Computational Statistics and Data Analysis 56 (2012) 741–751

Contents lists available at SciVerse ScienceDirect

Computational Statistics and Data Analysis

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

PCA document reconstruction for email classification

Juan Carlos Gomez*, Marie-Francine Moens

KULEUVEN, Computer Science Department, Celestijnenlaan 200A, B-3001 Heverlee, Belgium

ARTICLE INFO

Article history:
Received 9 February 2011
Received in revised form 18 July 2011
Accepted 20 September 2011
Available online 1 October 2011

Keywords:
Class representation
PCA
Email filtering
Feature extraction

ABSTRACT

This paper presents a document classifier based on text content features and its application to email classification. We test the validity of a classifier which uses Principal Component Analysis Document Reconstruction (PCADR), where the idea is that principal component analysis (PCA) can compress optimally only the kind of documents – in our experiments email classes – that are used to compute the principal components (PCs), and that for other kinds of documents the compression will not perform well using only a few components. Thus, the classifier computes separately the PCA for each document class, and when a new instance arrives to be classified, this new example is projected in each set of computed PCs corresponding to each class, and then is reconstructed using the same PCs. The reconstruction error is computed and the classifier assigns the instance to the class with the smallest error or divergence from the class representation. We test this approach in email filtering by distinguishing between two message classes (e.g. spam from ham, or phishing from ham). The experiments show that PCADR is able to obtain very good results with the different validation datasets employed, reaching a better performance than the popular Support Vector Machine classifier.

1. Introduction

In automatic document classification, the aim consists of automatically assigning a new unseen document to one or more predefined classes, based only on certain features of the new instance. Document classification can be used for document filtering and routing to topic-specific processing mechanisms such as information extraction and machine translation. However, it is equally useful for filtering and routing documents directly to humans. Applications are e.g. filtering of news articles for knowledge workers, routing of customer documents in a customer service department, identification of criminal activities and filtering of undesired emails.

In the present work, we focus on document classification applied on filtering emails, in order to present to the user only the desired messages, since it is an important task given that email is one of the most popular ways of communication between people in all social, politic or economic organizations; with people using this electronic medium to share information, ideas and knowledge.

The popularity of emails is greatly due to the economy and rapidity of sending emails, making it possible to share any kind of information with several distant people at the same time. Nevertheless, these attributes that make emails popular, constitute also their weak points, since, given the facility to send them, the number of unsolicited messages with commercial and financial purposes is enormous. The most common type is spam, which include mainly publicity, but there are more dangerous types like phishing, which tries to steal financial identities. The problem of undesired emails is a serious growing issue (Fawcett, 2003), which not only consumes users' time and energy to identify and remove these messages, but also causes many problems such as taking up the limited mailbox space, wasting network bandwidth, loosing important personal

* Corresponding author. Tel.: +32 16 32 76 55.
E-mail addresses: juancarlos.gomez@cs.kuleuven.be, jcgcarranza@gmail.com (J.C. Gomez), sien.moens@cs.kuleuven.be (M.-F. Moens).

0167-9473/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2011.09.023

emails and even leads to direct financial losses. Given that, automatic classification of messages is not only desired but required to deal with this problem.

Within the several existing techniques to deal with email filtering (Guzella and Caminhas, 2009), a very promising approach is the use of content-based classifiers (Yu and Xu, 2008), using the text content of the messages rather than black lists, header or sender information. Some seminal papers in this sense include the use of bag-of-words representations and Bayesian classifiers like (Androutsopoulos et al., 2000; Brutlag and Meek, 2000; Robinson, 2003; Carreras et al., 2001). There are also interesting works on phishing detection like (Fette et al., 2007; Abu-Nimeh et al., 2007), describing a set of important features to distinguish phishing emails. More recent works include the use of *n*-grams as features (Kanaris et al., 2007) and Support Vector Machines (Sculley and Wachman, 2007) and compression models as classifiers (Bratko et al., 2006). Although state-of-the-art email filtering methods perform with high true positive and low false positive rates, there is a constant search into novel ways of solving the problem, since spammers and phishers are always evolving their techniques.

In this work we apply and test the validity of a novel approach for email filtering based on document reconstruction with Principal Component Analysis (PCADR). The technique of PCA reconstruction has been successfully used in computer vision for pedestrian detection (Malagón-Borja and Fuentes, 2009) and novelty detection (Hoffmann, 2007), but as far as we know it has not been used for text document classification or email filtering. The idea of this approach is that PCA can only perform a good reconstruction of the data that was used to compute the PCA basis, and that for other kind of data the reconstruction is poor. Thus, the classifier performs separately the PCA for each message class, obtaining a set of *principal components* (PCs) for each class. When a new instance arrives to be classified, this new example is projected in each set of computed PCs (reducing the dimensionality of the example to obtain a *reduced example*) and then is reconstructed using the same PCs. The reconstruction is understood as a projection of the reduced example into the original space, obtaining in this way a *reconstructed example* which has a dimensionality equal to the original example. Finally, the reconstruction error (the difference between the original example and the reconstructed example) is computed and the classifier assigns the example to the class with the smallest error.

One of the main drawbacks of PCA when applied to large datasets is the expensive time it requires to perform an eigenvalue decomposition ($O(n^3)$ using traditional methods) to find the PCs. In this work we deal with this problem by using the Power Factorization Method (PFM), a technique that is a generalization of the classic Power Method (Hotelling, 1933) which has been successfully used in some image analysis applications like multiframe motion segmentation (Vidal et al., 2008). PFM is simple and fast to approximate a fixed number of eigenvectors from a dataset.

We test the PCADR approach with several public email corpora: PU1, Ling-Spam, Phishing, SpamAssassin and TREC-07 spam corpus. These corpora contain messages collected in different periods of time and under very different set-ups. In our experiments, we use 10-fold cross validation, anticipatory testing and cross-corpus testing. The first validation reveals the general behavior schema of the classifier. In the anticipatory testing, we order emails from one corpus by date, we train the classifier on older emails and test on more recent ones. This setting helps to understand if the variance of the data captured by the PCA is able to persist over time. The cross-corpus testing is designed to test the classifier in a more drastic scenario, changing not only the timeframe, but also in data structure. This is accomplished by training with messages from one corpus and testing with messages from a different one. The goal of this last experimental setting is to understand if PCADR is able to capture the essential core information from the classes it wants to describe.

