International Journal of Security and Its Applications
Vol. 10, No. 1 (2016), pp.31-44
http://dx.doi.org/10.14257/ijsia.2016.10.1.04

# Email Spam Filtering Based on the MNMF Algorithm

Zun-xiong Liu, Shan-shan Tian, Zhi-qiang Huang, Jiang-wei Liu

*School of Information Engineering, East China Jiaotong University, Nanchang 330013, China*
*Darrent.liu@gmail.com*

***Abstract***

*Content-based email spam filtering is a challenging problem in which emails are often represented as high-dimensional data. This paper proposes an approach to email spam filtering based on max-margin semi-NMF (MNMF). MNMF combines the ideas of semi-NMF and max-margin and performs dimension reduction and classification simultaneously. In MNMF, we employ the same approach as Semi-NMF to update the coefficient matrix (while the other parameters are fixed) instead of quadratic programming. Simulation experiments were performed on two public Chinese email corpuses. The results show that MNMF is much faster and performs much better than support vector machine (SVM) classifiers that use features extracted by principal component analysis or linear discriminant analysis, and the MNMF method also outperforms SVM classification schemes in combination with feature extractions based on NMF and Semi-NMF.*

***Keywords:*** *email spam filtering; dimension reduction; support vector machine; non-negative matrix factorisation; max-margin semi-NMF*

## 1. Introduction

With the development of the Internet and the growth of email usage, spam—also known as unsolicited bulk email (UBE), junk mail, or unsolicited commercial email (UCE)—has also dramatically increased. Spam emails waste network resources and cost people time and attention in coping with the unwanted messages as well as information security problems. To combat spam emails, a variety of spam-filtering techniques have been developed; among these, automatic email filtering based on content analysis seems to be the most effective.

Content-based spam filtering can be regarded as the special case of text categorisation where the target values are spam and ham (non-spam) and where emails are represented as high-dimensional data in the vector space model (VSM) [1]. In this context, the problem called the curse of dimension arises. Specifically, such data not only give rise to more complex calculations but also make the generalisation learned from the spam filtering poor. Generally, dimension reduction is employed in email spam filtering. A few of the many dimension-reduction methods include principal component analysis (PCA), linear discriminate analysis (LDA), and non-negative matrix factorisation (NMF). In email spam filtering, the widely used classifiers are naive Bayesian (NB) [2], support vector machine (SVM) [3, 4], logistic regression (LR) [5] and neural network (NN) [6]. Bin Cui *et al*. [6] put forward the NN-based approach to classify personal emails with PCA to reduce dimension. Their experimental results showed that with the help of PCA, the data became more separable and the NN training converged much faster.

NMF is a popular dimension-reduction method, approximating the original data with a smaller number of dimensions. NMF has extensively been used in fields such as computer vision [7, 8] and pattern recognition [9, 10]. NMF aims at decomposing

ISSN: 1738-9976 IJSIA

a non-negative data matrix into a product of a non-negative basis matrix with a non-negative coefficient matrix. Andreas G. K. Janecek *et al*. [11] proposed a method to classify emails with SVM, where NMF was used for feature extraction. This approach performed better than classifying the original input data with the SVM classifier. In semi-non-negative matrix factorisation (Semi-NMF) [12], the non-negativity constraints were relaxed on the data matrix and the basis matrix. Vijay Kumar B.G *et al*. [13] introduced a max-margin framework for Semi-NMF and proposed an algorithm to solve it. Their approach was demonstrated to outperform discriminative NMF (DNMF) [14] and SVM classifiers with Semi-NMF for feature extraction. The traditional approaches to email spam filtering, which perform dimension reduction and classification in two separate steps, have achieved success to some degree [15, 16]. By contrast, dimension reduction and classification are carried out simultaneously in MNMF. In this paper, we propose a MNMF-based method for email spam filtering; this method updates the coefficient matrix within MNMF using the same approach as Semi-NMF instead of quadratic programming. The proposed method is evaluated on two public Chinese email corpuses (CDSCE [17] and trec06c [18]), and the experimental results show that it outperforms SVM classification schemes that use features extracted by PCA, LDA, NMF and Semi-NMF.

The paper is organised as follows. In the next section, NMF and Semi-NMF are briefly reviewed, and two simple algorithms for solving them are presented. In Section 3, an explanation of the MNMF algorithm is given. Section 4 presents spam-filtering experiments on two Chinese email corpuses and their results. Finally, conclusions of this study are drawn, and suggestions for future work are given in Section 5.

## 2. NMF and Semi-NMF

Email spam filtering can be treated as a binary classification problem where the incoming emails have to be classified as spam emails or ham emails. In this paper, VSM is used to represent emails. Thus, a given set of emails can be represented as a non-negative term-document matrix $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_i,\cdots,\mathbf{x}_n]\in\Re^{m\times n}$ and the corresponding label vector $\mathbf{y}=[y_1,\cdots,y_i,\cdots,y_n]\in\Re^{1\times n}$, where $m$ denotes the number of terms (also called the email dimension), $n$ denotes the number of emails, $\mathbf{x}_i\in\Re^m$ and $y_i\in\{-1,1\}$ ..

### 2.1. NMF

NMF decomposes a non-negative matrix into a product of two non-negative matrices with the reduced dimension [19]. In other words, given a non-negative matrix $\mathbf{X}$, NMF finds the non-negative basis matrix $\mathbf{W}\in\Re^{m\times r}$ and the non-negative coefficient matrix $\mathbf{H}\in\Re^{r\times n}$ such that $\mathbf{X}\approx\mathbf{WH}$. Usually, $r$ is chosen to be smaller than $n$ or $m$ in order to accomplish dimension reduction. The above approximate equation can be rewritten column by column as $\mathbf{x}_i\approx\mathbf{W}\mathbf{h}_i$, where $\mathbf{x}_i\in\Re^m$ and $\mathbf{h}_i\in\Re^r$ are the corresponding columns of $\mathbf{X}$ and $\mathbf{H}$, respectively. Thus, each vector $\mathbf{x}_i$ is approximated by a linear combination of the columns of $\mathbf{W}$, weighted by the components of $\mathbf{h}_i$.

