Front. Electr. Electron. Eng. China 2011, 6(2): 192–200
DOI 10.1007/s11460-011-0128-0

RESEARCH ARTICLE

Lijun ZHANG, Zhengguang CHEN, Miao ZHENG, Xiaofei HE

Robust non-negative matrix factorization

Abstract Non-negative matrix factorization (NMF) is a recently popularized technique for learning parts-based, linear representations of non-negative data. The traditional NMF is optimized under the Gaussian noise or Poisson noise assumption, and hence not suitable if the data are grossly corrupted. To improve the robustness of NMF, a novel algorithm named robust non-negative matrix factorization (RNMF) is proposed in this paper. We assume that some entries of the data matrix may be arbitrarily corrupted, but the corruption is sparse. RNMF decomposes the non-negative data matrix as the summation of one sparse error matrix and the product of two non-negative matrices. An efficient iterative approach is developed to solve the optimization problem of RNMF. We present experimental results on two face databases to verify the effectiveness of the proposed method.

Keywords robust non-negative matrix factorization (RNMF), convex optimization, dimensionality reduction

1 Introduction

In real world applications as diverse as information retrieval, remote sensing, biology and economics, one is often confronted with high-dimensional data. Because of the *curse of dimensionality*, procedures that are analytically or computationally manageable in low-dimensional spaces can become completely impractical in high dimensions [1]. For example, nearest neighbor methods usually break down when applied to high-dimensional data, because the neighborhoods are no longer *local* [2]. As a result, dimensionality reduction [3] has become an essential data preprocessing step in most data mining applications.

During the past decades, various techniques of matrix factorization have been used to find the low-dimensional structure hidden in the original data. The most famous matrix factorization based dimensionality reduction methods include principal component analysis (PCA) [4], latent semantic indexing (LSI) [5] and non-negative matrix factorization (NMF) [6]. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{n\times m}$ be the data matrix consisting of n features and m samples. All three methods construct approximate factorizations of the form:

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}, \tag{1}$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{H} \in \mathbb{R}^{r\times m}$. The columns of $\boldsymbol{W}$ is regarded as basis vectors, and the columns of $\boldsymbol{H}$ give the new low-dimensional representations for the m samples. As can be seen, all three methods learn to represent each sample as a linear combination of the basis vectors.

In PCA and LSI, the columns of $\boldsymbol{W}$ are constrained to be orthonormal, and the matrix $\boldsymbol{H}$ is obtained by projecting the data samples onto the subspace spanned by the columns of $\boldsymbol{W}$, i.e., $\boldsymbol{H} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}$. PCA, which is based on eigen decomposition, reduces the dimensionality of the data by finding a few orthonormal projections (columns in $\boldsymbol{W}$) such that the variance of the projected data is maximized. In fact, it turns out that these projections are just the leading eigenvectors of the data's covariance matrix, which are called principal components. Different from PCA, LSI is based on singular value decomposition (SVD) [7]. LSI projects the data samples onto a low-dimensional space using the left singular vectors of $\boldsymbol{X}$. Also, it can be proved that the product $\boldsymbol{W}\boldsymbol{H}$ is the best low-rank approximation to $\boldsymbol{X}$.

One drawback of PCA and LSI is that the negative values appeared in the projections make the factorization hard to interpret. When working in the domains where data are naturally non-negative, we would like to find non-negative basis vectors and represent the samples as non-negative combinations of these basis vectors. NMF [6] is such a technique for factorizing the non-negative data matrix $\boldsymbol{X}$ as the product of two non-

Received September 1, 2010; accepted November 4, 2010

Lijun ZHANG (✉), Zhengguang CHEN, Miao ZHENG
Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027, China
E-mail: zljzju@zju.edu.cn
Xiaofei HE
State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou 310058, China

negative matrices. The non-negative constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. Compared with PCA and LSI, the results of NMF are more consistent with human intuition. And NMF has been successfully applied to document clustering [8], face recognition [9,10] and microarray analysis [11,12].

