Expert Systems with Applications 41 (2014) 861–868

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Minimizer of the Reconstruction Error for multi-class document categorization

Juan Carlos Gomez *, Marie-Francine Moens

Department of Computer Science, KU Leuven, Celestijnenlaan 200A, Heverlee, B-3001, Belgium

ARTICLE INFO

Keywords:
Document categorization
Text mining
Dimensionality reduction
Principal Component Analysis

ABSTRACT

In the present article we introduce and validate an approach for single-label multi-class document categorization based on text content features. The introduced approach uses the statistical property of Principal Component Analysis, which minimizes the reconstruction error of the training documents used to compute a low-rank category transformation matrix. Such matrix transforms the original set of training documents from a given category to a new low-rank space and then optimally reconstructs them to the original space with a minimum reconstruction error. The proposed method, called Minimizer of the Reconstruction Error (mRE) classifier, uses this property, and extends and applies it to new unseen test documents. Several experiments on four multi-class datasets for text categorization are conducted in order to test the stable and generally better performance of the proposed approach in comparison with other popular classification methods.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

The current information era allows users to generate large amounts of digital information. A good proportion of this information is in the form of text documents. Text is still the main way of communication among humans. Such text documents are daily exchanged among individuals, companies, organizations, etc. In this direction, it is generally recognized that document categorization plays an important role in the flow of document interchanges, since it facilitates the tasks of accessing and retrieving relevant information by users and systems. Document categorization is a key component for many practical applications such as digital library management, opinion analysis, and Web search engines. Nevertheless, document categorization is very difficult because of the high dimensionality of document representations to be classified and their content diversity.

In the current work we tackle the problem of single-label multi-class text categorization. This problem is defined as: given a training set of documents $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_c]$, organized in c categories, where $\mathbf{T}_i = [\mathbf{t}_{i,1}, \mathbf{t}_{i,2}, \ldots, \mathbf{t}_{i,m}]$ corresponds to a term-document matrix composed by m documents in the i-th category, and given a test document $\mathbf{y}$; the goal is then to train a model over the set $\mathbf{T}$ which is able to assign *one* of the c categories to the new test document $\mathbf{y}$.

In the single-label multi-class problem the categories are mutually exclusive, that means each document can belong to only one category. This assumption could not occur in some real scenarios, where some documents could belong to more than one category (multi-label problem) at the same time. Nevertheless, a method for single-label multi-class categorization could be generalized and could be transformed into a set of independent binary categorization problems (Sebastiani, 2002), which is in fact the approach of many systems for multi-label categorization (Tsoumakas, Katakis, & Vlahavas, 2010).

In this work, we present a novel text document classifier which relies on and extends the statistical properties of Principal Component Analysis (PCA). In essence the proposed model uses the framework which derives PCA from the minimization of reconstruction error of the training examples. The model then uses such property to classify new unseen documents, using the idea that new documents are better reconstructed by a transformation matrix which was computed from similar documents. We call this method the Minimizer of Reconstruction Error (mRE) classifier. During training, the mRE classifier computes a set of category transformation matrices $\mathbf{W}_i$; $i = 1, 2, \ldots, c$ of rank r by means of PCA. Inside the classifier, such rank r could be learned using a standard k-fold validation over the training set. During testing, given a new unseen document, the model projects such document using each one of the different category matrices, then reconstructs the document using again each one of the matrices and finally computes the reconstruction errors, by measuring the Frobenius norm of the difference between the set of reconstructed documents and the original one. The matrix which produces the minimum error indicates the category to be assigned to the new document.

* Corresponding author.
E-mail addresses: juancarlos.gomezcarranza@cs.kuleuven.be (J.C. Gomez), sien.moens@cs.kuleuven.be (M.-F. Moens).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.08.016

In order to test the validity of the mRE classifier, we perform experiments with several public datasets for text categorization: the Classic dataset, the 20Newsgroups dataset, and the WIPO-alpha (World Intellectual Property Organization) and WIPO-de datasets. We test the model using standard training/test splits of the data. The results show that the mRE classifier gives good and stable results across the different datasets and experiments. In the same direction, with the purpose of having a better overview of the performance of the mRE classifier, we present a comparison for every experiment with three other well known categorization methods: Multinomial Naive Bayes (NB), K-Nearest Neighbors (K-NN) and a linear Support Vector Machine (SVM), which have a very good behavior in text document categorization.

The contributions of our work are the following:

- The feasibility of applying the statistical property of minimizing the reconstruction error with PCA in a single-label multi-class text categorization task, exploiting the sparseness of the category matrices in order to perform a fast training of the model.
- The empirical evidence that mRE is able to properly model the categories of the documents by extracting transformation matrices which represent most of the information from the data in terms of variance and minimization of the reconstruction error of the training documents.
- The evidence that the property of minimizing the reconstruction error could be extended and applied to new unseen documents, where new documents similar to the ones used to compute a given matrix $\mathbf{W}_p$ are better reconstructed by such matrix, allowing in this way to assign the proper category for each new document.
- The evidence that a suitable rank r for the projection matrices $\mathbf{W}_i$ could be learned from the training data using a standard k-fold validation.

The rest of this paper is organized as follows: Section 2 is devoted to recall several statistical and probabilistic works in the field of dimensionality reduction for text document categorization related to mRE. In Section 3 we first give a brief introduction to PCA, from which we derive the mRE model, and secondly we present and describe the general architecture for the mRE classifier. Section 4 illustrates the experimental evaluation framework for the mRE classifier, describing the datasets and setup used during experimentation, and we present the results obtained from the experiments together with a discussion about them. Finally, in Section 5 we present the conclusions and future work.