To obtain the basis matrix and the coefficient matrix, the following optimisation problem of minimising the reconstruction error should be solved:

$$\min_{\mathbf{W}\geq 0,\mathbf{H}\geq 0} \| \mathbf{X}-\mathbf{WH} \|_F^2, \tag{1}$$

 

where $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$ denote, respectively, that all elements of $\mathbf{W}$ and $\mathbf{H}$ are non-negative, and $\|.\|_F$ is the Frobenius norm [20]. The multiplicative updating algorithm for NMF is as follows:

---

Algorithm for NMF

---

input: $\mathbf{X}$, $r$, *maxiter*.

output: $\mathbf{W}$, $\mathbf{H}$.

begin

initialise $\mathbf{W} = rand(m, r) \geq 0$, $\mathbf{H} = rand(r, n) \geq 0$, $iter = 0$.

repeat

Update $\mathbf{H} := \mathbf{H} \odot \frac{\mathbf{W}^T\mathbf{X}}{\mathbf{W}^T\mathbf{W}\mathbf{H}}$.

Update $\mathbf{W} := \mathbf{W} \odot \frac{\mathbf{X}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T}$.

$iter := iter + 1$.

until $iter \leq maxiter$ or convergence.

end

---

In the above algorithm, $\odot$ denotes element-wise multiplication, and division is also element-wise. Moreover, *iter* is the number of completed iterations, and *maxiter* is the maximum number of allowable iterations. The algorithm will be terminated when *iter* reaches *maxiter* or when the reconstruction error is less than some small threshold value ($10^{-6}$ is selected). In the latter case, we say that the algorithm is near convergence.

### 2.2 Semi-NMF

In Semi-NMF [12], the non-negativity constraints on the data matrix $\mathbf{X}$ and the basis matrix $\mathbf{W}$ are relaxed, and the iterative updating algorithm is used to solve this minimisation problem. Semi-NMF is based on the perspective of clustering, where the columns of $\mathbf{W}$ denote the different cluster centroids and $\mathbf{H}$ represents the information about cluster indicators. Semi-NMF is formulated as

$$\arg\min_{\mathbf{H} \geq 0} \| \mathbf{X} - \mathbf{W}\mathbf{H} \|_F^2 . \quad (2)$$

The algorithm for Semi-NMF is reviewed below. Note that $\mathbf{A}^+$ and $\mathbf{A}^-$ are the positive and negative part, respectively, of the matrix $\mathbf{A}$; they result from $\mathbf{A}_{ij}^+ = (|\mathbf{A}_{ij}| + \mathbf{A}_{ij})/2$ and $\mathbf{A}_{ij}^- = (|\mathbf{A}_{ij}| - \mathbf{A}_{ij})/2$.

---

Algorithm for Semi-NMF

---

input: $\mathbf{X}$, $r$, *maxiter*.
output: $\mathbf{W}$, $\mathbf{H}$.
begin
initialise $\mathbf{H} = rand(r, n) \geq 0$, $iter = 0$.
repeat
1. Update $\mathbf{W} := \mathbf{X}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$.

---

2. Update $\mathbf{H} := \mathbf{H} \odot \sqrt{\frac{[\mathbf{W}^T\mathbf{X}]^+ + [\mathbf{W}^T\mathbf{W}]^-\mathbf{H}}{[\mathbf{W}^T\mathbf{W}]^+\mathbf{H} + [\mathbf{W}^T\mathbf{X}]^-}}$.
3. $iter := iter + 1$.

until $iter \le maxiter$ or convergence.
end

## 3. Main Title

The idea of maximum margin, which first appeared with SVM classifications, has been successfully combined with other approaches, such as Semi-NMF [12] and clustering [21]. MNMF [13] resulted from introducing max-margin framework into Semi-NMF, and an iterative updating algorithm was established to solve it. In fact, all types of NMF variants are mainly obtained by relaxing non-negativity constraints or by introducing different discriminant constraints in the NMF optimisation problem; these variants include Semi-NMF [12], graph regularised non-negative matrix factorisation (GNMF) [22] and max-margin semi-NMF (MNMF) [13].

MNMF aims to perform max-margin SVM classification and matrix decomposition simultaneously; MNMF's general strategy is to introduce a soft-margin SVM classification constraint into the NMF optimisation problem and reformulate the problem as

$$\begin{aligned} &\arg\min_{\mathbf{W},\mathbf{H},\mathbf{w},b,\varepsilon} \lambda \| \mathbf{X} - \mathbf{WH} \|_F^2 + \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n}\varepsilon_i \\ &\text{s.t.} \quad \mathbf{H} \ge 0 \\ &\qquad y_i(\mathbf{w}^T\mathbf{W}^T\mathbf{x}_i + b) \ge 1 - \varepsilon_i,\ \varepsilon_i \ge 0,\ i = 1,\cdots,n. \end{aligned} \tag{3}$$

Here $\lambda > 0$ is the weight parameter, $C > 0$ is the penalty parameter, $\mathbf{w}$ and $b$ are the hyperplane parameters, $\mathbf{x}_i$ and $y_i$ are, respectively, the data vector and the corresponding label for the $i^{th}$ email, and the slack variable $\varepsilon_i$ is a measure of the misclassification error for the $i^{th}$ sample. Obviously, the first term in the above optimisation problem corresponds to the reconstruction error with Semi-NMF, and the remaining terms represent the soft-margin classification with SVM. In this way, the above formulation aims at minimising the reconstruction error and maximising the soft-margin simultaneously. Directly solving equation (3) is difficult, so an iterative updating algorithm was proposed to solve it. In MNMF, quadratic programming was utilised to obtain the coefficient matrix $\mathbf{H}$. Here we take the same approach as Semi-NMF for calculating the coefficient matrix $\mathbf{H}$ while keeping $\mathbf{W}$, $\mathbf{w}$, $b$ and $\varepsilon$ fixed. To solve the problem, an updating iteration contains the following three procedures.