In NMF, the two matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ are found by minimizing the approximation error subject to the non-negative constraints. Traditionally, the approximation error is measured by the square Euclidean distance or the generalized Kullback-Leibler divergence between $\boldsymbol{X}$ and $\boldsymbol{WH}$ [13]. These two cost functions are optimal if the approximation error is caused by Gaussian noise or Poisson noise [14]. However, in reality, it is very common that some entries of the data are grossly corrupted, which violates these noise assumptions significantly. For example, due to the sensor failure or the presence of obstacles, pixels of a digital image may change violently.

Inspired by the recent studies in robust principal component analysis [15], we propose a novel algorithm named Robust non-negative matrix factorization (RNMF). In RNMF, the errors in the data can take arbitrary values, but are assumed to be sparse. Specifically, we introduce an error matrix $\boldsymbol{S}$ to explicitly capture the sparse corruption. The non-negative data matrix $\boldsymbol{X}$ is then decomposed as $\boldsymbol{WH}+\boldsymbol{S}$, where $\boldsymbol{W}$ and $\boldsymbol{H}$ are constrained to be non-negative, and $\boldsymbol{S}$ is required to be sparse. An iterative approach is introduced to efficiently compute a solution by solving a sequence of convex optimization problems. Experiments on two face data sets have demonstrated the advantages of RNMF.

The rest of the paper is organized as follows. In Sect. 2, we provide a brief description of the related work. Our proposed RNMF is introduced in Sect. 3. In Sect. 4, we develop an efficient iterative approach to solve the optimization problem. Experiments are presented in Sect. 5. Finally, we provide some concluding remarks and suggestions for future work in Sect. 6.

2 Related work

In this section, we give a brief review of three most famous matrix factorization based dimensionality reduction methods.

2.1 PCA

PCA [4] can be defined in terms of the orthogonal projections that maximize the variance in the projected subspace. Given a set of n-dimensional samples $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m$, we first consider the projection onto a one-dimensional space using an n-dimensional vector $\boldsymbol{w}$. Let $\bar{\boldsymbol{x}}$ be the sample mean. The optimization problem of PCA is given by

$$\begin{aligned}\max_{\boldsymbol{w}}\ & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{w}\\ \text{s.t.}\ & \boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=1,\end{aligned}\tag{2}$$

where $\boldsymbol{C}$ is the data covariance matrix defined by

$$\boldsymbol{C}=\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i-\overline{\boldsymbol{x}})(\boldsymbol{x}_i-\overline{\boldsymbol{x}})^{\mathrm{T}}.\tag{3}$$

It can be proved that the optimal $\boldsymbol{w}$ equals to the eigenvector of $\boldsymbol{C}$ associated with the largest eigenvalue. This eigenvector is called as the first principal component. If we consider the general case of an r-dimensional projection space, the optimal linear projections for which the variance of the projected data is maximized are just the r leading eigenvectors $\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_r$ of $\boldsymbol{C}$.

2.2 LSI

LSI [5] is one of the most popular algorithms for text analysis. Suppose the rank of the data matrix $\boldsymbol{X}\in\mathbb{R}^{n\times m}$ is k. LSI factorizes $\boldsymbol{X}$ using SVD [7]:

$$\boldsymbol{X}=\boldsymbol{U}\Sigma\boldsymbol{V}^{\mathrm{T}},\tag{4}$$

where $\Sigma=\mathrm{diag}(\sigma_1,\sigma_2,\ldots,\sigma_k)\in\mathbb{R}^{k\times k}$ is the diagonal matrix consisting of the k singular values of $\boldsymbol{X}$. $\boldsymbol{U}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k]\in\mathbb{R}^{n\times k}$ and $\boldsymbol{V}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_k]\in\mathbb{R}^{m\times k}$ contain the k left singular vectors and right singular vectors of $\boldsymbol{X}$, respectively. To project the data samples onto an r-dimensional subspace, LSI uses the r left singular vectors of $\boldsymbol{X}$ associated with the largest singular values as the projections. It is easy to check that the left singular vectors are just the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$. Thus, if the samples have a zero mean, LSI is equivalent to PCA.