In order to have a better overview of the performance of the PCADR classifier, we present a comparison for every experiment with the SMO classifier, a popular Support Vector Machine (SVM) with good behavior in text document classification.

The contributions of our work are the evidence that the technique of PCADR behaves very well in classifying email messages, its easy modeling and fast implementation which permit to extend this technique to other (text) classification tasks, and the evidence that PCADR is able to extract and synthesize the essential information for robustly representing a class. This method gives stable classification results when testing on different corpora, and is able to generalize the class patterns in extreme circumstances, when training under a given setup and testing with a complete different one. Our findings contribute to the development of more advanced email filters and open new opportunities for text classification in general.

The remainder of this paper is organized as follows. Section 2 gives an overview of the related research. Section 3 introduces the architecture of the classifier PCADR. Section 4 describes the corpora used in this work, the preprocessing, the training and testing of the models and discusses our experimental evaluation of PCADR for email classification. Section 5 concludes this work with lessons learnt and future research directions.

2. Related research

Because in text classification, the documents are often represented by a large vocabulary of individual terms, dimensionality reduction has been popular for this task since the 90s. One of the most common techniques is Latent Semantic Analysis (LSA) (Deerwester et al., 1990). LSA is an application of principal component analysis (PCA) where a document is represented along its semantic axes or topics. The dimensions in LSA are computed by singular value decomposition of the term correlation matrix obtained from a large document collection. In a text categorization task, documents are represented by a LSA vector model both when training and testing the categorization system (e.g. Ishii et al., 2006; Pu and Yang, 2006). Other models such as probabilistic Latent Semantic Analysis (pLSA) (Hofmann, 1999) and Latent Dirichlet Allocation (Blei et al., 2003) are currently popular as topic representation models; where documents are represented 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).

Linear Discriminant Analysis (LDA) is a classification/dimensionality reduction technique (Fisher, 1936), which uses the class 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 an adequate class separability. LDA performs a projection of the complete training set in the new space (one projection per class). To classify a new unseen example it is projected into the new space and then its projection is compared with the mean of each projected training class. LDA also can be used as a dimensionality reduction technique similar to PCA, but using class information to improve the separation between classes in the new space (Anderson, 2003). In Torkkola (2001), 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 classify text documents. Nevertheless LDA as classifier tends to perform worse than a Support Vector Machine (SVM) for text classification (Kim et al., 2005).

Non-Negative Matrix Factorization (NMF) is another dimensionality reduction technique, similar to PCA. It projects the data to a new space, but coefficients obtained with NMF are only positive. This method has recently become popular for text classification (Barman et al., 2006; Berry et al., 2009; Silva and Ribeiro, 2009).

In the field of email filtering, several different methods have been proposed (e.g. Gansterer et al., 2005; Goodman et al., 2005; Cormack, 2007; Guzella and Caminhas, 2009). We cite here some seminal papers for spam classification using traditional Bayesian filters like (Androutsopoulos et al., 2000; Robinson, 2003; Carreras et al., 2001). There are also interesting works on phishing detection like (Fette et al., 2007; Abu-Nimeh et al., 2007; Gansterer and Pölz, 2009), where the authors describe discriminative features to distinguish phishing emails. (Brutlag and Meek, 2000) investigate the effect of feature selection by means of the mutual information statistic on email filtering. (Xia and Wong, 2006) discuss email categorization in the context of personal information management. In recent work like (Bratko et al., 2006; Bíró et al., 2008; Kanaris et al., 2007) the authors use respectively compression models, Latent Dirichlet Allocation models and *n*-grams to produce more robust features for email filtering. In Gomez and Moens (2010) and Gomez and Moens (2011), Biased Discriminant Analysis (BDA) and Average Neighbor Margin Maximization (ANMM), two extended versions of LDA, are used for email filtering. In Janecek and Gansterer (2010), the authors use several methods based on NMF for email filtering.

In line with the LSA approach, which is in essence a PCA applied to a term–document matrix, most of the works devoted to text classification and email filtering where PCA is used, employ PCA as a first step to reduce the dimension of the term space, after which the classification is performed using standard classification algorithms (e.g., SVM, Naive Bayes, *k*-nearest neighbor) (Gee, 2003; Gansterer et al., 2007). In this work, we focus on the discriminative properties of PCA to build a classifier for filtering email messages in a framework where PCA is used for document reconstruction. To the best of our knowledge there is no work devoted to PCA document reconstruction in email filtering or text classification in general.

3. Classifier architecture

3.1. Principal component analysis for document reconstruction

Principal Components Analysis (PCA) was first developed by Pearson (1901) and its statistical properties were investigated in detail by Hotelling in his seminal paper (Hotelling, 1933). Anderson (2003) has given one the most comprehensive exposition of this technique. In general, PCA is a popular method that reduces data dimensionality by performing a covariance analysis between factors (Jolliffe, 1986). Such technique has been successfully used as an initial step in many applications in computer vision, data compression, pattern recognition, etc. The formulation of standard linear PCA, using emails as the data to compress, is as follows. Let $\{(\mathbf{x}_1, c_1), (\mathbf{x}_2, c_2), \dots, (\mathbf{x}_n, c_n)\}$, with $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, be a set of labeled email messages with their corresponding classes, where $\mathbf{x}_i \in \mathbb{R}^d$ is the *i*-th email, represented as a *d* dimensional *column* vector (where *d* is the number of features used to represent a message), and $c_i \in \mathbf{C}$ is the label of $\mathbf{x}_i$. The "mean message vector" of the set is defined as:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i. \tag{1}$$

Then, considering $\mathbf{M} = \mathbf{X} - \boldsymbol{\mu}$ as the original data centered by subtracting the mean vector, the covariance matrix **Co** is given by:

$$\mathbf{Co} = \frac{1}{n}(\mathbf{M})(\mathbf{M})^T. \tag{2}$$

The next step consists of computing the eigenvalues and their corresponding eigenvectors from the covariance matrix; these vectors form a PCA basis or the *principal components* (PCs) of **Co**. These eigenvectors can be computed in several ways. In our system we do not compute the eigenvectors directly from **Co**, rather we use a relation of the covariance matrix with the Singular Value Decomposition (SVD), since SVD is less restrictive and can be performed on any $d \times n$ matrix.