**First, solve for $\mathbf{W}$ with $\mathbf{H}$, $\mathbf{w}$ and $b$ fixed.** So the optimisation problem becomes

$$\begin{aligned} &\arg\min_{\mathbf{W},\varepsilon} \| \mathbf{X} - \mathbf{WH} \|_F^2 + \frac{C}{\lambda}\sum_{i=1}^{n}\varepsilon_i \\ &\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{W}^T\mathbf{x}_i + b) \ge 1 - \varepsilon_i,\ \varepsilon_i \ge 0,\ i = 1,\cdots,n. \end{aligned} \tag{4}$$

With the Lagrange optimisation method, equation (4) is transformed into

$$L(\mathbf{W},\varepsilon,\alpha,\beta) = \| \mathbf{X} - \mathbf{WH} \|_F^2 + \frac{C}{\lambda}\sum_{i=1}^{n}\varepsilon_i - \sum_{i=1}^{n}\alpha_i[y_i(\mathbf{w}^T\mathbf{W}^T\mathbf{x}_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^{n}\beta_i\varepsilon_i, \tag{5}$$

where $\alpha$, $\beta$ are the non-negative Lagrange multiplier vectors with dimension $n$. Taking the partial derivatives of $L$ with respect to $\mathbf{W}$ and $\varepsilon$ and setting them equal to zero, we have

 

$$\frac{\partial L}{\partial \mathbf{W}} = 0 \Rightarrow \mathbf{W} = (2\mathbf{X}\mathbf{H}^T + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \mathbf{w}^T)(2\mathbf{H}\mathbf{H}^T)^{-1}$$
$$\frac{\partial L}{\partial \varepsilon_i} = 0 \Rightarrow \frac{C}{\lambda} - \alpha_i - \beta_i = 0,\ 0 \le \alpha_i \le \frac{C}{\lambda},\ i = 1, \cdots, n. \quad (6)$$

Substituting equation (6) into (5) and using linear algebra, we obtain the dual problem:

$$\arg\max_{\alpha} \alpha^T (\mathbf{T}_1 - \mathbf{T}_2)\alpha + (\mathbf{t}_3 - \mathbf{t}_4 - \mathbf{t}_5 - \mathbf{t}_6 + \mathbf{t}_7)\alpha$$
$$s.t. \quad 0 \le \alpha_i \le \frac{C}{\lambda},\ i = 1, \cdots, n, \quad (7)$$

where $\alpha \in \Re^n$ , $\mathbf{T}_1, \mathbf{T}_2 \in \Re^{n \times n}$ , $\mathbf{t}_3, \mathbf{t}_4, \mathbf{t}_5, \mathbf{t}_6, \mathbf{t}_7 \in \Re^{1 \times n}$ , $\mathbf{T}_1 = [\sum_{k=1}^{n} y_i y_j \mathbf{h}_k^T \mathbf{B}\mathbf{M}_i^T \mathbf{M}_j \mathbf{B}\mathbf{h}_k]_{ij}$ , $\mathbf{T}_2 = [y_i y_j \mathbf{w}^T \mathbf{B}\mathbf{M}_j^T \mathbf{x}_i]_{ij}$ , $\mathbf{t}_3 = [4\sum_{k=1}^{n} y_i \mathbf{h}_k^T \mathbf{B}\mathbf{H}\mathbf{X}^T \mathbf{M}_i \mathbf{B}\mathbf{h}_k]_{1i}$ , $\mathbf{t}_4 = [2\sum_{k=1}^{n} y_i \mathbf{h}_k^T \mathbf{B}\mathbf{w}\mathbf{x}_i^T \mathbf{x}_k]_{1i}$ , $\mathbf{t}_5 = [2y_i \mathbf{w}^T \mathbf{B}\mathbf{H}\mathbf{X}^T \mathbf{x}_i]_{1i}$ , $\mathbf{t}_6 = [by_i]_{1i}$ , $\mathbf{t}_7 = [1]_{1i}$ , $\mathbf{B} = (2\mathbf{H}\mathbf{H}^T)^{-1}$ , $\mathbf{M}_i = \mathbf{x}_i \mathbf{w}^T$ , and $\mathbf{h}_k$ is the $k^{th}$ column of the matrix $\mathbf{H}$. Here, the notation $\mathbf{t} = [v]_{ij}$ denotes that the element of the $i^{th}$ row and $j^{th}$ column of matrix $\mathbf{t}$ is $v$, where $v$ is calculated with those specific values of $i$ and $j$. The constant term $\frac{C}{\lambda}$ in equation (7) is a tuning parameter. When $\lambda >> C$, $\frac{C}{\lambda} \to 0$, and the elements of vector $\alpha$ tend to zero in equation (7). Hence, for large values of $\lambda$, the updating rule for $\mathbf{W}$ tends to be same as that of Semi-NMF. Furthermore, the above optimisation problem is a quadratic programming problem in $\alpha$ and can be solved with some conventional quadratic programming tools [23].

**Second, solve for $\mathbf{w}$, $b$, $\varepsilon$ while keeping $\mathbf{W}$ and $\mathbf{H}$ fixed.** In the case, the MNMF optimisation problem changes into the standard binary soft-margin SVM classification:

$$\arg\min_{\mathbf{w},b,\varepsilon} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \varepsilon_i$$
$$s.t. \quad y_i(\mathbf{w}^T \mathbf{W}^T \mathbf{x}_i + b) \ge 1 - \varepsilon_i \quad (8)$$
$$\varepsilon_i \ge 0,\ i = 1, \cdots, n.$$

After introducing the Lagrange multipliers $\alpha_i \ge 0$ and $\beta_i \ge 0 (i = 1, \cdots, n)$ [24], we get the Lagrange function $L(\mathbf{w}, b, \varepsilon, \alpha, \beta) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{n} \varepsilon_i - \sum_{i=1}^{n} \alpha_i [y_i (\mathbf{w}^T \mathbf{W}^T \mathbf{x}_i + b) - 1 + \varepsilon_i] - \sum_{i=1}^{n} \beta_i \varepsilon_i$ .

