] are respectively defined as follows:

$$MacroF1 = \frac{2\sum_{i=1}^{|C|}r_i\sum_{i=1}^{|C|}p_i}{|C|\left(\sum_{i=1}^{|C|}p_i+\sum_{i=1}^{|C|}r_i\right)},$$

and

$$MicroF1 = \frac{2\sum_{i=1}^{|C|}TP_i}{\sum_{i=1}^{|C|}FP_i+\sum_{i=1}^{|C|}FN_i+2\sum_{i=1}^{|C|}TP_i}.$$

According to the definition on MacroF1 and MicroF1, MacroF1 gives the same weight to all categories, thus, MacroF1 is mainly influenced by the performance of rare categories for the corpus liking Reuters-21578 and WebKB, due to skewed category distribution. On the contrary, in a corpus, MicroF1 will be dominated by the performance of those categories owning a large scale of documents. So the union of MacroF1 and MicroF1 allows us to appraise an SVM classifier's performance thoroughly.

Table 4 lists the names of the algorithms and approaches compared in this section. As the latest review [5] demonstrates that LSI based on SVD can improve the performance of the document classification, we only compare EISA and ISA with LSI based on SVD.

**Table 3**
Notations for different metrics.

| Names | Descriptions |
|---|---|
| $TP_j$ | # Of documents correctly assigned |
| $FN_j$ | # Of documents falsely accepted |
| $FP_j$ | # Of documents falsely rejected |
| $p_j$ | Precision: $p_j = TP_j/(TP_j + FP_j)$ |
| $r_j$ | Recall: $r_j = TP_j/(TP_j + FN_j)$ |
| $F1_j$ | F1: $F1_j = 2p_jr_j/(r_j + p_j)$ |

**Table 4**
Different approaches.

| Names | Descriptions |
|---|---|
| Base | Without dimension reduction |
| SVD-1 | $d_{lsi}^T = d^T U \Lambda^{-1}$ |
| SVD | $d_{lsi}^T = d^T U$ |
| SVD1 | $d_{lsi}^T = d^T U \Lambda$ |
| ISA | $d_{isa}^T = d^T H$ |
| EISA | $d_{isa}^T = d^T H^*$ |

### 4.1. Dataset and experimental settings

#### 4.1.1. WebKB

The WebKB dataset[1] contains web pages gathered from university computer science department. The pages are divided into seven categories: student, faculty, staff, course, project, department and other. After the parsing process, there are 8203 pages (7031 training and 1172 testing) left in the corpus.

For the "Title","H1", and "URL" parts in the page, we give the weight 5 times more than those in "Body". We kept 28,473 unigram terms that occurred at least once in the training set.

#### 4.1.2. 20-Newsgroup

This dataset from 20 Usenet newsgroups [2] consists of 19,997 text messages (about one thousand text messages per category), and approximately 4% of the articles are cross-posted. The stop words list has 823 words, and we kept words that occurred at least once and messages that had at least one term.

Altogether, there are 19,899 messages (13,272 training and 6637 testing) left in the corpus. We only keep "Subject", "Keywords", and "Content". The total number of unigram terms is 31,138. Words in "Subject" and "Keywords" are given the weight 5 times more than those in "Contents".

#### 4.1.3. Reuters-21578

The Reuters-21578 news dataset[3] is based on the Trinity College Dublin version. After removing all unlabelled documents and documents with more than one class labels, we only retained categories that had at least one document in both the training and testing sets and got a collection of 52 categories.

Totally, there are 6495 training texts and 2557 testing texts left in this 52-category corpus. After removing 338 stop words (also provided by Trinity College dataset), and unigram terms that occur less than three times, we got 11,430 unique unigram terms. Words in titles are given the weight 5 times more than those in abstracts.

For the above three corpora, we used the tokenizer tool provided in the Trinity College sample. Stemming and word clustering were not applied. Entropy score for initial matrix $H$ are extracted from the whole corpus. Original document vectors are constructed with TF-IDF score.

#### 4.1.4. Parameter settings

The SVMTorch package[4] and $SVM^{multiclass}$ package[5] are used in classifying tasks. The two SVM classifiers can perform multiclass tasks directly with one-*vs*-others decomposition and default parameter values.

As for SVD computation, we use LAS [10] algorithm in the latest SVDLIBC library.[6] The latest work on LAS [10] shows that LAS has a highest performance for dimension reduction if there is no loss of $H$'s information. Following suggestions from other works on SVD [10,1], in constructing $H$, all document vectors in the corpus are involved. However, the class information is only extracted from the training set. Experiments were performed on a machine with Intel Core2 Duo 2.8 GHz CPU, 4 GB memory.