The SVD is a technique for decomposing a matrix into a set of rotation and scale matrices. Using the same **M** as above, we have the decomposition as:

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{d\times n}$ is the original centered data, $\mathbf{U} \in \mathbb{R}^{d\times d}$, $\mathbf{S}$ is a diagonal matrix of size $\mathbb{R}^{d\times n}$, and $\mathbf{V} \in \mathbb{R}^{n\times n}$; with both $\mathbf{U}$ and $\mathbf{V}$ being orthogonal. The following equation show the relation between SVD and the covariance matrix using the matrix $\mathbf{M}$ as the link:

$$\mathbf{MM}^T = (\mathbf{USV}^T)\,(\mathbf{USV}^T)^T = \mathbf{USV}^T\mathbf{VSU}^T = \mathbf{US}^2\mathbf{U}^T. \tag{4}$$

We use that relation to compute the PCA basis (i.e. the PCs) using the Power Factorization Method (PFM), a technique that is a generalization of the Power Method and which reduces the required processing time by computing only a given (fixed) number of PCs instead of performing the complete decomposition. This method is explained with more detail in Section 3.2.

The computed PCs are then sorted in decreasing order using the eigenvalues as reference, in this manner a projection onto the space defined by the first l PCs ($1 \ll l \ll d$) would be optimal with respect to the information loss.

Now, let $\mathbf{W}$ be the matrix whose columns are the first l PCs extracted from the matrix $\mathbf{X}$ using the process described above, with $\mathbf{W} \in \mathbb{R}^{d\times l}$. The projection of an email message $\mathbf{x}$ (represented as a column vector) into the eigenspace is:

$$\mathbf{p} = \mathbf{W}^T(\mathbf{x} - \boldsymbol{\mu}). \tag{5}$$

The concept of document/message reconstruction with PCA can be understood as follows: the message vector is first projected into the PCs, and from this projection, the idea is to try to recover the original message vector using the same PCs. Thus the reconstructed message vector $\mathbf{x}'$ is:

$$\mathbf{x}' = \mathbf{Wp} + \boldsymbol{\mu} = \mathbf{WW}^T(\mathbf{x} - \boldsymbol{\mu}) + \boldsymbol{\mu}. \tag{6}$$

Finally, the reconstruction error is defined as the difference between the original message vector and the reconstructed message vector. Using the Euclidean distance we have:

$$r = |\mathbf{x} - \mathbf{x}'| = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}. \tag{7}$$

In general, when more PCs are used to obtain the projection, the information loss will be less; thus we will have a more accurate reconstruction of the message vector. Additionally, the more similar the message vector $\mathbf{x}$ is to the messages used to generate the matrix $\mathbf{W}$, the better the reconstruction will be for a fixed number of PCs.

3.2. *Power factorization method*

Performing PCA using traditional methods like the QZ algorithm (Moler and Stewart, 1973), is expensive for large matrices because the internal decomposition takes $O(n^3)$, then the use of alternative methods is useful. The Power Factorization Method (PFM) is a generalization of the classic Power Method exposed by Hotelling (1933), and is a fast technique for approximating low rank matrices. The PFM technique is discussed in detail in Hartley and Schaffalitzky (2004) and Morita and Kanade (1997). In order to carry out PCA using PFM in our system, we have the centered message matrix $\mathbf{M} \in \mathbb{R}^{d\times n}$ defined above. Then, the idea is to represent the column space of $\mathbf{M}$ by a small number l of vectors. Also, we want to find the matrix $\widehat{\mathbf{M}}$ of rank l that is closest to $\mathbf{M}$ using the Frobenius norm. A common way to do it is by using SVD. Let $\mathbf{M} = \mathbf{USV}^T$, where the diagonal entries of $\mathbf{S}$ (the singular values) are in descending order. Then the first l columns of $\mathbf{U}$ form an orthonormal basis for the dimension-l subspace, i.e. the PCA basis or the PCs we need. Additionally, the closest rank-l matrix to $\mathbf{M}$ is the matrix $\widehat{\mathbf{M}} = \mathbf{US}^{(l)}\mathbf{V}^T$, where $\mathbf{S}^{(l)}$ represents the (diagonal) matrix formed from $\mathbf{S}$ by setting all but the first l diagonal entries to zero.

In the PFM algorithm, two matrices are estimated $\mathbf{W} \in \mathbb{R}^{d\times l}$ and $\mathbf{B} \in \mathbb{R}^{n\times l}$, such that $\widehat{\mathbf{X}} = \mathbf{WB}^T$ is the closest rank-l approximation to $\mathbf{M}$. Here, $\mathbf{W}$ has orthonormal columns, which therefore are the PCs we are looking for and $\mathbf{W}$ is the projection matrix we need. The matrix estimation is done by a simple iterative procedure, which starts from an initial value for the matrix $\mathbf{W}_0$, then:

1. $k \leftarrow 0$.
2. $\mathbf{B}_k = \mathbf{M}^T\mathbf{W}_k$.
3. $\mathbf{W}_{k+1} = \mathbf{MB}_k$.
4. Apply the Gram–Schmidt algorithm (QR-factorization) to orthonormalize the columns of $\mathbf{W}_{k+1}$.
5. $k \leftarrow k + 1$.