Additionally, the dual problem of equation (8) is

$$\arg\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \left\langle \mathbf{W}^T \mathbf{x}_i, \mathbf{W}^T \mathbf{x}_j \right\rangle$$
$$s.t. \quad 0 \le \alpha_i \le C,\ i = 1, \cdots, n \quad (9)$$
$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

Here $\langle a, b \rangle$ denotes the inner product between the two vectors $a$ and $b$. Solving the above problem, we get the vector $\alpha$. Then, the optimal separating hyperplane is given by

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{W}^T \mathbf{x}_i$$
$$b = -\frac{1}{2}\left\langle \mathbf{w}, \mathbf{W}^T \mathbf{x}_r + \mathbf{W}^T \mathbf{x}_s \right\rangle, \quad (10)$$

where $\mathbf{W}^T \mathbf{x}_r$ and $\mathbf{W}^T \mathbf{x}_s$ are any support vector from each class satisfying $\alpha_r$, $\alpha_s > 0$ and $y_r \ne y_s$

**Finally, solve for $\mathbf{H}$ with $\mathbf{W}$, $\mathbf{w}$, $b$ and $\varepsilon$ fixed.** Because only the reconstruction error term of the MNMF cost function depends on $\mathbf{H}$, we obtain the following optimisation problem:

$$\begin{aligned} &\arg\min_{\mathbf{H}} \| \mathbf{X} - \mathbf{W}\mathbf{H} \|_F^2 \\ &\text{s.t.} \quad \mathbf{H} \geq 0. \end{aligned} \tag{11}$$

It is apparent that the optimisation problem in equation (11) is the same as Semi-NMF. Because the number of samples is large, the computational efficiency of the quadratic programming method to solve equation (11) in MNMF [13] is low. Therefore, we use the second step of Semi-NMF algorithm instead of the quadratic programming method:

$$\mathbf{H} := \mathbf{H} \odot \sqrt{\frac{[\mathbf{W}^T\mathbf{X}]^+ + [\mathbf{W}^T\mathbf{W}]^- \mathbf{H}}{[\mathbf{W}^T\mathbf{W}]^+ \mathbf{H} + [\mathbf{W}^T\mathbf{X}]^-}}. \tag{12}$$

Based on the above analyses, we can describe the version of the MNMF algorithm used in our experiments as follows:

---

Algorithm for the MNMF Used Here

---

input: $\mathbf{X}$, $r$, $maxiter$, $y$, $\lambda$, $C$.

output: $\mathbf{W}$, $\mathbf{H}$, $\mathbf{w}$, $b$.

begin

initialise $\mathbf{W} = rand(m,r)$, $\mathbf{H} = rand(r,n) \geq 0$, $iter = 0$.

repeat

1. If $iter == 0$, Compute $\mathbf{W}$ using the first step of Semi-NMF algorithm;
   Else Compute $\mathbf{W}$ using Equation (6) and Equation (7).
2. Compute $\mathbf{w}$ and $b$ using Equation (10).
3. Compute $\mathbf{H}$ using Equation (12).
4. $iter := iter + 1$.

until $iter \leq maxiter$ or convergence.

end

---

Given the training dataset, all of the basis matrix $\mathbf{W}$, the coefficient matrix $\mathbf{H}$ and the hyperplane parameters $\mathbf{w}$ with $b$ are calculated by the MNMF model as it learns. With a new sample $\mathbf{x}_{test}$, the feature vector $f_{test} = \mathbf{W}^T\mathbf{x}_{test}$ can be obtained, and the predicted label for it will be determined by $\mathbf{y}_{test} = sign(\mathbf{w}^T f_{test} + b)$.

## 4. Experiments

### 4.1. Data Sets and Evaluation Metrics

We used MNMF for email spam filtering and performed experiments on two public Chinese email corpuses, CDSCE (CCERT Data Sets of Chinese Emails) and trec06c. These corpuses are derived from the CERNET Computer Emergency Response Team (CCERT) and the Text Retrieval Conference (TREC), respectively. CDSCE contains 45396 spam emails and 18314 ham emails, which were collected during June and July 2005. We randomly selected 2000 spam emails and 2000 ham emails for the simulation experiments. To obtain a term-document matrix, some pre-processing steps needed to be executed, mainly including Chinese word segmentation, email representation, and feature selection. With the two corpuses, only the contents of email subject and body (the two important constitutions of email) were extracted, and the messages were preprocessed with the Chinese word segmentation tool ICTCLAS [25], which was developed by Chinese Academy of

 

Sciences. VSM was utilised for email representation, and the ltc weighting [26] was selected for the term weighting. In this case, the term-weighting formula is

$$a_{ij} = \frac{\log(f_{ij}+1.0)\times\log(n/n_i)}{\sqrt{\sum_{i=1}^{m}\left[\log(f_{ij}+1.0)\times\log(n/n_i)\right]^2}}, \quad (13)$$

where $a_{ij}$ is the weight of the term $i$ in the email $j$, $f_{ij}$ is the occurrence frequency of the term $i$ in the email $j$, $n$ is the number of all emails, $n_i$ is the number of emails with the term $i$, and $m$ is the number of all terms. Thus, the obtained data are denoted in a high-dimensional matrix. After that, the Chi-square statistic (CHI) [27] was used in the experiments for feature selection, and 2000 features were selected. The samples with noise values, all-zero values, *etc*., were deleted, and the ratio of spam to ham was set to 1:1. Finally the term-document matrix with the selected CDSCE data was obtained; the matrix has a size of 2000×3900 and contains 1950 spam emails and 1950 non-spam emails.