### 4.2. Overall comparison

We give an overview on the performance of six different approaches:[7] Base, SVD-1, SVD, SVD1, ISA, and EISA. In the spectrum of dimensions, we selected the delegate point that has the best performance on MacroF1, or has the biggest sum of MacroF1 and MicroF1. In this experiment, different scores for the initial matrix $H$ are used in different benchmarks. That can validate

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data.
[2] http://kdd.ics.uci.edu/databases/20newsgroups.
[3] http://ronaldo.cs.tcd.i.e./esslli07/sw/step01.tgz.
[4] http://www.idiap.ch/~bengio/projects/SVMTorch.html.
[5] http://svmlight.joachims.org/svm_multiclass.html.
[6] http://tedlab.mit.edu/~dr/SVDLIBC/.
[7] Source code is available at http://epcc.sjtu.edu.cn/jzhou/research/is/.

**Table 5**
Overall performance comparison of different methods.

| corpus | Method | Dim | MicroF1 | MacroF1 |
|---|---|---|---|---|
| Reuters | Base | 11430 | 0.9245 | 0.7957 |
| $h_{i,j} = e_i \times cf_i$ | SVD-1 | 200 | 0.9183 | 0.7733 |
| | SVD | 700 | 0.9183 | 0.7966 |
| | SVD1 | 800 | 0.9316 | 0.8014 |
| | ISA | 5000 | 0.9280 | 0.8028 |
| | EISA | 800 | 0.9206 | 0.7810 |
| 20-Newsgroup | Base | 31138 | 0.8430 | 0.8436 |
| $h_{i,j} = tf_{i,j} \times e_i$ | SVD-1 | 3000 | 0.8169 | 0.8169 |
| | SVD | 8000 | 0.8424 | 0.84323 |
| | SVD1 | 6000 | 0.8350 | 0.8367 |
| | ISA | 12000 | 0.8407 | 0.8429 |
| | EISA | 6000 | 0.8507 | 0.8529 |
| WebKB | Base | 28473 | 0.7504 | 0.6506 |
| $h_{i,j} = e_i$ | SVD-1 | 1200 | 0.7736 | 0.7042 |
| | SVD | 1300 | 0.7959 | 0.7129 |
| | SVD1 | 1100 | 0.7916 | 0.6926 |
| | ISA | 1300 | 0.7854 | 0.7224 |
| | EISA | 2000 | 0.7976 | 0.7406 |

$*$Previous work's summary [33] on WebKB has reported that, *F*-measure scores could be varying in the scope of [0.488,0.757] without feature selection.

the effectiveness of EISA and ISA more thoroughly in different settings. Later, we will evaluate the impact of using different scoring methods.

Table 5 shows that SVD, SVD1 and EISA have the best performance in this comparison. Although SVD-1 can achieve its best performance in small dimensions, SVD-1 achieves lower scores than other methods. ISA uses more dimensions and has better performance than SVD-1 and SVD in Reuters and 20-newsgroups. EISA enhances on ISA and has the best performance in 20-newsgroup and WebKB.

### 4.3. Effectiveness of SVD1

This section studies the effectiveness of SVD1 for dimension reduction. In the experiment, the precision in classification tasks is used as the performance metric of different mapping formulas.

For Reuters-21578, Fig. 2 shows that SVD and SVD1 have produced a comparable performance and they outperform SVD-1 significantly, especially for a larger dimension size. Fig. 3 illustrates the performance of the WebKB dataset. For MicroF1, the difference between SVD and SVD1 is relatively small, and is getting bigger with larger dimension size. After dimension is bigger than 2000, SVD exhibits more stable performance while SVD-1's performance drops. For MacroF1, SVD is slightly better than SVD1, and SVD1 is sometime slightly better, especially for a larger dimension. Overall speaking, performance of SVD1's performance is close to SVD and they perform better than SVD-1 for a higher dimension.

The reason of the inferior performance of SVD-1 can be attributed to the inverse weight $\sigma_i^{-1}$ in mapping formula $d_{lsi}^T = d^T U \Lambda^{-1}$. Because imprecision of SVD calculations, $\Lambda_k^{-1}$ creates large numerical errors for those small scalar values. Fig. 4 shows that in the experiment $\sigma_{max}$ and $\sigma_{min}$ in $\Lambda$ are 35.9358 and $3.0126E{-}12$, respectively. At high dimensions, the imprecise $\sigma_i$ results in large numerical errors and causes lower performance for SVD-1. In contrast, SVD and SVD1 did not suffer from the imprecise singular values and have more stable performance on high dimensions.

We also study the impact of diag-norm-matrix *N* in ISA. Experimental results show that scalar values does not fluctuate too much, $n_{max}$ and $n_{min}$ are 9.4189 and 0.2897, respectively. Thus, the impact of the diag-norm-matrix *N* in ISA is more stable.

Fig. 5 demonstrates how different forms of *N* impact on the performance of ISA. We use $d_{isa}^T = d^T H$ to match $d_{lsi}^T = d^T U \Lambda$, use $d_{isa}^T = d^T H N^{-1}$ to match $d_{lsi}^T = d^T U$, and use $d_{isa}^T = d^T H N^{-2}$ to match $d_{lsi}^T = d^T U \Lambda^{-1}$. Compared with SVDs, Fig. 5 shows that the impact of different forms of *N* in ISA is small because the norm in *N* does not change too much.