According to Hartley and Schaffalitzky (2004), the product $\mathbf{W}_k\mathbf{B}_k^T$ is guaranteed to converge linearly (even from a random starting point for $\mathbf{W}_0$) to the closest rank-l matrix $\widehat{\mathbf{M}}$ to $\mathbf{M}$. In the same work,the authors establish that a small number (in this work we use 6) of iterations of the procedure is sufficient to converge to the PCA basis.

Table 1
Number of messages per corpus.

Corpus	Spam	Phishing	Ham	Total
PU1	481		618	1099
PC		1250	1250	2500
LS	481		2412	2893
SA	4150		1897	6047
TREC	50,199		25,220	75,419

3.3. Classification using PCADR

PCA can be seen as a process to reveal the internal structure of the data that are being analyzed, by means of looking for the set of PCs that best describes the variance or the distribution of such data. Therefore, these PCs are going to preserve better the information of the messages on which the PCA was applied, or of those that are similar. Thus, if we have a set of PCs that were obtained from a set of, for example, spam messages only, these must better reconstruct other spam messages than another type of messages (for example ham), and vice versa, if we have a set of PCs obtained from ham messages, the reconstruction of the spam messages will not be as good.

Given such assumption, a classifier based on PCA document/message reconstruction (PCADR) can be created. This classifier will decide when an unseen incoming message belongs to one of two exclusive classes (e.g. spam from ham, or phishing from ham). The algorithm to perform this binary classification is the following:

Before the actual classification, we train the classifier:

1. Preprocess the labeled messages to obtain two message matrices $\mathbf{X}$ for class 1 and $\mathbf{Y}$ for class 2
2. Perform PCA for the message matrix $\mathbf{X}$ to obtain the projection matrix $\mathbf{W}_x$, composed by l PCs, and the mean vector $\boldsymbol{\mu}_x$
3. Perform PCA for the message matrix $\mathbf{Y}$ to obtain the projection matrix $\mathbf{W}_y$, composed by l PCs, and the mean vector $\boldsymbol{\mu}_y$

For each new message we assign the appropriate class:

1. Preprocess the message to obtain the message vector $\mathbf{z}$
2. Use the matrices $\mathbf{W}_x$ and $\mathbf{W}_y$ and the mean vectors $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_y$ to perform two reconstructions
   (a) $\mathbf{z}'_x = \mathbf{W}_x^T \mathbf{W}_x(\mathbf{z} - \mu_x) + \mu_x$
   (b) $\mathbf{z}'_y = \mathbf{W}_y^T \mathbf{W}_y(\mathbf{z} - \mu_y) + \mu_y$.
3. Obtain the reconstruction errors
   (a) $r_x = |\mathbf{z} - \mathbf{z}'_x|$
   (b) $r_y = |\mathbf{z} - \mathbf{z}'_y|$.
4. The total reconstruction error is then $r_t = ar_y - br_x$, where a and b are positive values used to weigh the errors, depending on the importance of each class.
5. Classify the message using the following criterion class $(\mathbf{z}) = \begin{cases} \text{class 1 if } r_t \geq 0 \\ \text{class 2 if } r_t < 0 \end{cases}$.

4. Experiments and results

4.1. Corpora

The public email corpora we use for performing our tests are: the PU1[1] corpus (Androutsopoulos et al., 2000); the Ling-Spam[2] (*LS*) corpus (Androutsopoulos et al., 2000); the SpamAssassin (*SA*)[3] corpus; the TREC 2007 Public Spam Corpus (*TREC*)[4] (Cormack, 2007); and finally the Phishing Corpus (*PC*), created by randomly selecting 1250 phishing messages from the Nazario's corpus[5] and 1250 ham messages from the TREC corpus. The number of emails in each corpus is listed in Table 1 in ascending order.

4.2. Preprocessing

In general an email consists of two parts: the header and the body message. The header contains information about the message in the form of many fields like sender, subject, receiver, servers, etc. The body contains the message itself and usually takes one of two forms: HTML or plain-text. The HTML emails contain a set of tags to format the text to be displayed on screen. Before applying PCADR, the corpora of emails are pre-processed by removing all the structured information,

[1] Available at: http://nlp.cs.aueb.gr/software.html.
[2] Available at: http://nlp.cs.aueb.gr/software.html.
[3] Available at: http://spamassassin.apache.org/publiccorpus/.
[4] Available at: http://plg.uwaterloo.ca/~gvcormac/treccorpus07/.
[5] Available at: http://monkey.org/~jose/wiki/doku.php?id=PhishingCorpus.

$$\mathbf{X} = \left[\begin{array}{ccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_n \\ tfidf_{1,1} & tfidf_{1,2} & tfidf_{1,3} & \cdots & tfidf_{1,n} \\ tfidf_{2,1} & tfidf_{2,2} & tfidf_{2,3} & \cdots & tfidf_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ tfidf_{d,1} & tfidf_{d,2} & tfidf_{d,3} & \cdots & tfidf_{d,n} \end{array}\right]$$

$$\mathbf{Y} = \left[\begin{array}{ccccc} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_m \\ tfidf_{1,1} & tfidf_{1,2} & tfidf_{1,3} & \cdots & tfidf_{1,m} \\ tfidf_{2,1} & tfidf_{2,2} & tfidf_{2,3} & \cdots & tfidf_{2,m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ tfidf_{d,1} & tfidf_{d,2} & tfidf_{d,3} & \cdots & tfidf_{d,m} \end{array}\right]$$

Fig. 1. Structure of the data matrices.

i.e. the header and the HTML tags. In this way, only the text content from the document is extracted. The next step consists of building the vocabulary of the email messages. We choose to remove words that are evenly distributed over the classes by means of a mutual information statistic, obtaining 1000 initial features for the PU1 and PC datasets and 5000 initial features for the rest of the corpora. These numbers were selected depending on the number of unique terms in the corpus, considering that PU1 and PC are the smallest corpora. Additionally, we weight the remaining words in each document by a TF–IDF schema. In this way the importance of each term increases proportionally to the number of times it appears in the document, but is offset by the frequency of the term in the whole corpus. We do this preprocessing for each corpus, and what we obtain at the end are the message matrices $\mathbf{X} \in \mathbb{R}^{d\times n}$ and $\mathbf{Y} \in \mathbb{R}^{d\times m}$, where each column is a message vector of size $d \in \{1000, 5000\}$, and n and m are the numbers of messages of each class in the corpus. These message matrices are the ones used to perform the PCADR. The structure of the matrices **X** and **Y** is shown is Fig. 1.