The trec06c corpus provided by TREC 2006 spam track contains 42854 spam emails and 21766 ham emails, among which 5000 spam emails and 5000 legitimate emails were selected to be preprocessed with the same method as used on the CDSCE data. The result is a 2000×10000 term-document matrix.

Some evaluation metrics used in our experiments to assess email classification performance are precision, recall and F1 [28]. Assuming that $n$ emails are tested with the trained spam-filtering system, a confusion matrix can be obtained, as shown in Table 1.

**Table 1. Confusion Matrix of the Email Classification Results**

| $n = a+b+c+d$ | | Actual Class | |
|---|---|---|---|
| | | Spam | Ham |
| Predicted Class | Spam | $a$ | $b$ |
| | Ham | $c$ | $d$ |

From Table 1, precision is defined as $P = \frac{a}{a+b}$, reflecting the ability of the filtering system to find spam correctly. The higher the precision is, the lower the misclassification rate of ham is. Additionally, recall reflects the ability of the filtering system to discover spam, given by $R = \frac{a}{a+c}$. The higher the recall is, the lower the misclassification rate of spam is. Higher precision or recall may indicate better performance. However, precision and recall are opposite metrics. In our experiments, F1, the harmonic mean of precision and recall, is selected as the main evaluation measure. In terms of both precision and recall, F1 is defined as follows:

$$F1 = \frac{2\times P\times R}{P+R}. \quad (14)$$

### 4.2. Designs and Results Analysis

To assess MNMF-based email spam filtering, three sets of comparative experiments were designed. First, in Experiment E1, the performances of MNMF, SVM classifier with PCA and SVM classifier with LDA are compared on the criteria of the varying dimensions; the projection classification with $r = 2$ is shown. Second, the experiments with NMF plus SVM, Semi-NMF plus SVM, and MNMF are conducted; these experiments together are denoted as Experiment E2. Finally, Experiment E3 is conducted with NMF plus SVM, Semi-NMF plus SVM, and MNMF; each is tested with a varied number of loops.

In all experiments, SVM and the second procedure in MNMF's updating iteration are solved by LIBSVM [29], and PCA and LDA are implemented with the Statistical Pattern Recognition Toolbox [30]. Linear kernel functions are used for all our experiments because linear methods are mainly considered. In addition, the number of the training and testing samples for each dataset are the same, and each sample has the ratio of ham to spam of 1:2 because there are more spam emails than ham.

**4.2.1 Experiment E1:** In Experiment E1, the parameters (the weight parameter $\lambda$, the penalty parameter $C$ and the number of maximum iterations *maxiter* ) are set empirically. The parameter $\lambda$ is set to 10 here and in Experiments E2 and E3. Meanwhile, $C$ and *maxiter* are set to 50 and 4, respectively. According to the above-specified ratios, both the training and testing samples from trec06c are chosen randomly to contain 487 ham and 974 spam emails (not all the processed data are used) and do not overlap. Then, the training and testing of PCA plus SVM, LDA plus SVM, and MNMF on the feature dimensions of $2,5,10,20,30,50,70,100$ are conducted. The experimental results are shown in Table 2, Table 3 and Figure 1. Table 2 and Table 3 show the F1 values and the training times of these three methods. For dimension equal to 2 ($r=2$), Figure 1 depicts the projection of the testing samples (on the most significant basis vectors) to points and the SVM separating hyperplanes under PCA plus SVM, LDA plus SVM, and MNMF (at different iterations).

**Table 2. Comparison of F1 between PCA plus SVM, LDA Plus SVM, and MNMF**

| Method | *r* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | 5 | 10 | 20 | 30 | 50 | 70 | 100 |
| PCA+SVM | 0.9026 | 0.9152 | 0.9264 | 0.9356 | 0.9384 | 0.9517 | 0.9515 | 0.9546 |
| LDA+SVM | 0.9321 | 0.9321 | 0.9336 | 0.9331 | 0.9330 | 0.9325 | 0.9320 | 0.9315 |
| MNMF | 0.9491 | 0.9491 | 0.9564 | 0.9558 | 0.9609 | 0.9601 | 0.9561 | 0.9580 |

**Table 3. Comparison of Training Time (in seconds) of the Three Methods, as Described in Table 2**

| Method | *r* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 2 | 5 | 10 | 20 | 30 | 50 | 70 | 100 |
| PCA+SVM | 58.769+ 0.057 | 58.064+ 0.087 | 57.934+ 0.056 | 57.670+ 0.165 | 58.123+ 0.485 | 57.807+ 0.582 | 58.445+ 0.529 | 57.711+ 0.961 |
| LDA+SVM | 111.239+ 0.003 | 113.397+ 0.005 | 111.617+ 0.010 | 112.030+ 0.021 | 111.674+ 0.033 | 112.957+ 0.059 | 126.635+ 0.088 | 117.504+ 0.137 |
| MNMF | 41.077 | 41.801 | 42.303 | 45.577 | 43.462 | 46.723 | 49.128 | 57.376 |

 