2. Related research

Since the mRE classifier relies on PCA, it is directly related with several works which employs PCA or similar techniques that transform the original data to a new space, where the variables in this new space are linear combinations of the original features. Using this transformation, it is expected that, under given conditions, the new variables describe in a better way the original data. In the areas of natural language processing, text mining and information retrieval, one of the most known technique is Latent Semantic Analysis (LSA) (also known as Latent Semantic Indexing) (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). LSA uses the term-document matrix of a collection of documents and finds a low-rank approximation of such matrix using Singular Value Decomposition (SVD), producing a set of concepts related to the documents and terms. In matrix terms, LSA is similar to PCA, but without centering the term-document matrix. In line with this approach, most of the works devoted to text categorization where PCA or LSA are used, employ PCA as a first step to project the original term-document matrix to a new low-rank space, considering only the first few components with the highest variance. After this initial dimensionality reduction phase the categorization is performed using standard classification algorithms (e.g., SVM, NB, K-NN, etc.) (Gomez, Boiy, & Moens, 2012; Kim, Howland, & Park, 2005; Li & Jain, 1998; Schütze, Hull, & Pedersen, 1995; Sebastiani, 2002; Weigend, Wiener, & Pedersen, 1999). Contrary to these basic approaches, inside the mRE classifier, we do not use the original training documents projected to a low-rank space to train a model, rather we use the transformation matrices computed from such original training documents as the classification model. The computed transformation matrices are able to optimally compress and reconstruct the original documents used to create the corresponding matrix, and are useful to classify new unseen documents.

Linear Discriminant Analysis (LDA) is a categorization/dimensionality reduction technique (Fisher, 1936), which uses the category information to project the data into a new space where the ratio of between-class-variance to within-class-variance is maximized in order to obtain adequate category separability. LDA could be used as a dimensionality reduction technique similar to PCA, but including the category information to improve the separation between classes in the new space (Anderson, 2003); and could be used as well to perform categorization. LDA categorizes a new unseen document by projecting it into the new space and then its projection is compared with the mean of each projected training category. Torkkola (2001) was one of the first authors to use LDA for text categorization purposes. There, the author mentions that PCA aims at an optimal representation of the data but that it does not help for an optimal discrimination of the data, and then proposes LDA to categorize text documents. Nevertheless LDA as classifier tends to perform worse than a SVM for text classification (Kim et al., 2005). In the present work we actually exploit the discriminative properties of PCA for text categorization, by including the category information in form of a transformation matrix per category, which minimize the reconstruction error of the training documents inside the corresponding category.

Non-Negative Matrix Factorization (NMF) is another dimensionality reduction technique. NMF, similar to PCA, projects the data to a new space, but the values in the transformation matrices obtained with NMF are only positive (Barman, Igbal, & Lee, 2006; Berry, Gillis, & Glineaur, 2009). Similarly, other models such as probabilistic Latent Semantic Analysis (pLSA) (Hoffmann, 2007) and Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003) are probabilistic extensions of the LSA model, and are currently popular as topic representation models. These models reduce the dimensions of the documents by representing them as a mixture of topic distributions and topics as a mixture of words distributions. These models have the disadvantage that identifying the correct number of latent components is a difficult and computationally expensive problem (Blei et al., 2003). In the mRE classifier, we could estimate the proper number of latent components (the rank r of the matrix $\mathbf{W}$), by applying a standard k-fold cross validation over the training set.

As stated before, in this work we propose the mRE classifier, which relies on the framework derived from the property of PCA to minimize the reconstruction error of the training documents used to compute the matrix $\mathbf{W}$ and extend it to apply such property to unseen documents. The same property of PCA has already been used in the computer vision tasks of object detection (Malagón-Borja & Fuentes, 2009) and novelty detection (Hoffmann, 2007), and in a spam filtering task (Gomez & Moens, 2012). However, to the best of our knowledge there is no work devoted to minimize the reconstruction error on single-label multi-class text categorization, where the best rank r for the matrix $\mathbf{W}$ is learned from the training data.

3. Proposed model

In this section we start by giving a brief description of PCA. We then describe the training and test phases of the mRE model, which extends the framework from which PCA is derived, in order to perform categorization of documents.

3.1. PCA

PCA is one of the most popular methods for approximating a given dataset using a linear combination of the original features, producing a compressed representation of the data by using fewer combined features. PCA intends to perform simultaneously dimensionality reduction, minimum mean-square error of approximation and retainment of maximum variance of the original data in the new representation (Jolliffe, 2002). Because of its properties PCA is very useful for tasks such as visualization and feature extraction. PCA was first developed by Pearson (1901) and its statistical properties were investigated in detail by Hotelling (1933). Anderson in Anderson (2003) and Joliffe in Jolliffe (2002) have given some of the most comprehensive expositions of this technique.