4.3. Training and testing the classification model

The model for training is constructed by applying PCA to the message matrices **X** and **Y** as explained in Sections 3.1 and 3.2, to obtain the projection matrices $\mathbf{W}_x$ and $\mathbf{W}_y$ as explained in Section 3.3, which are stored and used later for the testing phase. Experimentally we defined the value $l = 128$ to be the rank of the **W** matrices, i.e. PCADR is performed using 128 features. We chose this value since the rank of the **W** matrices should be significantly reduced (in this case about or less than one order of magnitude) than the one of the original data matrices **X** and **Y**. The proposed rank reduction makes it worth applying the PFM to fast extract the PCs, while preserving the variance inside the original data (Gomez and Moens, 2010, 2011).

In the case of the experiments where we classify spam versus ham, spam is considered as the positive class, or matrix **X**. In the case of phishing versus ham, phishing is the positive class, or matrix **X**. Consequently, for the values of a and b, which indicate the weight of each error in the testing phase as explained in Section 3.3, we defined experimentally values of $a = 1$ and $b = 1.03$, giving more importance to the ham class. This is done because in general, the idea is to preserve the ham messages, i.e., to reduce the false positives. Changing these weights leads to building a classifier with a different behavior.

In order to assess the time complexity of the PCA based on PFM and the accuracy of the classifier based on the PCs of a class, we also perform experiments with the traditional singular value decomposition, using the implementation provided by the Jama package, which is based on the QZ algorithm. Given that the PCs extracted with the PFM could be different than the ones found by the traditional method, slight differences in the reconstruction are expected and consequently small changes in the measures of performance are also expected, but, these differences are not statistically significant ($\rho > 0.05$, using a Wilcoxon signed rank test testing the hypothesis that $F(x)\langle\rangle G(y)$ i.e. the values of one method tend to be different than the values of the other).

Additionally, in order to compare the performance of the PCADR classifier, we also use the popular SMO classifier (Platt, 1998), a linear SVM which is known by its very good performance in sparse data and which is especially well suited for text classification. We used the SMO implementation from the Weka package, using the following settings: a lineal kernel (polynomial with exponent 1); complexity constant equal to 100, which makes the SVM works better (Sculley and Wachman, 2007); gamma of 0.001; no normalization of the variables, which makes the SVM to work faster with sparse instances (Witten and Frank, 2000); and building of logistic models, in order to obtain better constructions of the ROC (see below). The SVM is trained using the whole set of terms (without any feature selection or extraction), and using binary representation (1 if the feature is present in the message and 0 if not) since it is known that a binary representation performs better than the TF–IDF representation with a SVM (Drucker et al., 1999). The total number of terms varies depending of the corpus. We used several reported parameters from the literature trying to obtain the best (and fast) performance for SVM in text classification.

For testing we performed several types of experiments: 10-fold cross validation, anticipatory testing and cross-corpus testing. It is known that 10-fold cross validation is not the best way to test an email filter (Bratko et al., 2006), but we performed these experiments to test how well the PCADR performs under a fixed scenario. We applied 10-fold cross validation for: PU1, PC and LS corpora. For the PU1 and LS corpora, they were already divided by the creators (Androutsopoulos et al., 2000) into 10 parts of equal size, with equal proportion of ham and spam messages across the

Table 2
Description of the cross-corpus experiments.

Experiment	Training corpus	Testing corpus
1	Subset of 4500 messages (2250 spam and 2250 ham) from TREC corpus	Whole LS corpus
2	Whole LS corpus	Subset of 4500 messages (2250 spam and 2250 ham) from TREC corpus
3	Subset of 9000 messages (4500 spam and 4500 ham) from TREC corpus	Whole SA corpus
4	Whole SA corpus	Subset of 9000 messages (4500 spam and 4500 ham) from TREC corpus
5	Whole LS corpus	Whole SA corpus
6	Whole SA corpus	Whole LS corpus

Table 3
Performance of the methods for the PU1 corpus using 10-fold cross validation (the quantities are averages over the 10-folds).

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.96966	0.97358	0.98916	39.29	3.57
PCADR (SVD)	128	0.96748	0.97176	0.98946	58.23	4.06
SVM	23,312	0.96150	0.96542	0.99176	25.29	16.93

10 parts. The PC corpus was randomly split into 10 parts, keeping the same proportion of phishing and ham consistent across the 10 parts.

We also performed one experiment in order to test the anticipatory proprieties of the classifiers by training with data in the past and test with data in the future. This experiment was done for the TREC corpus, where the emails are sorted by date, using a one-off schema, by taking a small part of the examples with an early date for training, and the later data for testing. We took 9020 messages, corresponding to the first week, for training and the remaining 66,399 messages, corresponding to (almost) 11 weeks in the future for testing.

Finally, the cross-corpus experiments were done by training the models with messages from one corpus and tested with messages from a complete different corpus. We performed six cross-corpus experiments which are specified in Table 2. The cross-corpus experiments were designed thinking about extreme cases where an email filter is trained with email messages that were selected under one given setup (users, inboxes, dates, subjects, etc.) and is tested under a complete different one. In this way we can see how well the features and the classifiers are able to generalize on the email classes.