LSI can also be interpreted as finding the best low-rank approximation to $\boldsymbol{X}$. Let $\boldsymbol{W}\in\mathbb{R}^{n\times r}$ be the projection matrix. The objective function of LSI can be stated below:

$$\begin{aligned}\min_{\boldsymbol{W}}\ & \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X}\|_{\mathrm{F}}^2\\ \text{s.t.}\ & \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=I,\end{aligned}\tag{5}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the matrix Frobenius norm.

2.3 NMF

Given a non-negative data matrix $\boldsymbol{X}\in\mathbb{R}^{n\times m}$, NMF [6] aims to find two non-negative matrices $\boldsymbol{W}\in\mathbb{R}^{n\times r}$ and $\boldsymbol{H}\in\mathbb{R}^{r\times m}$ such that

$$\boldsymbol{X}\approx\boldsymbol{W}\boldsymbol{H}.\tag{6}$$

The non-negative constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations.

To find an approximate factorization, we need to define cost functions that quantify the quality of the approximation. The two most popular cost functions are:

1) The square Euclidean distance between $\boldsymbol{X}$ and $\boldsymbol{WH}$, i.e.,

$$\|\boldsymbol{X}-\boldsymbol{WH}\|_{\mathrm{F}}^2=\sum_{i=1}^{n}\sum_{j=1}^{m}\|X_{ij}-(\boldsymbol{WH})_{ij}\|^2. \quad (7)$$

2) The generalized Kullback-Leibler divergence between $\boldsymbol{X}$ and $\boldsymbol{WH}$, i.e.,

$$D(\boldsymbol{X}\|\boldsymbol{WH}) =\sum_{i=1}^{n}\sum_{j=1}^{m}\left(X_{ij}\log\frac{X_{ij}}{(\boldsymbol{WH})_{ij}}-X_{ij}+(\boldsymbol{WH})_{ij}\right). \quad (8)$$

Both of the two cost functions can be solved by multiplicative algorithms [13]. The update rules for the square Euclidean distance are given by

$$H_{ij}\leftarrow\frac{(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{X})_{ij}}{(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{WH})_{ij}}H_{ij},\quad W_{ij}\leftarrow\frac{(\boldsymbol{XH}^{\mathrm{T}})_{ij}}{(\boldsymbol{WHH}^{\mathrm{T}})_{ij}}W_{ij}. \quad (9)$$

And the update rules for the divergence are as follows

$$H_{ij}\leftarrow\frac{\sum_k W_{ki}X_{kj}/(\boldsymbol{WH})_{kj}}{\sum_k W_{ki}}H_{ij}, \quad (10)$$

$$W_{ij}\leftarrow\frac{\sum_k H_{jk}X_{ik}/(\boldsymbol{WH})_{ik}}{\sum_k H_{jk}}W_{ij}. \quad (11)$$

3 RNMF

3.1 Motivation

The two widely adopted cost functions of NMF (the square Euclidean distance and the generalized Kullback-Leibler divergence) are optimal for Gaussian noise and Poisson noise, respectively [14]. However, in many applications such as image processing and remote sensing, the errors in the data may be arbitrarily large. The traditional NMF will break down under this case, since the error assumptions are violated significantly.

3.2 Objective

Motivated by the recent studies in robust principal component analysis [15], we propose a novel algorithm named RNMF to handle the case with gross errors. We assume that some entries of the data matrix may be arbitrarily corrupted, but the corruption is sparse. Specifically, we introduce an error matrix $\boldsymbol{S}\in\mathbb{R}^{n\times m}$ to explicitly capture the sparse corruption. The goal of RNMF is to approximate the non-negative matrix $\boldsymbol{X}$ as

$$\boldsymbol{X}\approx\boldsymbol{WH}+\boldsymbol{S}, \quad (12)$$

where $\boldsymbol{W}\in\mathbb{R}^{n\times r}$ and $\boldsymbol{H}\in\mathbb{R}^{r\times m}$ are constrained to be non-negative, and $\boldsymbol{S}\in\mathbb{R}^{n\times m}$ is required to be sparse. Due to the presence of $\boldsymbol{S}$, $\boldsymbol{W}$ and $\boldsymbol{H}$ are protected from the corruption. Thus, the above decomposition is more robust than the traditional NMF.