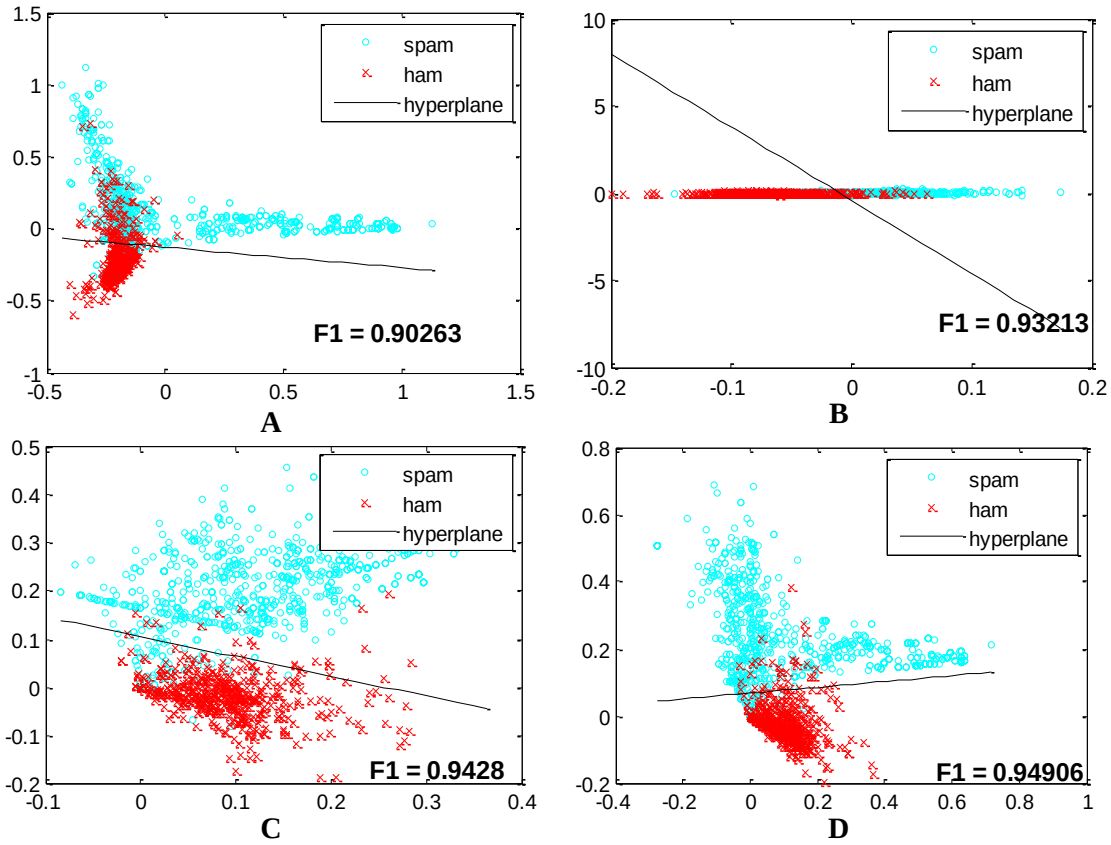


**Figure 1. The Projection of Testing Samples and the Separating Hyperplanes Obtained with Training Samples using (A) PCA, (B) LDA, (C) MNMF bases ($2^{nd}$ Iteration) and (D) MNMF bases ($4^{th}$ Iteration), Respectively**

The results in Table 2 show that, for a fixed feature dimension, the F1 value from MNMF is always higher than the F1 values from PCA plus SVM and from LDA plus SVM. Moreover, as the feature dimension is set larger and larger, MNMF, PCA plus SVM or LDA plus SVM generally performs better and better, and MNMF produces a good F1 value even when the feature dimension is low. In Table 3, for the cells with two values linked with plus, the previous one is the time spent on feature extraction, and the latter is the time needed for SVM classifiers. Table 3 shows that MNMF spends less time than the other two approaches on training with the selected feature dimensions and is at least twice as fast than LDA plus SVM, which is the slowest among the three approaches. Furthermore, in PCA plus SVM or LDA plus SVM much more time is spent on feature extraction than in SVM classification because of the LIBSVM-based implementation. Thus, MNMF not only performs the best but also spends the least time.

Figure 1(A) and Figure 1(B) depict the projection of testing samples on two significant base vectors produced by PCA and LDA, respectively, and on the SVM separating hyperplanes resulting from the training samples. Figure 1(C) and Figure 1(D) display the projection and the separating hyperplanes from MNMF after the second and the fourth iterations. Figure 1 also shows that MNMF can obtain the higher F1 value with a small number of iterations, and the projection of testing samples on MNMF bases can be classified more easily.

Hence, MNMF performs better than the traditional methods with SVM plus PCA and SVM plus LDA in email spam filtering.

**4.2.2 Experiment E2:** In Experiment E2, experiments are conducted on CDSCE and trec06c, and *maxiter* takes the default value of 6 (the same as in E3). Each of the two datasets is partitioned both randomly and equally into the training sample and the testing sample, where the ratio of ham to spam is 1:2 in each. With each of the term-documents (their sizes are 2000×3900 and 2000×10000, respectively), training and testing samples with the same size (1461 samples) are obtained; both consist of 487 ham and 974 spam emails. The three approaches, NMF plus SVM, Semi-NMF plus SVM, and MNMF, are trained and tested on trec06c with different penalty parameter values (while the dimension is set to 2); then the F1 values are calculated and shown in Table 4. Using these F1 values, we choose a value for the penalty parameter $C$. Table 4 shows that when $C=1$, MNMF achieves its smallest F1 value. Thus, we fix $C=1$ from now on, including for the experiment E3.

**Table 4. F1 for NMF Plus SVM, Semi-NMF Plus SVM, and MNMF with Different Penalty Parameters $C$**

| Method | $C$ | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 10 | 20 | 50 | 70 | 100 |
| NMF+SVM | 0.8665 | 0.8665 | 0.8670 | 0.8665 | 0.8663 | 0.8670 |
| Semi-NMF+SVM | 0.8600 | 0.8652 | 0.8651 | 0.8740 | 0.8707 | 0.8707 |
| MNMF | 0.9256 | 0.9668 | 0.9561 | 0.9694 | 0.9673 | 0.9489 |

In the experiment E2, the models of NMF plus SVM, Semi-NMF plus SVM, and MNMF on different feature dimensions are trained and tested on two different datasets. The F1 values of the testing samples obtained from the above three methods are plotted in Figure 2.