### 4.4. SVD1 vs. ISA

Fig. 6 shows a comparison of ISA with SVD1 for the WebKB dataset. It shows that ISA does not perform well when the dimension size of reduced space is small. However, ISA can have a better performance than SVD1 after the dimension increases to over 2000. Fig. 7 shows a similar trend for the Reuters-21578 dataset. ISA can become competitive to SVD1 after dimension 5000. If we use the full size of *H*, Fig. 6 and Fig. 7 are showing that ISA can perform as well as SVD1.

The results are expected as we have discussed earlier. A small number of top-*k* vectors is not sufficient to represent the original matrix or original training data. We address this in our algorithm design in two aspects: (1) integrate centroid vectors to capture more class-oriented characteristics of the original data, which will be evaluated in next subSection and (2) use

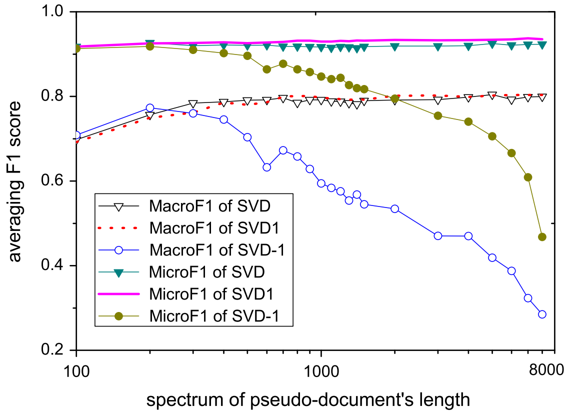


**Fig. 2.** Performance of three mapping formulas for dimension reduction in Reuters.

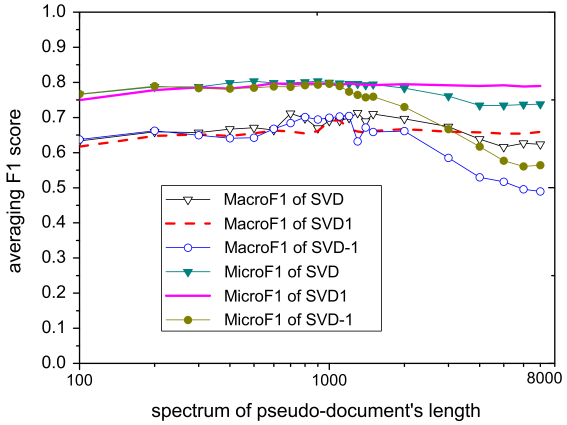


**Fig. 3.** Performance of three mapping formulas for dimension reduction in WebKB.

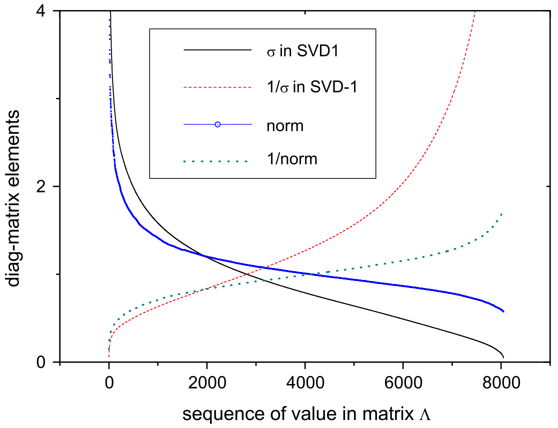


**Fig. 4.** Different elements in diag-matrix in Reuter-21578. Scalar values are sorted in a non-increasing order.

a larger size for dimension reduction while still sustaining lower processing time by exploiting sparsity of document vectors produced in our scheme. This will be also evaluated in next subsection.

Before we move on evaluating our improvements to deal with small sizes in dimension reduction, we conduct one experiment to check if ISA can perform comparably with SVD1 if applying an SVD directly on top-$k$ vectors. Table 6 lists the results of comparison for the Reuters dataset. For example, we select the top $k = 1000$ vectors in matrix $H$ according to the norm of