As is common in the binary email filtering problem, we present results taking spam (or phishing) messages as the positive class, since these are the messages filtered by the models. Results are expressed for each experiment using the spam (phishing) F1 measure (computed only for the positive spam or phishing class), which summarizes the spam (phishing) precision and spam (phishing) recall; accuracy, which gives a general overview of the classification with the defined parameters; and the area under the ROC (Receiver Operating Characteristic) curve (here called ROCA). The ROCA metric aims at a high true positive rate and a Low false positive rate and it is an important measure for commercial settings in email filtering, where the cost for misclassifying a legitimate email as spam or phishing is high. In the case of the SVM classifier, the classifier first fits a logistic model to its output in order to produce a ROC for the classification. In the case of the PCADR classifier, the ROC is constructed by varying the classification threshold according to the total reconstruction error in the training data, where $r_t \geq$ threshold indicates a positive classification (usually threshold $= 0$), obtaining in this way the set of points to plot the ROC. Finally, in both cases, using a library from Weka, based on the Mann Whitney statistic (Mann and Whitney, 1947), we obtain the ROCA measure.

All the experiments were performed using a Core i7 1.7 Ghz PC with 4 GB in RAM using Windows and Java.

4.4. 10-fold cross validation

Tables 3–5 show the results for the PCADR and SVM classifiers for the three corpora: PU1, PC, and LS, performing a normal classification using 10-fold cross validation. The first column of each table indicates the method: PCADR (PFM), where the PCs used in the reconstruction were estimated using the PFM method; PCADR (SVD), where the PCs were found using the traditional singular value decomposition; and SVM, the SMO used with the whole set of terms. The second column shows the number of features used by each classifier. The third, fourth and fifth columns respectively show the performance in terms of: F1, accuracy and ROCA, respectively. The two last columns show the training and testing time respectively. The training time for PCADR includes the preprocessing of the messages (text extraction and vectorization) and the calculations to approximate the matrices $\mathbf{W}_x$ and $\mathbf{W}_y$. Training time for SVM includes the preprocessing of messages (text extraction and vectorization) and the training of the classifier. Testing time for PCADR includes the preprocessing of messages, the projection with the $\mathbf{W}$ matrices and the classification with the PCADR model. Testing time for the SVM includes the preprocessing of the messages and the classification with its model. Testing time is given for classifying the whole test set. Both training and testing times are expressed in seconds.

From these 10-fold cross validation experiments we can observe that the SVM classifier needs a large number of features in order to be able to map from features to classes; on the other hand PCADR performs quite well and is competitive with

Table 4
Performance of the methods for the PC corpus using 10-fold cross validation (the quantities are averages over the 10-folds).

Method	Number of features	Phishing F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.98334	0.98346	0.99694	115.95	7.31
PCADR (SVD)	128	0.98334	0.98346	0.99704	242.73	7.69
SVM	39,506	0.98271	0.98263	0.99803	106.90	20.04

Table 5
Performance of the methods for the LS corpus using 10-fold cross validation (the quantities are averages over the 10-folds).

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.97553	0.99170	0.99861	162.25	21.76
PCADR (SVD)	128	0.97553	0.99170	0.99788	3199.63	23.50
SVM	55,833	0.96286	0.98791	0.99903	60.00	39.83

Table 6
Performance of the methods for the TREC corpus using one-off experiment. Here we sort the corpus by date, and then we take from the beginning of the corpus 9020 emails that have an early date for training and the remaining 66,399 emails for testing.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.97275	0.96439	0.98820	597.05	4033.30
PCADR (SVD)	128	0.97262	0.96425	0.98788	22595.25	4023.55
SVM	47,534	0.98041	0.97490	0.99510	457.50	5081.72

a small set of features, meaning that PCA is capturing most of the data variance from the training set in the projection matrices $\mathbf{W}_x$ and $\mathbf{W}_y$, and these matrices are able to reconstruct well the unlabeled messages of the same class. The advantage of using few features is reflected in the testing time, where SVM requires more time to perform the classification, given the large number of features it uses. In these experiments the PCADR classifier is able to outperform the SVM classifier in terms of F1 and accuracy, but not in terms of the ROCA measure. Nevertheless, running Wilcoxon signed rank tests (testing the hypothesis that $F(x)\langle\rangle G(y)$, i.e. the values of one method tend to be different than the values of the other) over the classification results for each corpus we found that the differences between PCADR and SVM are not statistically significant ($\rho > 0.05$) in any of the three corpora with any of the measures of performance. This means both classifiers perform similar in a fixed email classification scenario.

Additionally, in the case of the PU1 and PC corpora, where 1000 words where selected to form the vocabulary, there is not a big difference between the two methods PFM and traditional SVD to compute the PCs used for classification in PCADR. Nevertheless, when we used more words to form the vocabulary, like with the LS corpus, the benefit of PFM over SVD can be clearly seen, since training time with the PFM is only a fraction of the time with the SVD traditional method.

4.5. *Anticipatory experiment*

Table 6 shows the results of the classification for the TREC-07 spam corpus using the anticipatory testing as described above. In this case, the SVM surpasses the PCADR classifier in terms of all the measures, but in general, the results from PCADR are still competitive and result in a better testing time. This behavior was expected since the training set is composed by a larger number of messages and a bigger variation of topics than in the 10-fold cross validation experiments (several thousands of messages). The testing set has a similar diversity and composition than the training one, which requires a big diversity in the (or a big number of) variables used to map between examples and classes. Then, as the SVM captures a much bigger number of features from the testing set than the PCADR, the SVM is able to perform a better classification. This result implies that a bigger vocabulary and/or a bigger number of selected PCs to perform the classification with the PCADR classifier would be necessary to capture the variance in the data so to better represent the classes. Similar to the experiment with the LS corpus, here the training time with the PFM is only a fraction of the training time with the traditional SVD method.

4.6. *Cross-corpus experiments*

The cross-corpus experiments are more interesting for the task of email classification, since in the previous 10-fold cross validation and anticipatory experiments the structure of the training and testing sets are very similar. Both sets are obtained from the same corpus under similar conditions, i.e. collected from the same server(s), the same set of users, the same period of time, labeled using the same rules, etc. In the cross-corpus experiments, different set-ups for training and testing are used, with the purpose of simulating a more realistic scenario, where the settings for training and testing of a filter could be different as is often the case with commercial products.