There are several frameworks for performing PCA (Miranda, Borgne, & Bontempi, 2008; Ilin & Raiko, 2010). The most popular one is by an analysis of variance. The derivation of PCA in this framework generates the basis by iteratively finding the orthogonal directions of maximum retained variances, such that the greatest variance by any projection of the data comes to lie on the first component, the second greatest variance on the second component, and so on (Hotelling, 1933; Jolliffe, 2002). Another way for deriving PCA is using the minimization of the mean-square error in data compression. This approach looks in the direction of maximizing the decorrelation of the data using orthogonal transformations (Bishop, 2006; Duda, Hart, & Stork, 2001; Ilin & Raiko, 2010). Using the data compression formulation, PCA finds a lower-dimensional linear representation of the data, such that the original data could be reconstructed from the compressed representation with the minimum square error. This second formulation is the one of interest in this work, from which we derive the mRE classifier. The brief formulation of PCA from data compression using documents as data is given below, for a larger explanation the reader is referred to Bishop (2006), Diamantaras and Kung (1996) and Duda et al. (2001).

Let $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_m]$ be a term-document matrix of training data, were $\mathbf{t}_i \in \mathbb{R}^n$ is the i-th document expressed as a *column* vector with n rows. The rows represent the terms from a vocabulary.

PCA looks for a matrix $\mathbf{W} \in \mathbb{R}^{n \times r} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_r]$, composed by a set of $r \leqslant n$ orthogonal *column* vectors, with unit length, which transforms the original documents to a new orthogonal basis. We call such vectors *principal components* (PCs). The following equation expresses the transformation:

$$\mathbf{z}_i = \mathbf{W}^T(\mathbf{t}_i - \vec{\mu}); \quad i = 1, 2, \ldots, m \tag{1}$$

where $\mathbf{z}_i$ is the i-th projected document in the new space, and $\vec{\mu}$ is the mean column vector of the term-document matrix. A mean of zero in this matrix is required for finding a basis that minimizes the mean-square error of the approximation of the data. Inversely, we can express the original documents as:

$$\mathbf{t}_i = \mathbf{W}\mathbf{z}_i + \vec{\mu}; \quad i = 1, 2, \ldots, m \tag{2}$$

Under the data compression approach, PCA finds $\mathbf{W}$ such that the Frobenius norm of the reconstruction is minimized:

$$f = \sum_{i=1}^{m} \|\mathbf{t}_i - (\mathbf{W}\mathbf{z}_i + \vec{\mu})\|_F \tag{3}$$

Now, if we choose $r = n$ as the dimension of $\mathbf{W}$ we do not lose any information during the transformations, and the reconstruction error (i.e., the above Frobenius norm) is zero. However, if we choose $r < n$, some information is lost because not all the variance is present. The choosing of a $r < n$ is common for *data compression* and dimensionality reduction, because the largest variances (and the most interesting dynamics of the data) are associated with the r first components. Then, the error of reconstruction for a given dimension $r < n$ is:

$$e = \sum_{k=r+1}^{n} \mathbf{w}_k^T \mathbf{A} \mathbf{w}_k \tag{4}$$

where $\mathbf{A} = \sum_{i=1}^{m} (\mathbf{t}_i - \vec{\mu})(\mathbf{t}_i - \vec{\mu})^T$ is the covariance matrix of the mean centered data matrix, and $\mathbf{w}_k$ is the k-th column vector of the $\mathbf{W}$ matrix. The minimization of this error corresponds to the following eigenproblem:

$$\mathbf{A}\mathbf{W} = \lambda \mathbf{W} \tag{5}$$

The reconstruction error is then minimized when $\mathbf{w}_i$ are the eigenvectors of the covariance matrix. This will produce the optimal low rank r approximation to $\mathbf{T}$ in the least squares sense, with r as the rank of matrix $\mathbf{W}$, corresponding to non-zero eigenvalues.

3.2. Minimizer of Reconstruction Error classifier

PCA can be seen as a process to reveal the internal structure of the documents that are being analyzed, by means of looking for the set of PCs (the $\mathbf{W}$ matrix or rank r) that best describes the variance or the distribution of the documents, while minimizing the reconstruction error of such documents. Therefore, this matrix $\mathbf{W}$ is going to preserve better the information of the documents on which the PCA was applied.

Since PCA minimizes the reconstruction error of the original documents, the idea behind the current approach is to extend such property to new, unseen documents. Thus, if we have a matrix $\mathbf{W}_p$ that was obtained from a given category p of documents, this $\mathbf{W}_p$ must better reconstruct new unseen documents which are *similar* to the ones used to extract such matrix, and will reconstruct poorly unseen documents which are not similar or closely related to the original ones.

Given such assumptions we build the Minimizer of Reconstruction Error (mRE) classifier, which generalizes the concept of error reconstruction to a single-label, multi-class scenario, where the task is to assign an unseen document to *one* of a predefined set of c categories, by finding the matrix $\mathbf{W}_p$ (and its corresponding category p) that better reconstructs the new document.

Before explaining in detail the mRE classifier, we first consider some practical issues regarding the computation of PCA in order to make the method more efficient. Firstly, extracting the eigenvectors from the covariance matrix is computationally expensive and memory consuming. With text documents, the covariance matrix $\mathbf{A}$ is square with the number n of terms, and for large datasets, with thousands of terms, to store such a large matrix is unfeasible. The computation of the eigenvectors using a traditional algorithm like QZ (Moler & Stewart, 1973) has complexity of $O(n^3)$, which makes it hard to use with large datasets. Instead of computing $\mathbf{W}$ from the covariance matrix, in the mRE classifier we use the relation of PCA with SVD, where the PCs correspond with the singular vectors of a SVD of the normalized centered data: $\mathrm{SVD}((\mathbf{T} - \mathbf{M})/m^{\frac{1}{2}})$, which is computationally more efficient (Shlens).