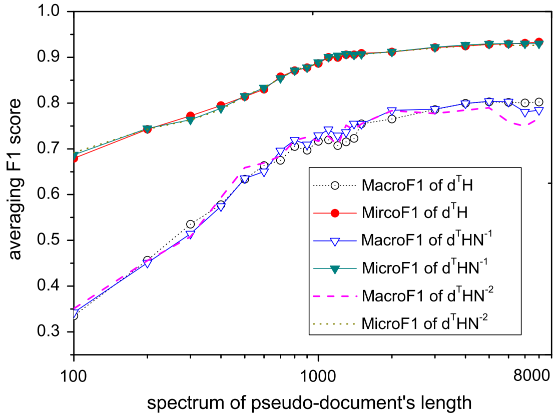


**Fig. 5.** The impact of different forms of Norm on ISA in Reuters-21578.

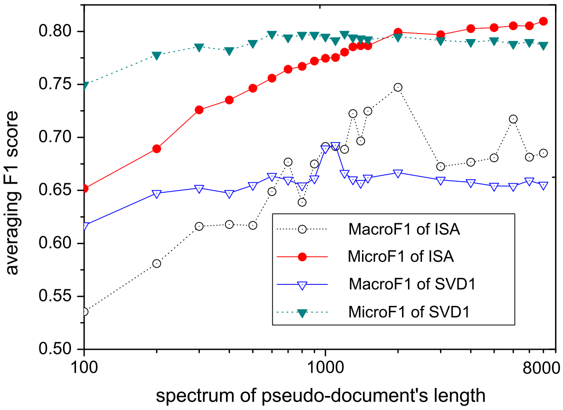


**Fig. 6.** SVD1 vs. ISA in WebKB.

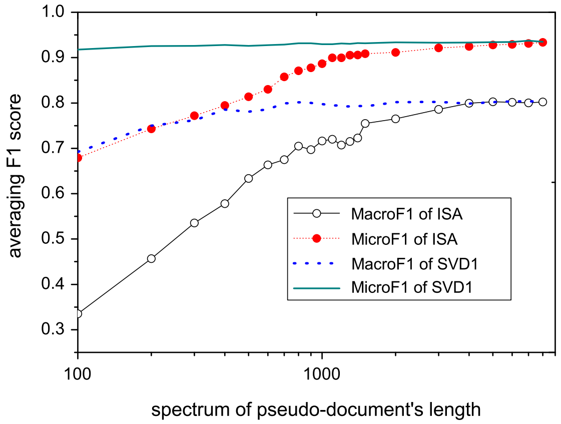


**Fig. 7.** SVD1 vs. ISA in Reuters-21578.

those vectors, then, LAS2 is used to carry out singular value decomposition for $H_{1000}$. $d_{lsi}^{T} = d^{T}U\Lambda$ is used to get the largest dimension of pseudo-document vectors, while ISA use $H_{1000}$ fully in its mapping formula $d_{isa}^{T} = d^{T}H_{1000}$. Table 6 shows that ISA can still perform as well as SVD1 not only in high dimensions but also in low dimensions if SVD is applied in selected top vectors.

**Table 6**
Performance comparison between SVD1 and ISA in Reuters-21578 after SVD is applied to top-*k* vectors.

| $H_k$ | ISA | | SVD1 | |
|---|---|---|---|---|
| | MicroF1 | MacroF1 | MicroF1 | MacroF1 |
| $k = 1000$ | 0.8868 | 0.7162 | 0.8858 | 0.7164 |
| $k = 2000$ | 0.9116 | 0.7651 | 0.9116 | 0.7653 |
| $k = 3000$ | 0.9214 | 0.7858 | 0.9221 | 0.7866 |
| $k = 4000$ | 0.9245 | 0.7996 | 0.9245 | 0.7996 |
| $k = 5000$ | 0.9280 | 0.8028 | 0.9280 | 0.8001 |
| $k = 6000$ | 0.9292 | 0.8016 | 0.9292 | 0.8010 |

## 4.5. Sparsity of pseudo-documents and a comparison of ISA and EISA

### 4.5.1. Sparsity

Pseudo-document vectors derived by ISA are sparse and we can exploit this characteristic for fast computation even using a bigger value $k$ for top-$k$ vector selection. We measure the sparsity using density ratio, which is defined as the number of nonzero values in all pseudo-document vectors divided by the total number of values, namely, $|nonzeros|/(m \times k)$. In this formula, $m$ is the number of pseudo documents in the reduced space, and $k$ is the dimension length of pseudo-documents.

Table 7 shows that, when the dimension is small, multiplication of many original vectors with top-$k$ vectors has results to be zero and thus pseudo-document vectors are zeroed out for ISA. Furthermore, a small $H_k$ will generate an obvious approximation error, as illustrated in Fig. 7. The gap between ISA and SVD1 becomes larger as the dimension decreases.

Centroid vectors have a positive effect on classification tasks as discuss below. Though EISA has a little denser pseudo-document vector than ISA, EISA can eradicate zero-out defects in ISA because centroid vectors can capture more non-zero features through feature summation, as shown in Table 7. Moreover, integrating the centroid concept in ISA can improve its overall accuracy, especially for a small $H_k$, which has been shown in Fig. 8.