The optimal $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{S}$ can be found by minimizing the approximation error. Thus, we also need to define a suitable cost function to measure the approximation error. The two cost functions of traditional NMF can be applied here. In this paper, we choose the square Euclidean distance between $\boldsymbol{X}$ and $\boldsymbol{WH}+\boldsymbol{S}$ as our cost function, due to its simplicity. Then, the objective function of RNMF is given by

$$\|\boldsymbol{X}-\boldsymbol{WH}-\boldsymbol{S}\|_{\mathrm{F}}^2=\sum_{i=1}^{n}\sum_{j=1}^{m}\|X_{ij}-(\boldsymbol{WH})_{ij}-S_{ij}\|^2. \quad (13)$$

Let $\|\cdot\|_0$ be the matrix ℓ_0-norm which counts the number of nonzero elements in its argument. To enforce the sparsity, we add an ℓ_0-norm constraint on $\boldsymbol{S}$:

$$\|\boldsymbol{S}\|_0\leqslant v, \quad (14)$$

where v is the parameter that specifies the maximum number of nonzero elements in $\boldsymbol{S}$. Finally, we obtain the following optimization problem:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{S}}\ &\|\boldsymbol{X}-\boldsymbol{WH}-\boldsymbol{S}\|_{\mathrm{F}}^2\\ \text{s.t.}\ &\boldsymbol{W}\geqslant 0,\boldsymbol{H}\geqslant 0,\|\boldsymbol{S}\|_0\leqslant v.\end{aligned} \quad (15)$$

Since the ℓ_0-norm is difficult to solve, we replace the ℓ_0-norm constraint with a ℓ_1-norm regularizer, which has been a standard technique for sparse solution. Then, the optimization problem (15) is reformulated as

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{S}}\ &\|\boldsymbol{X}-\boldsymbol{WH}-\boldsymbol{S}\|_{\mathrm{F}}^2+\lambda\|\boldsymbol{S}\|_1\\ \text{s.t.}\ &\boldsymbol{W}\geqslant 0,\boldsymbol{H}\geqslant 0,\end{aligned} \quad (16)$$

where $\|\boldsymbol{S}\|_1=\sum_{i=1}^{n}\sum_{j=1}^{m}|S_{ij}|$ and $\lambda>0$ is the regularization parameters, which controls the sparsity of $\boldsymbol{S}$.

4 Optimization

In the following, we introduce an iterative approach based on the coordinate descent to solve problem (16). The values of $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{S}$ are updated individually, while holding the other variables constant. Problem (16) is not convex in $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{S}$ jointly, but convex in them separately. Thus, a local optimal solution can be found by solving a sequence of convex optimization problems.

4.1 Optimize $\boldsymbol{H}$ and $\boldsymbol{W}$ for fixed $\boldsymbol{S}$

The optimization problems for updating $\boldsymbol{W}$ and $\boldsymbol{H}$ are in the form of non-negative quadratic programming, which can be solved by multiplicative updates [16].