Secondly, usually the rank r of the matrix $\mathbf{W}$ is chosen to be small in comparison with the total number of terms ($r \ll n$), because only a few components carry most of the information from the data. In this direction, we use here the Lanczos algorithm (Lanczos, 1950), which is an adaptation of the classic Power Method exposed by Hotelling (1933) as a fast SVD calculator to extract the transformation matrices $\mathbf{W}_i$ with a relatively small rank r from the data matrices $\mathbf{T}_i$. The Lanczos algorithm starts from an initial

Table 1
Number of documents and categories for the datasets used for experimentation.

Dataset	Training documents	Test	Categories	No. categories
Classic	4956	2123	CACM,CISI,CRAN,MED	4
20NewsGroups	11293	7528	alt.atheism, comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x misc.forsale, rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey sci.crypt, sci.electronics sci.med, sci.space soc.religion.christian talk.politics.guns talk.politics.mideast talk.politics.misc talk.religion.misc	20
WIPO-alpha	46324	28926	A,B,C,D,E,F,G,H	8
WIPO-de	50555	21271	A,B,C,D,E,F,G,H	8

random matrix $\mathbf{W}_{initial}$ of rank r and iteratively refines it until it reaches convergence using an orthonormalization of the columns (Lanczos, 1950).

Finally, for text document categorization, the term-document matrix $\mathbf{T}$ is highly sparse. Even if during the mean subtraction, part of the sparsity is lost, the use of a global vocabulary (i.e., extracted from all the categories in the training dataset), and the computation of PCA per category, leaves the data matrix still highly sparse. We exploit such sparseness (storing only the non-zero elements and performing multiplications only with such elements) in order to compute efficiently the low rank matrix $\mathbf{W}$.

The general architecture of the mRE method is divided in a training and a test phase. The training phase starts with a set $\mathbf{T} = [\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_c]$ of training documents organized in c categories. $\mathbf{T}_i = [\mathbf{t}_{i,1}, \mathbf{t}_{i,2}, \ldots, \mathbf{t}_{i,m}]$, corresponds to a term-document (category) matrix composed by m document column vectors of size n in the i-th category. The number m changes depending on i (different number of documents inside different categories). Then, we compute the mean vector $\vec{\mu}_i$ for each category i, and we form a matrix $\mathbf{M}_i \in \mathbb{R}^{n \times m}$ by repeating in each column the value of the mean $\vec{\mu}_i$. Afterwards, we subtract the corresponding mean to each category matrix $\mathbf{T}_i - \mathbf{M}_i$ and normalize the result by the square root of the number m of documents. Finally, we define the number r of PCs we want to extract from the data and compute the rank r matrix $\mathbf{W}_i$ for each category i using PCA obtained by SVD. The parameter r defines the quantity of information carried by the matrix $\mathbf{W}$, but not all the information is important in such matrix. Then, the selection of a suitable value for r is important and depends on the properties of the data at hand. Such parameters could be learned in the mRE classifier using standard techniques, like a k-fold cross validation over the training set. The training phase is illustrated in Algorithm 1.

Algorithm 1. Training phase for mRE

Require: - A set of training document matrices
$\mathbf{T} = \{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_c\}$, organized in c categories
- The size of the low rank approximation r
Ensure: - A set of projection matrices $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_c\}$
- A set of mean vectors $\vec{\mu} = \{\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_c\}$. One mean per category
for $i = 1$ to c **do**
$\vec{\mu}_i \leftarrow$ `ComputeVectorMean`$(\mathbf{T}_i)$
$\mathbf{M}_i \leftarrow$ `RepeatMeanInColumns`$(\vec{\mu}_i)$
$\mathbf{T}_i \leftarrow (\mathbf{T}_i - \mathbf{M}_i)/m^{\frac{1}{2}}$
$\mathbf{W}_i \leftarrow$ `SVD` $(\mathbf{T}_i, r)$
end for

The test phase of the mRE classifier is as follows: starting with a new unseen document $\mathbf{y} \in \mathbb{R}^{n \times 1}$ to be classified, expressed as a column vector, for each category i we first extract from the new document the category mean vector $\vec{\mu}_i$. We then project the example using the corresponding transformation matrix $\mathbf{W}_i$. Afterwards, we reconstruct back the document using the same matrix $\mathbf{W}_i$ and we add again the category mean vector $\vec{\mu}_i$. We then compute the reconstruction error as the Frobenius norm of the difference between the original example and the reconstructed one. The process is repeated for each category $i = 1, 2, \ldots, c$. Finally, the classifier assigns the example to the category with the minimum reconstruction error. This process is detailed in Algorithm 2.

Algorithm 2. Testing phase for mRE

Require: - A set of projection matrices $\mathbf{W}_i$; $i = 1, 2, \ldots, c$
- A set of mean vectors $\vec{\mu}_1, \vec{\mu}_2, \ldots, \vec{\mu}_c$
- A new unseen document $\mathbf{y}$
Ensure: - A predicted category $p \in \mathbf{C}$ for $\mathbf{y}$.
for $i = 1$ to c **do**
$\mathbf{p}_i \leftarrow \mathbf{y} - \vec{\mu}_i$
$\mathbf{z}_i \leftarrow \mathbf{p}_i \mathbf{W}_i$
$\mathbf{s}_i \leftarrow \mathbf{z}_i \mathbf{W}_i^T$
$\mathbf{q}_i \leftarrow \mathbf{s}_i + \vec{\mu}_i$
$e_i \leftarrow$ `FrobeniusNorm`$(\mathbf{y}, \mathbf{q}_i)$
end for
$p \leftarrow$ `IndexofMinimum`$(e_1, e_2, \ldots, e_c)$

4. Experimental evaluation

4.1. Datasets

The following section contains brief descriptions of the datasets used to test the validity of the mRE classifier.