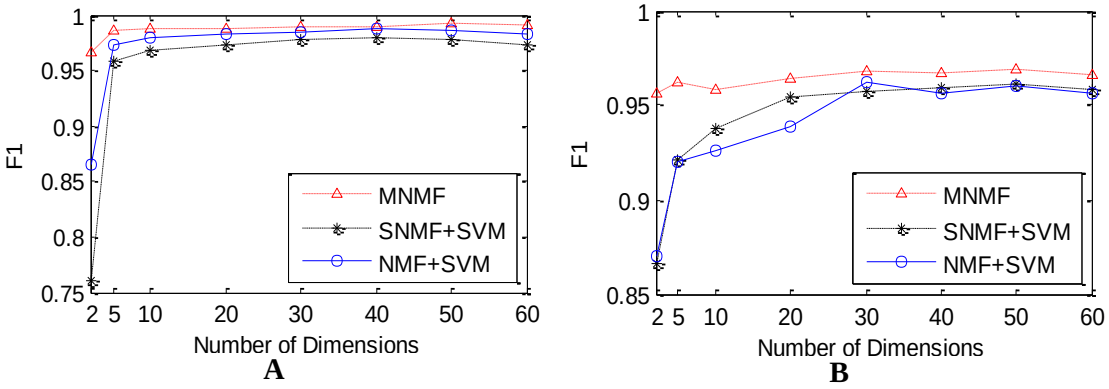


**Figure 2. Comparison of the F1 Values of MNMF with NMF plus SVM and Semi-NMF Plus SVM on the CDSCE (A) and Terc06c (B)**

Figure 2 shows that in each dimension, MNMF performs the best, and its F1 values are always higher than those of the other two methods. Moreover, MNMF achieves a high F1 value even when the feature dimension is low. In particular, the smaller the number of dimensions is, the more outstandingly MNMF outperforms the other methods. As the number of dimensions increases, the curve of F1 values with respect to feature dimension approaches the same value for each method. The F1 values of the three methods approach stable states when the dimension is 10 for CDSCE and 30 for trec06c; these values are employed in E3 to test the algorithms' stability. These results are consistent with the results of Experiment E1.

**4.2.3 Experiment E3:** Experiment E3 compares the stability of NMF plus SVM, Semi-NMF plus SVM, and MNMF for email spam filtering. With CDSCE and

trec06c, the training and testing data are obtained in the same way described in Experiment E2. The data obtained are used in NMF plus SVM, Semi-NMF plus SVM, and MNMF with the various numbers of loops; these numbers denote the number of iterations of some experiment. The mean and variance of F1 values are listed in Table 5.

**Table 5. F1's Mean and Variance from NMF Plus SVM, Semi-NMF Plus SVM, and MNMF (%)**

| Dataset | Method | Number of loops | | | | Avg. |
|---|---|---|---|---|---|---|
| | | 5 | 10 | 15 | 20 | |
| CDSCE | NMF+SVM | 98.14±0.31 | 97.64±0.46 | 97.86±0.33 | 97.73±0.30 | 97.84 |
| | Semi-NMF+SVM | 97.39±0.38 | 97.70±0.44 | 97.45±0.57 | 97.49±0.41 | 97.51 |
| | MNMF | 98.76±0.12 | 98.66±0.34 | 98.66±0.25 | 98.61±0.25 | 98.67 |
| trec06c | NMF+SVM | 94.84±0.73 | 95.09±0.71 | 95.05±0.87 | 94.90±0.82 | 94.97 |
| | Semi-NMF+SVM | 94.91±0.92 | 95.10±0.59 | 95.07±0.62 | 95.02±0.68 | 95.03 |
| | MNMF | 95.96±0.54 | 96.82±0.49 | 96.57±0.59 | 96.42±0.67 | 96.44 |

Table 5 shows that, under each fixed number of loops, the F1's means from MNMF are the best, and the F1's variances are the lowest among the different dataset experiments. Moreover, for each dataset, the average of MNMF is the highest among all the methods. With the different number of loops, the related algorithms perform well and eventually approximate a stable state. In particular, the MNMF algorithm performs the best with respect to stability.

## 5. Conclusions and Future Work

This paper proposes an approach to email spam filtering based on MNMF. To solve MNMF, the same approach as Semi-NMF is used to update the coefficient matrix instead of quadratic programming. Our proposed method was used to classify two corpuses of Chinese emails, and those results compared with the results obtained by some other methods. The experimental results show that the proposed method performs better than other methods composed of SVM classification and PCA, LDA, NMF or Semi-NMF for dimension reduction. Moreover, the proposed approach takes less time to converge than PCA plus SVM and LDA plus SVM. In future work, we will explore incorporating a nearest neighbour graph structure into the cost function of MNMF and applying MNMF in other fields, such as predicting financial failures.

## Acknowledgments

This work was supported by National Natural Science Foundation of China (61065003), Social Science Foundation of the State Education Ministry (13YJC630192) and Graduate Innovation Fund of East China Jiaotong University (YC2013-S172).

## References

[1] G. Salton, A. Wong and C. S. Yang, "A Vector Space Model for Automatic Indexing," Communications of the ACM, vol. 18, no. 11, **(1975)**, pp. 613-620.
[2] S. Youn and D. McLeod, "A Comparative Study for Email Classification," Advances and Innovations in Systems, Computing Sciences and Software Engineering, **(2007)**, pp. 387-391,
[3] H. Drucker, W. Donghui and V. N. Vapnik, "Support Vector Machines for Spam Categorization," IEEE Transactions on Neural Networks, vol. 10, no. 5, **(1999)**, pp. 1048-1054.
[4] O. Amayri and N. Bouguila, "A study of spam filtering using support vector machines," Artificial Intelligence Review, vol. 34, no. 1, **(2010)**, pp. 73-108.