First, we show how to update $\boldsymbol{H}$, given the values of $\boldsymbol{W}$ and $\boldsymbol{S}$. Let J denote the objective function in Eq. (16). For the fixed $\boldsymbol{W}$ and $\boldsymbol{S}$, the part of J that involves $\boldsymbol{H}$ is

$$\begin{aligned}&\|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_{\mathrm{F}}^2\\&=\|\boldsymbol{S}-\boldsymbol{X}+\boldsymbol{W}\boldsymbol{H}\|_{\mathrm{F}}^2\\&=\mathrm{Tr}\left[\left((\boldsymbol{S}-\boldsymbol{X})^{\mathrm{T}}+\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\right)(\boldsymbol{S}-\boldsymbol{X}+\boldsymbol{W}\boldsymbol{H})\right]\\&=\mathrm{Tr}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H})+2\mathrm{Tr}\left((\boldsymbol{S}-\boldsymbol{X})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\right)\\&\quad+\mathrm{Tr}\left((\boldsymbol{S}-\boldsymbol{X})^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X})\right).\end{aligned}\tag{17}$$

Dropping the last constant term $\mathrm{Tr}\big((\boldsymbol{S}-\boldsymbol{X})^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X})\big)$, we obtain the following convex problem for updating $\boldsymbol{H}$:

$$\begin{aligned}\min_{\boldsymbol{H}}\ &\mathrm{Tr}(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H})+2\mathrm{Tr}\left((\boldsymbol{S}-\boldsymbol{X})^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}\right)\\\text{s.t.}\ &\boldsymbol{H}\geqslant 0.\end{aligned}\tag{18}$$

Theorme 1 Consider the following non-negative quadratic programming problem:

$$\begin{aligned}\min_{\boldsymbol{v}}\ &\frac{1}{2}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{v}+\boldsymbol{b}^{\mathrm{T}}\boldsymbol{v}\\\text{s.t.}\ &\boldsymbol{v}\geqslant 0,\end{aligned}\tag{19}$$

where $\boldsymbol{A}$ is a symmetric positive semidefinite matrix. Let $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ denote the non-negative matrices with elements:

$$A_{ij}^+=\begin{cases}A_{ij},\text{if } A_{ij}\geqslant 0,\\0,\quad\text{otherwise,}\end{cases}\quad\text{and } A_{ij}^-=\begin{cases}|A_{ij}|,\ \text{if } A_{ij}<0,\\0,\quad\text{otherwise.}\end{cases}\tag{20}$$

It has been proved that the optimal solution of problem (19) can be obtained by the following multiplicative updates [16]:

$$v_i\leftarrow\left[\frac{-b_i+\sqrt{b_i^2+4(\boldsymbol{A}^+\boldsymbol{v})_i(\boldsymbol{A}^-\boldsymbol{v})_i}}{2(\boldsymbol{A}^+\boldsymbol{v})_i}\right]v_i.\tag{21}$$

The above update is guaranteed to decrease the value of the objective function at each iteration. Moreover, if the initial value of $\boldsymbol{v}$ is non-negative, its value stays non-negative in all the iterations.

Theorem 1 can be directly applied to solving our optimization problem (18). $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}$ in Eq. (18) corresponds to the matrix $\boldsymbol{A}$ in problem (19). Since the value of $\boldsymbol{W}$ stays non-negative in our optimization procedures, we have

$$\boldsymbol{A}^+=\boldsymbol{A}=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W},\ \text{and}\ \boldsymbol{A}^-=0.\tag{22}$$

Following Theorem 1, the update rule for $\boldsymbol{H}$ is given by

$$H_{ij}\leftarrow\left[\frac{\left|\left(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X})\right)_{ij}\right|-\left(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X})\right)_{ij}}{2(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H})_{ij}}\right]H_{ij}.\tag{23}$$

By reversing the roles of $\boldsymbol{H}$ and $\boldsymbol{W}$, we can derive the following update rule for $\boldsymbol{W}$:

$$W_{ij}\leftarrow\left[\frac{\left|\left((\boldsymbol{S}-\boldsymbol{X})\boldsymbol{H}^{\mathrm{T}}\right)_{ij}\right|-\left((\boldsymbol{S}-\boldsymbol{X})\boldsymbol{H}^{\mathrm{T}}\right)_{ij}}{2(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ij}}\right]W_{ij}.\tag{24}$$

4.2 Optimize $\boldsymbol{S}$ for fixed $\boldsymbol{H}$ and $\boldsymbol{W}$

The convex optimization problem for updating $\boldsymbol{S}$ can be solved efficiently via the *soft-thresholding operator* [17].