- *Classic*: This dataset[1] is a collection of abstracts from journals in different areas split in 4 categories. We split this dataset in 70% of the documents for training and the remaining 30% for testing.
- *20Newsgroups*: This dataset is a collection of approximately 20,000 newsgroup documents, evenly split across 20 different newsgroups: In this work we use the *bydate* version of this dataset,[2] which already has a train/test split.
- *WIPO-alpha*: This dataset[3] is in English and is a collection of patent applications submitted to the World Intellectual Property (WIPO). Each patent application includes a title, a list of inventors, a list of applicant companies or individuals, an

[1] Available at: http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/.
[2] Available at: http://people.csail.mit.edu/jrennie/20Newsgroups/.
[3] Available at: http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/index.html.

abstract, a claims section, and a long description. The dataset used here is split in the 8 sections of the International Patent Classification (IPC) code. This dataset already has a train/test split defined by WIPO.

- *WIPO-de*: Similarly to WIPO-alpha, this dataset[4] is a collection of patent applications submitted to WIPO but in German. It is also split in the 8 sections of the IPC code. This dataset already has a train/test split defined by WIPO.

Table 1 presents a summary of the datasets shown above. This table includes the used split of training/test documents, the actual names of the categories and the number of categories. Table 2 shows some statistics regarding the category distributions of the datasets. The last two columns in this table contain information about the degree of skewness of the category distributions. The former shows the ratio between the major and minor categories. Higher ratios result in more skewed category distributions. The latter shows the Shannon entropy values (Shannon, 1948); higher values of entropy imply more uncertainty in the distribution.

Additionally to the use of the complete train/test split for each dataset, we also use fractions of the training set from each dataset to build a model and the complete test set to test such model. This scenario of using only a fraction of the training set is used in order to test the ability of a given model to generalize the content of the different categories. In general the categories with few documents are difficult to model given the lack of information to build a robust classifier. In this paper we select portions containing 1%, 5%, 10%, 20%, 30% and 50% of the total training set.

For the complete training set and the portions we apply a pre-selection of terms in order to extract a vocabulary. We remove first the stop-words using the Rainbow list (Mccallum, 1996) and later we remove the terms that appear in five documents in the training set or in the given portion. This pre-selection helps to remove noisy terms but at the same time avoids an aggressive selection, which could remove important features. Moreover, it is known that for classifiers that exhibit a good performance in text categorization, like a SVM or NB, the use of a large vocabulary is beneficial to be able to map between terms and categories (Joachims, 1998). For the WIPO-alpha and WIPO-de dataset, which are structured documents, we extract only the information in the title and abstract fields, plus the first 30 lines of the long description, in order to avoid the verbosity of some documents. Table 3 shows the number of terms for each portion of the training set for each dataset.

4.2. General experimentation setup

The term-category matrices were built by vectorizing the text documents using a typical term-frequency inverse-document-frequency (tf-idf) schema, considering the vocabulary extracted from the training part of each dataset as the terms. When only a fraction of the training set is considered for training a model, the vocabulary is extracted from such fraction. The tf-idf vectors are normalized to the unit using norm 2. The documents are in the columns and the terms in the rows. All the methods start the training process using directly the normalized tf-idf term-document matrices.

In order to have a better overview of the performance of the mRE classifier, we compare its results with the ones obtained using other popular classification algorithms in text categorization. We used for comparison three classifiers: Multinomial Naive Bayes (NB), which is fast and performs well in text categorization, K-Nearest Neighbors (K-NN), and a linear Support Vector Machine (SVM) which is known by its very good performance with sparse data and which is especially well suited for text categorization.

[4] Available at: http://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/index.html.

Table 2
Statistics on the number of documents for the datasets used for experimentation.

Dataset	Documents per Category					
	Average	Min	Median	Max	Max/Min	Entropy
Classic	1769.75	1017	1429.0	3204	3.15	2.0
20NewsGroups	941.05	628	983.0	999	1.59	4.22
WIPO-alpha	9406.25	1710	10632.0	16245	9.50	3.0
WIPO-de	8978.25	2162	8361.0	19092	8.83	3.0

Table 3
Number of terms per percentage of the training set.

Dataset	Percentage of Training Set (%)						
	1	5	10	20	30	50	100
Classic	15	458	1021	1860	2522	3559	5485
20NewsGroups	187	2161	4085	7107	9595	13251	20650
WIPO-alpha	2706	8181	11831	16344	19860	25357	35366
WIPO-de	904	4733	8905	16155	22123	33094	55769

We use for these baseline methods the implementations from the Weka package (Hall et al., 2009) using Java to call the methods programmatically. In order to find the optimal parameters C for SVM (C is the regularization parameter, a larger C corresponding to assign a higher penalty to errors) and K for K-NN (K is the number of nearest neighbors considered to assign the category to a test document), we performed a 5-fold cross validation over the training set with different parameter values: $C = \{0.1, 1, 10, 100\}$ and $K = \{1, 2, 5, 10\}$ and we chose the one which maximizes the macro-F1 measure.