[5] M. Chang, W. Yih and C. Meek, "Partitioned Logistic Regression for Spam Filtering," in Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data mining, New York, NY, USA, **(2008)**, pp. 97-105
[6] B. Cui, A. Mondal, J. Shen, G. Cong, and K. Tan, "On Effective E-mail Classification via Neural Networks," in Proceedings of the 16th International Conference on Database and Expert Systems Applications, Berlin, Heidelberg, **(2005)**, pp. 85-94.
[7] C. Thurau and V. Hlavac, "Pose primitive based human action recognition in videos or still images," in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, **(2008)**, pp. 1-8.
[8] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," in Proceedings of the Asian Conference on Computer Vision, Jeju Island, Korea, **(2004)**, pp. 27-30.
[9] W. Xu, X. Liu and Y. Gong, "Document Clustering Based On Non-negative Matrix Factorization," in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, **(2003)**, pp. 267-273.
[10] W. Liu, N. Zheng and X. Li, "Nonnegative Matrix Factorization for EEG Signal Classification," in Proceedings of International Symposium on Neural Networks, **(2004)**, pp. 470-475.
[11] A. G. K. Janecek and W. N. Gansterer, "Utilizing Nonnegative Matrix Factorization for Email Classification Problems," in Text Mining: Applications and Theory, M. W. Berry and J. Kogan, Eds. Chichester, West Sussex, UK: John Wiley & Sons, Ltd, **(2010)**, pp. 57-80.
[12] C. H. Q. Ding, T. Li and M. I. Jordan, "Convex and Semi-Nonnegative Matrix Factorizations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 1, **(2010)**, pp. 45-55.
[13] B. G. V. Kumar, I. Kotsia and I. Patras, "Max-Margin Semi-NMF," in Proceedings of the British Machine Vision Conference, **(2011)**, pp. 129.1-129.11.
[14] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting Discriminant Information in Nonnegative Matrix Factorization With Application to Frontal Face Verification," IEEE Transactions on Neural Networks, vol. 17, no. 3, **(2006)**, pp. 683-695.
[15] H. Yin, F. Cheng and D. Zhang, "Using LDA and Ant Colony Algorithm for Spam Mail Filtering," in Proceedings of the 2009 Second International Symposium on Information Science and Engineering, Washington, DC, USA, **(2009)**, pp. 368-371.
[16] R. Wang, Feature Selection Strategies for Spam E-mail Filtering. Montreal, Quebec, Canada: Concordia University, **(2006)**.
[17] Q. Tran, "CCERT Data Sets of Chinese Emails (CDSCE)," **(2005)**. http://www.ccert.edu.cn/spam/sa/datasets.htm.
[18] G. Cormack, "TREC 2006 Spam Track Overview," in Proceedings of 15th Text REtrieval Conference, **(2006)**.
[19] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in Proceedings of Neural Information Processing Systems, **(2001)**, pp. 556-562.
[20] Y. Deng and D. Boley, "An optimal approximation problem for a matrix equation," International Journal of Computer Mathematics, vol. 86, no. 2, **(2009)**, pp. 321-332,.
[21] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans, "Maximum Margin Clustering," in Advances in Neural Information Processing Systems, 2004, pp. 1537-1544.
[22] D. Cain, X. He, J. Han, and T. S. Huang, "Graph Regularized Non-negative Matrix Factorization for Data Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, **(2011)**. pp. 1548-1560,
[23] MathWorks, "MATLAB Optimization Toolbox, User's Guide, version 5," **(2010)**.
[24] S. R. Gunn, "Support Vector Machines for Classification and Regression," ISIS Technical Report, School of Electronics and Computer Science, University of Southampton, U.K., **(1998)**.
[25] ICTCLAS, http://ictclas.org/.
[26] C. Buckley, G. Salton, J. Allan, and A. Singhal, "Automatic Query Expansion Using SMART: TREC 3," in Proceedings of the Third Text REtrieval Conference, **(1995)**, pp. 69-80
[27] H. Schütze, D. A. Hull and J. O. Pedersen, "A Comparison of Classifiers and Document Representations for the Routing Problem," in Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, **(1995)**, pp. 229-237.
[28] L. Zhang, J. Zhu and T. Yao, "An Evaluation of Statistical Spam Filtering Techniques," ACM Transactions on Asian Language Information Processing, vol. 3, no. 4, **(2004)**, pp. 243-269,
[29] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 27:1-27:27, **(2011)**. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.
[30] V. Franc and V. Hlavac, "Statistical Pattern Recognition Toolbox for Matlab," Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, Research Report, **(2004)**, pp. 1213-2365,. Software available at http://cmp.felk.cvut.cz/cmp/software/stprtool/.

 

## Authors

**Zunxiong Liu,**he was born in Jiangxi province,China,in 1966.He received his PHD degree in computer science and technology from Xi'an Jiaotong University,Xi'an,China.Now he is a professor in East China Jiaotong University,Jiangxi,China.His current research interests includes data mining,pattern recognition,finacial data anaiysis and statistical information processing.

**Shanshan Tian,**she was born in Henan province,China,in 1991.She received her B.E. degree in Information and Computing Science from Xinxiang University,Henan,China.Now she is pursuing her M.E.degree in comouter application technology in East China Jiaotong University,Jiangxi,China.She research interest concentrates on mutiple test.

**Zhiqiang Huang,**he was born in Jiangxi province,China,in 1989.He received his B.E. degree in computer science and technology from East China Jiaotong University,Jiangxi,China. He also received his M.E. degree in computer science and technology from East China Jiaotong University,Jiangxi,China.

**Jiangwei Liu,**he was born in Henan province,China,in 1983.He received his B.E. degree in computer science and technology from Henan University of Scinece and Technology,Henan,China.Now he is pursuing his M.E.degree in comouter application technology in East China Jiaotong University,Jiangxi,China.His research interest concentrates on data mining and financial data analysis.