### 4.5.2. EISA vs. ISA

Fig. 8 shows performance difference among EISA, ISA and SVD1. When the dimension size is small, top-$k$ vectors in ISA do not provide enough representation of all necessary features and thus ISA has a weak accuracy in such a case. On the other hand, EISA integrates centroid vectors that capture more important features and has much better accuracy. When the dimension size $k$ becomes large, top-$k$ vectors provide enough representation of necessary features for all classes and thus the gap between ISA and EISA is narrowed.

Fig. 9 illustrates that EISA can remedy ISA's weakness in low dimensions elegantly—EISA significantly outperforms SVD1, ISA, and SVD. The trend of MicroF1 and MacroF1 is close because 20-newsgroup is a much balanced corpus. Thus, we omit the performance results for MicroF1 here.

## 4.6. Processing time cost

For LSI-based classification, two key steps are dimension reduction computation and training using SVM. We assess and compare the time spent for different methods in terms of dimension reduction and SVM training.

### 4.6.1. Processing time of dimension reduction computation

EISA took less than one second to perform matrix decomposition for three corpora. For LAS2, Table 8 illustrates the cost of a few choices of targeted dimensions and processing time is extremely slow for larger dimensions. If matrix size becomes

**Table 7**
Nonzero density of pseudo-document vectors in 20-newsgroup using ISA or EISA.

| Dim | Nonzero density | | # Of zero vectors | |
|---|---|---|---|---|
| | ISA | EISA | ISA | EISA |
| 1000 | 0.0091 | 0.0256 | 2739 | 0 |
| 2000 | 0.0125 | 0.0210 | 447 | 0 |
| 4000 | 0.0194 | 0.0236 | 35 | 0 |
| 6000 | 0.0257 | 0.0284 | 4 | 0 |
| 8000 | 0.0313 | 0.0335 | 0 | 0 |
| 10000 | 0.0366 | 0.0386 | 0 | 0 |
| 12000 | 0.0417 | 0.0434 | 0 | 0 |
| 16000 | 0.0551 | 0.0565 | 0 | 0 |
| 20000 | 0.0759 | 0.0764 | 0 | 0 |

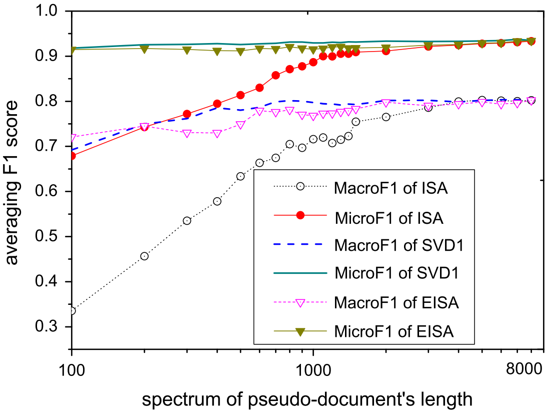


**Fig. 8.** SVD1 vs. ISA and EISA in Reuters-21578.

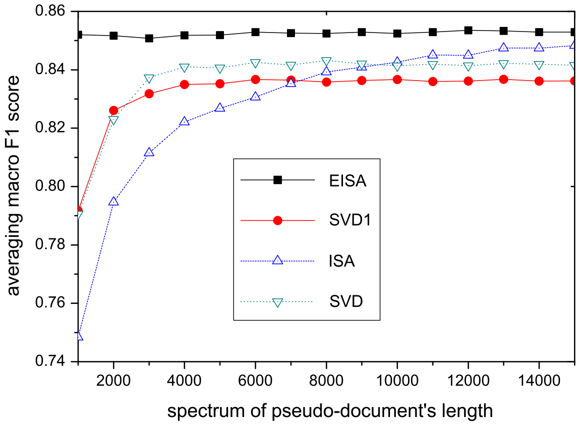


**Fig. 9.** EISA vs. SVD, ISA and SVD1 in 20-newsgroup.

**Table 8**
Elapsing seconds of LAS2 and EISA for matrix computation with different targeted dimensions.

| Dim | Reuters | | 20-Newsgroups | |
|---|---|---|---|---|
| | LAS2 | EISA | LAS2 | EISA |
| 1000 | 270 | 0.39 | 298 | 0.67 |
| 3000 | 3340 | 0.39 | 4266 | 0.67 |
| 6000 | 7757 | 0.39 | 32059 | 0.67 |

bigger, LAS2 time would be much more time-consuming. EISA only needs to sort the initial matrix $H$ after normalization in addition to compute centroid for each document class in the training data, so EISA has a much lower time cost.

### 4.6.2. Training time for classification

20-newsgroup corpus has 13,272 training test messages. In this experiments, for each sample dimension, the training process had been carry out 10 times with $SVM^{multiclass}$. We extracted the smallest time–cost for LAS2 while the biggest time–cost for EISA. Table 9 shows that SVD's vectors with dimension 3000 need 9.77 s in training while ISA's results with dimension 12,000 need only 5.22 s. The pseudo-document of LSI is a dense vector while the pseudo-document of ISA or EISA is a sparse vector. EISA's cost is as small as ISA because the zero density of EISA's pseudo-document is close to ISA's.