The model for mRE was implemented in MatLab to take advantage of its efficient matrix operations, but creating a JAR library of the whole model to call the methods from Java. The Lanczos algorithm approximates the matrices $\mathbf{W}_i$ starting from a random matrix; thus, in order to estimate the confidence level of the approximations, we performed 10 runs with mRE for each experiment. Similarly to the use of the SVM and the K-NN classifiers, with the mRE classifier we performed a 5-fold cross validation over the training data, in order to optimize the r parameter, which corresponds to the rank of the matrices $\mathbf{W}_i$. We considered ranks of $r = \{1, 2, 4, 8, 16, 32, 64, 128\}$, and we chose the one which maximizes the macro-F1 measure.

We compared the performance of all the methods using the standard classification measures accuracy and F1-measure, which are defined as $Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)}$ and $F1 = 2\frac{Precision \cdot Recall}{Precision+Recall}$, with $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, where TP is the number of true positive documents (positive documents categorized as positive), TN the number of true negative documents (negative documents categorized as negative), FP the number of false positive documents (negative documents categorized as positive) and FN the number of false negative documents (positive documents categorized as negative). The accuracy measures the proportion of corrected categorized documents, while the $F1$ measure represents the harmonic mean of precision and recall. We compute the micro and macro averages for $F1$. In our case, given that the categorization is single-label multi-class, the values of $F1$ micro-averaged are equal to the values of accuracy. We conducted all the experiments using a desktop Linux PC with a 3.4 GHz Intel Core i7 processor and with 16 GB of RAM.

4.3. Results

Tables 4–7 show the results of all the classifiers for the Classic, 20Newsgroups, WIPO-alpha and WIPO-de datasets respectively. The tables present in the first column the portion of the training

Table 4
Comparison of performance for the Classic dataset.

Training Set (%)	Accuracy				Macro F1			
	mRE	SVM	NB	K-NN	mRE	SVM	NB	K-NN
1	0.4588	0.5972	**0.6394**	0.5834	0.4604	0.4842	**0.5094**	0.49377
5	0.8695	**0.8847**	0.8366	0.5264	0.8590	**0.8772**	0.8164	0.3508
10	0.8886	**0.9223**	0.8829	0.4730	0.8876	**0.9222**	0.8778	0.2192
20	0.9124	**0.9454**	0.9213	0.4640	0.9157	**0.9476**	0.9221	0.1854
30	0.8944	**0.9510**	0.9359	0.4696	0.8968	**0.9540**	0.9362	0.2017
50	0.9133	**0.9543**	0.9463	0.4673	0.9183	**0.9571**	0.9488	0.1959
100	0.9218	**0.9585**	0.9496	0.4776	0.9267	**0.9608**	0.9523	0.2208

Table 5
Comparison of performance for the 20Newsgroups dataset.

Training Set (%)	Accuracy				Macro F1			
	mRE	SVM	NB	K-NN	mRE	SVM	NB	K-NN
1	0.1530	**0.2261**	0.2340	0.1710	0.1504	**0.2336**	0.2067	0.1689
5	0.4682	**0.5860**	0.5882	0.0791	0.5091	**0.5788**	0.5500	0.0540
10	0.5223	**0.6858**	0.6971	0.0984	0.5627	**0.6815**	0.6638	0.0801
20	0.7345	**0.7496**	0.7417	0.1005	0.7315	**0.7425**	0.7152	0.0905
30	**0.7743**	0.7673	0.7643	0.1012	**0.7691**	0.7608	0.7399	0.0985
50	**0.8066**	0.7926	0.7923	0.1332	**0.8000**	0.7857	0.7692	0.1483
100	**0.8259**	0.8162	0.8069	0.1797	**0.8212**	0.8099	0.7880	0.2112

Table 6
Comparison of performance for the WIPO-alpha dataset.

Training Set (%)	Accuracy				Macro F1			
	mRE	SVM	NB	K-NN	mRE	SVM	NB	K-NN
1	**0.5547**	0.5528	0.4564	0.4707	**0.4234**	0.4459	0.2673	0.3384
5	**0.6985**	0.6554	0.4969	0.4657	**0.6472**	0.6114	0.3048	0.3833
10	**0.7142**	0.6807	0.5278	0.4107	**0.6672**	0.6396	0.3389	0.3476
20	**0.7323**	0.7032	0.5576	0.5618	**0.7052**	0.6693	0.3773	0.5179
30	**0.7354**	0.7086	0.5701	0.5868	**0.7083**	0.6794	0.3945	0.5386
50	**0.7429**	0.7262	0.5891	0.6065	**0.7180**	0.6998	0.4169	0.5775
100	**0.7459**	0.7377	0.6100	0.6429	**0.7256**	0.7142	0.4405	0.6199

Table 7
Comparison of performance for the WIPO-de dataset.