For the fixed $\boldsymbol{H}$ and $\boldsymbol{W}$, the optimization problem for updating $\boldsymbol{S}$ is

$$\min_{\boldsymbol{S}}\ \|\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}-\boldsymbol{S}\|_{\mathrm{F}}^2+\lambda\|\boldsymbol{S}\|_1.\tag{25}$$

Theorme 2 Define the *soft-thresholding operator* $T_\nu(\cdot)$ as follows:

$$T_\nu(x)=\begin{cases}x-\nu,\text{if } x>\nu,\\x+\nu,\text{if } x<-\nu,\\0,\text{otherwise,}\end{cases}\tag{26}$$

where $x\in\mathbb{R}$ and $\nu>0$. This operator can be extended to vectors and matrices by applying it element-wise. Now, consider the following ℓ_1-minimization problem:

$$\min_{\boldsymbol{v}}\ \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_{\mathrm{F}}^2+\nu\|\boldsymbol{v}\|_1.\tag{27}$$

The unique solution $\boldsymbol{v}^*$ of Eq. (27) is given by $T_\nu(\boldsymbol{x})$ [17].

According to Theorem 2, the optimal solution to problem (25) is $T_{\frac{\lambda}{2}}(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H})$. Thus, the update rule for $\boldsymbol{S}$ is

$$\boldsymbol{S}\leftarrow T_{\frac{\lambda}{2}}(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H}).\tag{28}$$

As can be seen from the above equation, if $\lambda/2>\max_{ij}(\boldsymbol{X}-\boldsymbol{W}\boldsymbol{H})_{ij}$, all the elements in $\boldsymbol{S}$ will be zero. Thus, RNMF is equivalent to NMF when λ is large enough.

4.3 Algorithm

Following Theorems 1 and 2, we know that the objective function J is nonincreasing under the three iterative updating rules described in Eqs. (23), (24) and (28). Since the objective function is bounded below, the convergence of the algorithm is guaranteed.

Note that the solution to minimizing the objective function J is not unique. If $\boldsymbol{W}$ and $\boldsymbol{H}$ are the solution to J, then, $\boldsymbol{W}\boldsymbol{D}$ and $\boldsymbol{D}^{-1}\boldsymbol{H}$ will also form a solution for any positive diagonal matrix $\boldsymbol{D}$ [8]. To remove this freedom, we further require that the norm of each column vector (the base) in $\boldsymbol{W}$ is one. Then, we compensate the norms of the bases into $\boldsymbol{H}$ to keep the value of J unaltered. The two normalization steps are as follows:

$$W_{ij} \leftarrow \frac{W_{ij}}{\sqrt{\sum_{k=1}^{n} W_{kj}^2}}, \tag{29}$$

$$H_{ij} \leftarrow H_{ij} \sqrt{\sum_{k=1}^{n} W_{ki}^2}. \tag{30}$$

The detailed algorithmic procedure is presented in Algorithm 1.

Algorithm 1 Iterative algorithm for robust NMF

Input: The $n \times m$ data matrix ($\boldsymbol{X}$), the initial values of $\boldsymbol{W}$ and $\boldsymbol{H}$ ($\boldsymbol{W}^0$ and $\boldsymbol{H}^0$), the regularization parameter (λ), the number of iterations (t)

Output: The final values of $\boldsymbol{W}$, $\boldsymbol{H}$ and $\boldsymbol{S}$