## 4.7. Different scores for H

In this experiment, we illustrate the impact of different feature scoring methods on the classification accuracy for ISA. Specifically, we compare classification accuracy using 0–1, term frequency filtering by entropy ($tf_{i,j} \times e_i$), term frequency

**Table 9**
CPU time of SVM training cost in seconds for 20-newsgroup.

| LAS2 | | ISA | | EISA | |
|---|---|---|---|---|---|
| dim | Cost | Dim | Cost | Dim | Cost |
| 1000 | 3.70 | 3000 | 1.42 | 3000 | 1.47 |
| 2000 | 6.28 | 6000 | 1.81 | 6000 | 1.86 |
| 3000 | 9.77 | 10000 | 4.43 | 10000 | 4.72 |
| 4000 | 13.35 | 12000 | 5.22 | 12000 | 5.23 |
| 5000 | 16.69 | 15000 | 6.28 | 15000 | 6.35 |
| 6000 | 20.03 | 19897 | 8.30 | 19897 | 8.35 |

**Table 10**
MicroF1 comparison with different term-document weights in ISA in Reuters-21578.

| Dimension | $tf_{i,j} \times e_i$ | 0–1 | $tf_{i,j}$ | $tf_{i,j} \times e_i \times cf_i$ |
|---|---|---|---|---|
| 1000 | 0.8736 | 0.8850 | 0.8814 | 0.7592 |
| 2000 | 0.8936 | 0.8940 | 0.8956 | 0.8662 |
| 3000 | 0.9096 | 0.8936 | 0.8979 | 0.9073 |
| 4000 | 0.9163 | 0.8956 | 0.8972 | 0.9097 |
| 5000 | 0.9143 | 0.8983 | 0.9007 | 0.9186 |
| 6000 | 0.9155 | 0.9022 | 0.8999 | 0.9257 |
| 7000 | 0.9147 | 0.9011 | 0.8999 | 0.9241 |
| 8000 | 0.9151 | 0.9108 | 0.8995 | 0.9253 |

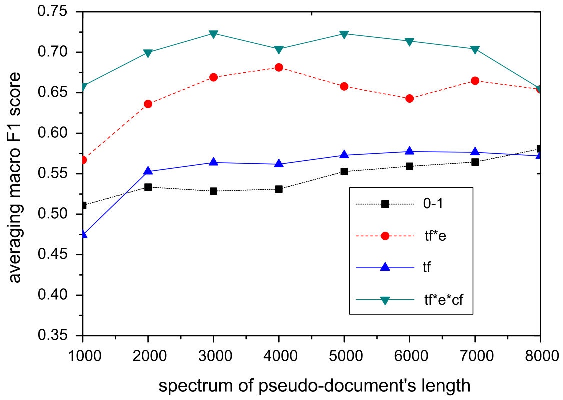


**Fig. 10.** MacroF1 on Reuters-21578 using different term-document weights in ISA.

$(tf_{i,j})$, and term frequency filtering by CFC-E $(tf_{i,j} \times e_i \times cf_i)$. Table 10 shows that all these methods worked even we use 0–1 score for an initial matrix $H$ because there exists a functional equivalence between ISA and SVD1.

However, Fig. 10 shows that CFC-E can perform better than other methods on MacroF1 score.

CFC-E is a global filtering score composed with entropy and class frequency $cf_i$. It is well known that entropy prefers those rare occurrence terms in the classification, so $tf_{i,j} \times e_i$ score has a better performance than $tf_{i,j}$ or 0–1 score in Fig. 10. $cf_i$ utilizes class-oriented information from training data to make vectors from different classes to be more orthogonal, which enhances the preference to those rare terms in corpus. Thus, pseudo-document vectors produced by CFC-E in ISA could be more discriminative to a classifier, and ISA with CFC-E score may have a better performance in the classifying tasks.

Because ISA has a functional equivalence to SVD1, it is possible that the global scores have the same effect on SVDs. This encourages us to assess impact of using those scores in the original SVD computation for classification.

Fig. 11 illustrates MacroF1 results for SVD $(d_{lsi}^T = d^T U)$ when the above scoring scheme is used. In this experiment, we selected the "alt.atheism", "misc.forsale", "rec.autos", and "talk.politics.guns" from 20-newsgroups. We observe that term-frequency filtering by CFC-E, term-frequency filtering by entropy, and term-frequency perform comparably after 1000. However, $tf_{i,j} \times e_i \times cf_i$ or $tf_{i,j} \times e_i$ is more effective than $tf_{i,j}$ on lower dimensions ($k < 1000$). This demonstrates that a global score is more effective than term-frequency for feature selection in the original SVD computation.