Tables 7–12 present the result for the six cross-corpus experiments explained above. Except for the results in Table 11 (experiment 5), where we train with the LS corpus and test with the SA corpus, in the rest of the results it is the PCADR

Table 7
Performance of the methods for cross-corpus experiment 1. Here we train with examples from the TREC corpus and then we test with examples from the LS corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.72253	0.91531	0.90112	341.97	257.39
PCADR (SVD)	128	0.71412	0.91393	0.89914	6379.02	238.73
SVM	52,883	0.31714	0.30487	0.73783	317.90	353.15

Table 8
Performance of the methods for cross-corpus experiment 2. Here we train with examples from the LS corpus and then we test with examples from the TREC corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.65582	0.67993	0.64093	200.81	329.90
PCADR (SVD)	128	0.65642	0.68149	0.64113	6689.64	321.86
SVM	59,661	0.45408	0.57161	0.59817	165.20	360.86

Table 9
Performance of the methods for cross-corpus experiment 3. Here we train with examples from the TREC corpus and then we test with examples from the SA corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.66807	0.81773	0.86722	588.93	427.67
PCADR (SVD)	128	0.66546	0.81558	0.86740	18351.16	355.34
SVM	47,534	0.60054	0.70505	0.80823	477.64	558.38

Table 10
Performance of the methods for cross-corpus experiment 4. Here we train with examples from the SA corpus and then we test with examples from the TREC corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.72822	0.65142	0.84313	333.88	574.44
PCADR (SVD)	128	0.73050	0.65353	0.84183	16253.53	512.53
SVM	57,021	0.68146	0.58631	0.78746	171.98	712.85

Table 11
Performance of the methods for cross-corpus experiment 5. Here we train with examples from the LS corpus and then we test with examples from the SA corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.69658	0.74681	0.89275	173.49	405.71
PCADR (SVD)	128	0.69737	0.74747	0.89311	3330.07	414.24
SVM	59,661	0.76428	0.83247	0.89827	102.65	598.43

classifier that outperforms the SVM in terms of all measures (F1, accuracy and ROCA), and even for the training time, the PFM version of PCADR is very competitive. Running Wilcoxon signed rank tests using the collected data from the several cross-corpus experiments, we test the hypothesis of $F(x) > G(y)$, i.e. if the values of PCADR tend to be better than the ones of SVM, and we found this hypothesis being true (even with the results of experiment 5) and the difference is statistically significant with $\rho < 0.05$ for every performance measure.

In experiment 1, shown in Table 7, we train with examples from the TREC corpus and test with examples from the LS corpus. Here we can observe that PCADR is able to better represent the classes using the PCs of the training set; while, the SVM has a poor performance using all unique terms. This means that the word features used in TREC to train the classifier are not present in the LS corpus, which makes it hard for the SVM to classify the examples in the LS corpus. On the other hand, the variance of the frequency of the terms extracted with PCADR is a better representation of the classes.

For the rest of the experiments we observe a similar behavior, but the differences in performance between PCADR and SVM are smaller compared to the first experiment. This means that the word features used to train the classifier are more present in the testing set. Even so, PCADR is better to represent the classes and to better generalize across corpora and time frames using the reconstruction matrices.

5. Conclusions

In this paper we have presented and evaluated a novel technique based on PCA document reconstruction (PCADR) in the context of email filtering while using only text-content features. The approach is understood as a classifier based on the good discrimination properties of the variance between datasets obtained when performing PCA. We have shown that

Table 12
Performance of the methods for cross-corpus experiment 6. Here we train with examples from the SA corpus and then we test with examples from the LS corpus.

Method	Number of features	Spam F1	Accuracy	ROCA	Training time (s)	Testing time (s)
PCADR (PFM)	128	0.81883	0.93881	0.95381	317	236.8
PCADR (SVD)	128	0.81115	0.93674	0.95236	13943.27	233.35
SVM	57,021	0.40589	0.55375	0.86661	174.9	364.3

this technique is able to well preserve the diversity of the data using only a few PCs and that the PCs represent important class information in terms of variance for each dataset. This information is expressed as a projection matrix, and when used to reconstruct an unseen example to be classified, the class matrix with similar properties enables to reconstruct such example with minor loss of information. The reconstructed example based on a given class matrix that is closest to the original example indicates the class of the example.

Results show that PCADR performs well when separating spam from ham, and phishing from ham messages. PCADR is able to outperform a SVM when considering the accuracy of the classification, and in terms of F1 and ROCA for most of the experiments, with the advantage of PCADR being faster than the SVM when classifying test examples. When computing the PCs based on the Power Factorization Method (PFM), PCADR is competitive in training time in comparison to a SVM. PCADR is especially well suited for classification when training with a labeled dataset collected using a given setup and testing with a dataset collected with another setup. This type of approach could be useful for commercial products where filters are used to classify messages not similar in superficial structure to the ones used for training.

In the future we want to apply the PCADR method to other text classification tasks. PCADR can be easily extended to deal with a multiclass problem by simply constructing a **W** matrix per class, and then performing the reconstruction with each matrix. Also, it would be interesting to combine PCADR with another classifier to produce a weighted decision about the class. In particular for email filtering, the PCADR classifier can be complemented with the use of other non-text-content features, like header, link and embedded image information.

Acknowledgments

This work was partially supported by the CONACYT postdoctoral grant I0010-2010-01/150813 and by the EU project FP6-027600 Antiphish.

References

Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S., 2007. A comparison of machine learning techniques for phishing detection. In: Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit: eCrime 2007. ACM, New York, pp. 60–69.

Anderson, T.W., 2003. An Introduction to Multivariate Statistical Analysis, 3rd ed.. Wiley, New Jersey.

Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D., 2000. An evaluation of naïve Bayesian anti-spam filtering. In: Proceedings of the 11th European Conference on Machine Learning: ECML 2009, Workshop on Machine Learning in the New Information Age. Springer-Verlag, Berlin, pp. 9–17.

Barman, P., Iqbal, N., Lee, S.Y., 2006. Non-negative matrix factorization based text mining: feature extraction and classification. In: Proceedings of the 13th International Conference ICONIP 2006. Springer-Verlag, Berlin, pp. 703–712.