Training Set (%)	Accuracy				Macro F1			
	mRE	SVM	NB	K-NN	mRE	SVM	NB	K-NN
1	**0.3759**	0.3485	0.3075	0.1417	**0.2708**	0.2708	0.1413	0.1096
5	**0.5280**	0.4844	0.3967	0.1817	**0.4640**	0.4298	0.2399	0.1293
10	**0.5673**	0.5336	0.4329	0.2062	**0.5131**	0.4856	0.2826	0.1703
20	**0.5913**	0.5901	0.4704	0.2694	**0.5475**	0.5389	0.3219	0.2155
30	**0.5963**	0.6028	0.4943	0.3000	**0.5661**	0.5643	0.3486	0.2144
50	**0.6442**	0.6351	0.5232	0.2961	**0.6123**	0.6076	0.3775	0.2368
100	**0.6717**	0.6701	0.5697	0.3281	**0.6754**	0.6743	0.4366	0.2919

set used to train the classifiers. Columns 2–5 show the performance in accuracy for the mRE, SVM, NB and K-NN classifiers respectively. Columns 6–9 show the performance in macro-*F*1 for the same classifiers. The results for the mRE, SVM and K-NN classifiers correspond to the ones of the optimal *r*,*C* and *K* parameters, respectively, obtained with 5-fold cross validation over the given training set.

The accuracies and macro-F1 measures for the mRE classifier are averages over the 10 runs performed in each experiment. The 95% confidence intervals with 9 degrees of freedom, estimated from a *t*-distribution, for all the accuracy and macro-F1 values for the mRE classifier are between ±8E-17 and ±1E-17. These very low intervals with high confidence show that the Lanczos algorithm is very stable and consistent when computing the approximations for the reconstruction matrices.

In the tables, the values in bold correspond to the best values for each measure regarding the corresponding portion of the training set used to build the specified classifier.

Classic dataset: in Table 4 we observe that the results for accuracy and macro-F1 for the mRE classifier with optimal rank *r* are stable and consistent with the different portions of the training set used to build the model. There is a tendency of increased performance when the size of the training set increases. The same tendency is observed for accuracy and macro-F1 for the SVM classifier. The NB classifier also performs quite stable and consistent in this dataset, however its results are generally lower than the ones of the SVM classifier. The K-NN classifier performs the worse in this case, showing not very consistent results, since it presents a better performance with the smallest fraction of the training set, while for other portions the performance is rather erratic, without a clear

tendency. The fact that K-NN perform the best when using only 1% of the training set could mean that this portion of the training set contains highly discriminative terms. The relatively low skewness and entropy of the category distribution in this dataset are indicators that there is a soft distribution of documents along the categories and the uncertainty of such distribution is low, meaning that it is a well-formed dataset. In such a case the SVM or NB classifiers are perhaps preferred over the mRE classifier, even if its performs competitively.

20Newsgroups dataset: in Table 5 we can observe that the mRE classifier, similarly than with the Classic dataset, shows a stable and consistent behavior, with a tendency towards an increased performance when the size of the training data increases. The same tendency is observed for both the SVM and the NB classifiers. In this case, the mRE classifier performs better than the other classifiers when using 30% or more of the data for training. The lower performance of the mRE classifier with less than 30% of the training data could be due that the model does not have enough data to learn the proper rank for the $\mathbf{W}_i$ matrices. The K-NN classifier applied on this dataset, as on the Classic dataset, has the worst results, showing inconsistent results and an erratic behavior, without a clear tendency in its performance. For this dataset the skewness is low, but the entropy is high, meaning the distribution of documents along the categories is soft but the uncertainty of such distribution is high. This dataset is more representative of a real world scenario, and from the results observed in the table above, we could infer that for cases with high entropy the mRE classifier presents a better performance, once we have enough training data to learn the proper rank for the matrices $\mathbf{W}_i$.

WIPO-alpha and WIPO-de datasets: these two datasets present similar properties: they have similar skewness and entropy, the same number of categories and the same order of magnitude in the number of training and test documents. We can observe from Tables 6 and 7 that the mRE classifier performs better than the other methods, in terms of both accuracy and macro-F1, for all the portions of the training set used to build the model. On both datasets the mRE classifier exhibits a consistent and stable performance, with a clear tendency to increase its performance when the training size increases. The same increasing tendency is observed for both the SVM and the NB classifiers, however, in both cases their performances are generally lower than the ones of the mRE classifier. For these datasets, contrary to the Classic and 20Newsgroups datasets, the K-NN classifier exhibits a stable and consistent behavior, showing a similar increasing performance as the ones of the other classifiers. Nevertheless, its performance is still low in comparison with the rest of the classifiers. For these datasets the skewness and the entropy are high, which is a better indication of a real world dataset; then, we can infer that for such cases the mRE classifier presents a better performance, disregarding the size of the training set used.

From all the tables above we can observe that the generally good performance of the mRE classifier is maintained disregarding the size of the training set used. This shows the ability of the method to generalize and properly model the categories inside each dataset. Additionally, it is observed that for datasets with high entropy and specially for datasets with high skewness, the mRE classifier performs generally better than the other methods. The SVM and NB classifiers exhibit also a generally good and stable performance along the datasets and portions of the training set used to build the model; nevertheless their performances are as well generally lower than the ones of the mRE classifier. The general low behavior of the K-NN classifier in all the datasets is perhaps due to the high number of terms, since it is known that K-NN reaches a better performance after a pre-selection of only discriminative features using a giving statistic (Rogati & Yang, 2002). It is also known that generally a linear SVM surpasses the K-NN classifier for text categorization tasks (Joachims, 1998; Rogati & Yang, 2002).