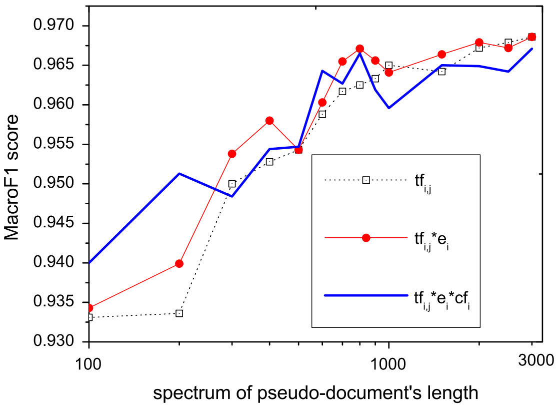


**Fig. 11.** MacroF1 on the 4-category test benchmark from 20-newsgroup corpus using different term-document weights in SVD.

## 5. Conclusions

To conduct dimension reduction quickly, we propose Imprecise Spectrum Analysis (ISA) and its extension (EISA) to carry out this process. Our approach can be as accurate as SVD-based LSI in document classification while outperforming significantly in terms of speed. EISA can achieve a better performance than ISA, especially in lower dimensions.

Three tested benchmarks show that our approach achieves significant dimension reduction and a classification learning algorithm such as SVM can exploit sparse computation to get low training time. Our experiments on different constructing scores for $H$ show that an excellent global score, such as entropy and CFC-E, can boost the performance of our approach.

Our tested benchmarks are focused on text classification. It will be interesting to investigate the effectiveness of ISA and EISA for other applications which benefit from fast dimension reduction. Another issue that can be studied is the optimal choice of $k$ in selecting top-$k$ vectors for the ISA and EISA methods.

## Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments on this paper. This work was supported in part by NSFC 61003012, 863 Program of China 2011AA01A202, and Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT), China, and by NSF IIS-1118106 in USA. The views expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

## References

[1] M. Ahat, S. Amor, M. Bui, S. Jhean-Larose, G. Denhiere, Document classification with lsa and pretopology.
[2] N. Ailon, B. Chazelle, Faster dimension reduction, Communications of the ACM 53 (2010) 97–104.
[3] G. Almpanidis, C. Kotropoulos, I. Pitas, Combining text and link analysis for focused crawling – an application for vertical search engines, Information Systems 32 (2007) 886–908.
[4] R. Ando, 2000. Latent semantic space: iterative scaling improves precision of inter-document similarity measurement, in: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 216–223.
[5] B. Baharudin, L. Lee, K. Khan, A review of machine learning algorithms for text-documents classification, Journal of Advances in Information Technology 1 (2010) 4–20.
[6] M. Becka, G. Oksa, On variable blocking factor in a parallel dynamic block–Jacobi svd algorithm, Parallel Computing 29 (2003) 1153–1174.
[7] J. Bellegarda, Exploiting latent semantic information in statistical language modeling, Proceedings of the IEEE 88 (2000) 1279–1296.
[8] M. Berry, Multiprocessor Sparse SVD Algorithms and Applications, Ph.D. thesis. Citeseer, 1991.
[9] J. Cadzow, Svd representation of unitarily invariant matrices, IEEE Transactions on Acoustics, Speech and Signal Processing 32 (1984) 512–516.
[10] J. Chen, Y. Saad, Lanczos vectors versus singular vectors for effective dimension reduction, IEEE Transactions on Knowledge and Data Engineering (2008) 1091–1103.
[11] M. Chen, L. Chen, C. Hsu, W. Zeng, An information granulation based data mining approach for classifying imbalanced data, Information Sciences 178 (2008) 3214–3227.
[12] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society for Information Science 41 (1990) 391–407.
[13] J. Demmel, K. Veselic, et al, Jacobi's method is more accurate than QR, SIAM Journal on Matrix Analysis and Applications 13 (1992) 1204–1245.
[14] I. Dhillon, D. Modha, Concept decompositions for large sparse text data using clustering, Machine Learning 42 (2001) 143–175.
[15] Z. Drmac, Accurate computation of the product-induced singular value decomposition with applications, SIAM Journal on Numerical Analysis (1998) 1969–1994.
[16] Z. Drmač, K. Veselić, New fast and accurate Jacobi svd algorithm. ii, SIAM Journal on Matrix Analysis and Applications 29 (2008) 1343.
[17] G. Gao, S. Thomas, An optimal parallel Jacobi-like solution method for the singular value decomposition, in: Proc. Int. Conf. Parallel Proc, 1988, pp. 47–53.
[18] J. Gao, J. Zhang, Sparsification strategies in latent semantic indexing, in: Proceedings of the 2003 Text Mining Workshop, Citeseer, 2003, pp. 93–103.
[19] S. Gao, W. Wu, C. Lee, T. Chua, A maximal figure-of-merit (mfom)-learning approach to robust classifier design for text categorization, ACM Transactions on Information Systems (TOIS) 24 (2006) 190–218.