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 2009. IEEE, Taipei, pp. 2782–2785.

Bíró, I., Szabó, J., Benczúr, A.A., 2008. Latent Dirichlet allocation in web spam filtering. In: Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web: AIRWeb 2008. ACM, New York, pp. 29–32.

Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J., 2003. Latent Dirichlet allocation. Journal of Machine Learning Research 3, 993–1022.

Bratko, A., Cormack, G., Filipic, B., Lynam, T., Zupan, B., 2006. Spam filtering using statistical data compression models. Journal of Machine Learning Research 7, 2673–2698.

Brutlag, J.D., Meek, C., 2000. Challenges of the email domain for text classification. In: Proceedings of the 17th International Conference on Machine Learning: ICML 2000. Morgan Kaufmann, San Francisco.

Carreras, X., Márquez, L., Salgado, J.G., 2001. Boosting trees for anti-spam email filtering. In: Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing: RANLP 2001. John Benjamins, pp. 58–64.

Cormack, G.V., 2007. Spam track overview. In: Proceedings of the 16th Text REtrieval Conference: TREC-2007. National Institute of Standards and Technology (NIST).

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, 391–407.

Drucker, H., Wu, D., Vapnik, V.N., 1999. Support vector machines for spam categorization. IEEE Transactions on Neural Networks 10 (5), 1048–1054.

Fawcett, T., 2003. In vivo spam filtering: a challenge problem for data mining. SIGKDD Explorations 5 (2), 203–231.

Fette, I., Sadeh, N., Tomasic, A., 2007. Learning to detect phishing emails. In: Proceedings of the 16th International World Wide Web Conference: WWW 2007. ACM, New York, pp. 649–656.

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

Gansterer, W.N., Ilger, M., Lechner, P., Neumayer, R., Strauss, J., 2005. Anti-spam methods — state of the art. Tech. rep.

Gansterer, W.N., Janecek, A.G.K., Neumayer, R., 2007. Spam filtering based on latent semantic indexing. In: Survey of Text Mining II: Clustering, Classification, and Retrieval. Springer-Verlag, London, pp. 165–183.

Gansterer, W.N., Pölz, D., 2009. E-mail classification for phishing defense. In: Proceedings of the 31st European Conference on Information Retrieval: ECIR 2009. Springer-Verlag, Toulouse, pp. 449–460.

Gee, K.R., 2003. Using latent semantic indexing to filter spam. In: Proceedings of the 2003 ACM Symposium on Applied Computing, Data Minning Track. ACM, New York, pp. 460–464.

Gomez, J.C., Moens, M.-F., 2010. Using biased discriminant analysis for email filtering. In: Proceedings of the 14th International Conference KES 2010. Springer-Verlag, Berlin, pp. 566–575.

Gomez, J.C., Moens, M.-F., 2011. Highly discriminative statistical features for email classification. Knowledge and Information Systems, in press (doi:10.1007/s10115-011-0403-7).

Goodman, J., Heckerman, D., Rounthwaite, R., 2005. Stopping spam. Scientific American 292 (4), 42–88.

Guzella, T.S., Caminhas, W.M., 2009. A review of machine learning approaches to spam filtering. Expert Systems with Applications 36, 10206–10222.

Hartley, R., Schaffalitzky, F., 2004. PowerFactorization: 3d reconstruction with missing or uncertain data. In: Proceedings of the Australia-Japan Advanced Workshop on Computer Vision: AJAW 2003.

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

Hofmann, T., 1999. Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR. ACM, New York, pp. 50–57.

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

Ishii, N., Murai, T., Yamada, T., Bao, Y., Suzuki, S., 2006. Text classification: combining grouping, LSA and kNN vs support vector machine. In: Knowledge-Based Intelligent Information and Engineering Systems, Lecture Notes in Computer Science, vol. 4252, pp. 393–400.

Janecek, A.G.K., Gansterer, W.N., 2010. Utilizing Nonnegative Matrix Factorization for Email Classification Problems. John Wiley & Sons, Chichester.

Jolliffe, I.T., 1986. Principal Component Analysis. Springer, New York.

Kanaris, I., Kanaris, K., Houvardas, I., Stamatatos, E., 2007. Words vs. character *n*-grams for anti-spam filtering. International Journal on Artificial Intelligence Tools 16 (6), 1047–1067.

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

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

Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. Annals of Mathematical Statistics 18 (1), 50–60.

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

Morita, T., Kanade, T., 1997. A sequential factorization method for recovering shape and motion from image streams. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (8), 858–867.

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

Platt, J.C., 1998. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. MIT Press, Cambridge, MA, 185–208.

Pu, Q., Yang, G.-W., 2006. Short-text classification based on ICA and LSA. In: Advances in Neural Networks, vol. 3972, Lecture Notes in Computer Science. pp. 265–270.

Robinson, G., 2003. A statistical approach to the spam problem. Linux Journal 2003 (107), 58–64.

Sculley, D., Wachman, G.M., 2007. Relaxed online SVMs for spam filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference. ACM, New York, pp. 9–17.

Silva, C., Ribeiro, B., 2009. Knowledge extraction with non-negative matrix factorization for text classification. In: Proceedings of the 10th International Conference IDEAL 2009. Springer-Verlag, Berlin, pp. 300–308.

Torkkola, K., 2001. Linear discriminant analysis in document classification. In: Proceedings of the 2001 IEEE ICDM Workshop on Text Mining. IEEE, Los Alamitos.

Vidal, R., Tron, R., Hartley, R., 2008. Multiframe motion segmentation with missing data using PowerFactorization and GPCA. International Journal of Computer Vision 79 (1), 85–105.

Witten, I.H., Frank, E., 2000. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Francisco, California.

Xia, Y., Wong, K.-F., 2006. Binarization approaches to email categorization. In: Proceedings of the 23rd Annual ACM symposium on Applied Computing: SAC 2008. ACM, New York, pp. 474–481.

Yu, B., Xu, Z.-B., 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowledge-Based Systems 21 (4), 355.