5. Conclusion

In this paper we have presented and evaluated a novel technique to classify documents, using their text content features, in a single-label multi-class scenario, where the task is to assign an unseen document into one of a set of predefined categories. The proposed model relies on the property of PCA to minimize the reconstruction error of the documents used to computed a low-rank transformation matrix $\mathbf{W}$. We called the proposed method Minimizer of the Reconstruction Error (mRE). The mRE classifier extends the property of error minimization of PCA and applies it to new unseen documents. We have shown that this technique is able to well preserve the diversity of the data from the different categories inside the transformation matrices $\mathbf{W}_i$. Then when such matrices are used to classify a new unseen document, the category matrix with similar properties to the new document, enables to reconstruct such document with minor loss of information. The reconstructed document based on a given category matrix that is closest to the original document indicates the category of the new document.

Results have shown that the mRE classifier performs well in terms of accuracy (or micro-F1) and macro-F1. During the training phase, mRE takes advantage from the sparsity of term-document matrices and the good performance of the Lanczos algorithm, in order to fast approximate the low rank category matrices $\mathbf{W}_i$. In this sense, the selection of the rank parameter r is an important issue for the mRE classifier, since depending on the dataset a different number of PCs are required to properly model the data in terms of minimization of the reconstruction error. Such parameter r is learned from the training documents by using a standard k-fold cross validation. Additionally, from the experiments we observed that for datasets with high entropy and specially for datasets with high skewness the mRE classifier performs generally better than the rest of the classifiers.

In the future we want to apply the mRE method to other text classification tasks. In particular we are interested in using it for hierarchical classification in large datasets, where there exist thousands of documents and thousands of categories and the categories are arranged in a taxonomy. The mRE could be used inside the taxonomy to model the internal categories. It would be interesting as well, to combine the output prediction of the mRE with other classifiers to refine the final prediction.

Acknowledgments

This research was supported partially by the CONACYT postdoctoral grant I0010-2010-01/150813 and by the KU Leuven project RADICAL (GOA 12/003).

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (third ed.). New Jersey: Wiley.

Barman, P., Igbal, N., & Lee, S. Y. (2006). Non-negative matrix factorization based text mining: Feature extraction and classification. In *Proceedings of the 13th international conference neural information processing ICONIP 2006* (pp. 703–712). Springer-Verlag.

Berry, M. W., Gillis, N., & Glineaur, F. (2009). Document classification using nonnegative matrix factorization and underapproximation. In *Proceedings of the IEEE international symposium on circuits and systems ISCAS 2009* (pp. 2782–2785). IEEE.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Springer.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Diamantaras, K. I., & Kung, S. Y. (1996). *Principal component neural networks: Theory and applications*. New York: Wiley.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (second ed.). Wiley-Interscience.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 179–188.

Gomez, J. C., Boiy, E., & Moens, M.-F. (2012). Highly discriminative statistical features for email classification. *Knowledge and Information Systems, 31*(1), 23–53.

Gomez, J. C., & Moens, M.-F. (2012). PCA document reconstruction for email classification. *Computational Statistics & Data Analysis, 56*(3), 741–751.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter, 11*(1), 10–18.

Hoffmann, H. (2007). Kernel PCA for novelty detection. *Pattern Recognition, 40*(3), 863–874.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, 24*(7), 498–520.

Ilin, A., & Raiko, T. (2010). Practical approaches to principal component analysis in the presence of missing values. *Journal of Machine Learning Research, 99*, 1957–2000.

Joachims, T. (1998). Text categorization with suport vector machines: Le arning with many relevant features. In *Proceedings of the 10th european conference on machine learning ECML 1998* (pp. 137–142). Springer-Verlag.

Jolliffe, I. T. (2002). *Principal component analysis* (second ed.). Springer.

Kim, H., Howland, P., & Park, H. (2005). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research, 6*, 37–53.

Lanczos, C. (1950). An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards, 45*(4), 255–282.

Li, Y. H., & Jain, A. K. (1998). Classification of text documents. *The Computer Journal, 41*(8), 537–546.

Malagón-Borja, L., & Fuentes, O. (2009). Object detection using image reconstruction with PCA. *Image Vision Computing, 27*(1-2), 2–9.

Mccallum, A. K. (1996). BOW: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/~mccallum/bow/.

Miranda, A. A., Borgne, Y.-A., & Bontempi, G. (2008). New routes from minimal approximation error to principal components. *Neural Processing Letters, 27*(3), 197–207.

Moler, C. B., & Stewart, G. W. (1973). An algorithm for generalized matrix eigenvalue problems. *Journal of Numerical Analysis, 10*(2), 241–256.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine, 2*(6), 559–572.

Rogati, M., & Yang, Y. (2002). High-performing feature selection for text classification. In *Proceedings of the 11th international conference on information and knowledge management CIKM 2002* (pp. 659–661). ACM.

Schütze, H., Hull, D. A., & Pedersen, J. O. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th annual international ACM SIGIR 1995* (pp. 229–237). ACM.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*(3), 379–423.

Shlens, J. (2009). A tutorial on principal component analysis. Systems Neurobiology Laboratory, University of California at San Diego.

Torkkola, K. (2001). Linear discriminant analysis in document classification. In *Proceedings of the IEEE international conference on data mining ICDM 2001 workshop on text mining*. IEEE.

Tsoumakas, G., Katakis, I., Vlahavas, I. (2010). Mining multi-label data. In *Data mining and knowledge discovery handbook* (pp. 667–685).

Weigend, A. S., Wiener, E. D., & Pedersen, J. O. (1999). Exploiting hierarchy in text categorization. *Information Retrieval, 1*(3), 193–216.