[20] D. Gleich, L. Zhukov, Svd based term suggestion and ranking system, in: Data Mining, 2004, ICDM'04, Fourth IEEE International Conference on, IEEE, 2004, pp. 391–394.
[21] J. Gotze, S. Paul, M. Sauer, An efficient Jacobi-like algorithm for parallel eigenvalue computation, IEEE Transactions on Computers (1993) 1058–1065.
[22] H. Guan, B. Xiao, J. Zhou, M. Guo, T. Yang, Fast dimension reduction for document classification based on imprecise spectrum analysis, in: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, 2010, pp. 1753–1756.
[23] H. Guan, J. Zhou, M. Guo, A class-feature-centroid classifier for text categorization, in: Proceedings of the 18th International Conference on World Wide Web, ACM, 2009, pp. 201–210.
[24] M. Hestenes, Inversion of matrices by biorthogonalization and related results, Journal of the Society for Industrial and Applied Mathematics (1958) 51–90.
[25] N. Higham, Accuracy and stability of numerical algorithms, Society for Industrial Mathematics (2002).
[26] G. Karypis, E. Han, Fast supervised dimensionality reduction algorithm with applications to document categorization & retrieval, in: Proceedings of the Ninth International Conference on Information and Knowledge Management, ACM, 2000, pp. 12–19.
[27] H. Kim, P. Howland, H. Park, Dimension reduction in text classification with support vector machines, Journal of Machine Learning Research 6 (2006) 37.
[28] C. Lee, G. Lee, Information gain and divergence-based feature selection for machine learning-based text categorization, Information Processing & Management 42 (2006) 155–165.
[29] G. Lu, Y. Wang, Feature extraction using a fast null space based linear discriminant analysis algorithm, Information Sciences 193 (2012) 72–80.
[30] S. Mitra, P. Kundu, W. Pedrycz, Feature selection using structural similarity, Information Sciences 198 (2012) 48–61.
[31] M. Mohammadi, B. Raahemi, A. Akbari, B. Nassersharif, H. Moeinzadeh, Improving linear discriminant analysis with artificial immune system-based evolutionary algorithms, Information Sciences 189 (2011) 219–232.
[32] C. Papadimitriou, H. Tamaki, P. Raghavan, S. Vempala, Latent semantic indexing: a probabilistic analysis, in: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, ACM, 1998, pp. 159–168.
[33] X. Qi, B. Davison, Web page classification: features and algorithms, ACM Computing Surveys (CSUR) 41 (2009) 1–31.
[34] S. Rajasekaran, M. Song, A novel scheme for the parallel computation of SVDs, Lecture Notes in Computer Science 4208 (2006) 129.
[35] A. Sameh, On Jacobi and Jacobi-like algorithms for a parallel computer, Mathematics of Computation 25 (1971) 579–590.
[36] K. Schneider, Weighted average pointwise mutual information for feature selection in text categorization, Knowledge Discovery in Databases: PKDD 2005 (2005) 252–263.
[37] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys (CSUR) 34 (2002) 1–47.
[38] G. Sleijpen, H. Van der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM Review (2000) 267–293.
[39] M. Soliman, S. Rajasekaran, R. Ammar, A block JRS algorithm for highly parallel computation of SVDs, High Performance Computing and Communications (2007) 346–357.
[40] P. Soucy, G. Mineau, Feature selection strategies for text categorization, Advances in Artificial Intelligence (2003). 993-993.
[41] V. Strumpen, H. Hoffmann, A. Agarwal, A stream algorithm for the SVD (2003).
[42] L. Trefethen, D. Bau, Numerical linear algebra, 50, Society for Industrial Mathematics (1997).
[43] H. Yan, W. Grosky, F. Fotouhi, Augmenting the power of LSI in text retrieval: singular value rescaling, Data & Knowledge Engineering 65 (2008) 108–125.
[44] Y. Yang, X. Liu, A re-examination of text categorization methods, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 42–49.
[45] Y. Yang, J. Pedersen, A comparative study on feature selection in text categorization, in: Machine Learning-International Workshop Then Conference, Citeseer, 1997, pp. 412–420.
[46] D. Zhang, Z. Zhu, A fast approximate algorithm for large-scale latent semantic indexing, in: Digital Information Management, 2008, ICDIM 2008, Third International Conference on, IEEE, 2008, pp. 626–631.
[47] P. Zhili, S. Xiaohu, M. Marchese, L. Yanchun, An enhanced text categorization method based on improved text frequency approach and mutual information algorithm, Progress in Natural Science 17 (2007) 1494–1500.
[48] B. Zhou, R. Brent, A parallel ring ordering algorithm for efficient one-sided Jacobi SVD computations, Journal of Parallel and Distributed Computing 42 (1997) 1–10.
[49] F. Zhuang, G. Karypis, X. Ning, Q. He, Z. Shi, Multi-view learning via probabilistic latent semantic analysis, Information Sciences 199 (2012) 20–30.