$\boldsymbol{W} \leftarrow \boldsymbol{W}^0$

$\boldsymbol{H} \leftarrow \boldsymbol{H}^0$

for $i = 1$ to t **do**

$\boldsymbol{S} \leftarrow \boldsymbol{X} - \boldsymbol{W}\boldsymbol{H}$

$$S_{ij} \leftarrow \begin{cases} S_{ij} - \dfrac{\lambda}{2}, \text{ if } S_{ij} > \dfrac{\lambda}{2}, \\ S_{ij} + \dfrac{\lambda}{2}, \text{ if } S_{ij} < -\dfrac{\lambda}{2}, \\ 0, \text{otherwise.} \end{cases}$$

$$W_{ij} \leftarrow \left[\frac{\left| \left((\boldsymbol{S}-\boldsymbol{X})\boldsymbol{H}^{\mathrm{T}} \right)_{ij} \right| - \left((\boldsymbol{S}-\boldsymbol{X})\boldsymbol{H}^{\mathrm{T}} \right)_{ij}}{2(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})_{ij}} \right] W_{ij}$$

$$H_{ij} \leftarrow \left[\frac{\left| \left(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X}) \right)_{ij} \right| - \left(\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{S}-\boldsymbol{X}) \right)_{ij}}{2(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H})_{ij}} \right] H_{ij}$$

$$W_{ij} \leftarrow \frac{W_{ij}}{\sqrt{\sum_{k=1}^{n} W_{kj}^2}}$$

$$H_{ij} \leftarrow H_{ij} \sqrt{\sum_{k=1}^{n} W_{ki}^2}$$

end for

5 Experiments

In this section, we evaluate the performance of our proposed RNMF algorithm for face clustering and face recognition. The following three dimensionality reduction methods are compared:

1) PCA [4]
2) LSI [5]
3) RNMF

Besides, we also provide the results of the Baseline method, which uses the original feature without dimensionality reduction. Note that the iterative procedure for solving RNMF can only find local minimum, and is sensitive to the initial values of $\boldsymbol{W}$ and $\boldsymbol{H}$. In the experiments, we run Algorithm 1 ten times with different start values and the best result in terms of the objective function of RNMF is recorded. Because RNMF is equivalent to the traditional NMF, when λ is large enough, we do not show the result of the traditional NMF explicitly, since it can be inferred from that of RNMF.

5.1 Data sets

Two face images databases are used in the experiments: the Yale face database and the AR face database.

The Yale face database[1)] contains 165 gray scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. All the face images are manually aligned and cropped. The size of each cropped image is 64×64 pixels, with 256 gray levels per pixel. Thus, each image is represented as a 4096-dimensional vector. To simulate the case that the images are grossly corrupted, we randomly select 30 percent of the images in the Yale face database for corruption. These images are corrupted by superimposing one 8×8 white block on them. Figure 1(a) shows some sample images from the corrupted Yale face database.

The AR face database[2)] was created by Aleix Martinez and Robert Benavente in the Computer Vision Center (CVC) at the U.A.B [18]. It consists of over 3200 color images corresponding to 126 people's faces. There are 26 different images for each subject. Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sun glasses and scarf). In this paper, we use the morphed images provided by Ref. [19]. These color images are resized to 47×64 and converted to gray-level images. This way, each image is represented as a 3008-dimensional vector. Figure 2 shows some sample images from the AR face database.

In the experiments, we pre-process the face images by scaling features (pixel values) to [0,1] (divided by 255).

5.2 Case study

It is very interesting to see the decomposition result of RNMF on the face database. We take the corrupted Yale face database as an example. RNMF is applied to reducing the dimensionality from 4096 to 15. That is, the 4096×165 data matrix $\boldsymbol{X}$ is decomposed as

$$\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H} + \boldsymbol{S}, \tag{31}$$

where $\boldsymbol{W} \in \mathbb{R}^{4096\times 15}$, $\boldsymbol{H} \in \mathbb{R}^{15\times 165}$, and $\boldsymbol{S} \in \mathbb{R}^{4096\times 165}$. Let $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_{165}]$ and $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_{165}]$. Then, the decomposition can be rewritten column by column as

$$\boldsymbol{x}_i \approx \boldsymbol{W}\boldsymbol{h}_i + \boldsymbol{s}_i, \ i = 1, 2, \ldots, m. \tag{32}$$

1) http://cvc.yale.edu/projects/yalefaces/yalefaces.html
2) http://www2.ece.ohio-state.edu/ aleix/ARdatabase.html

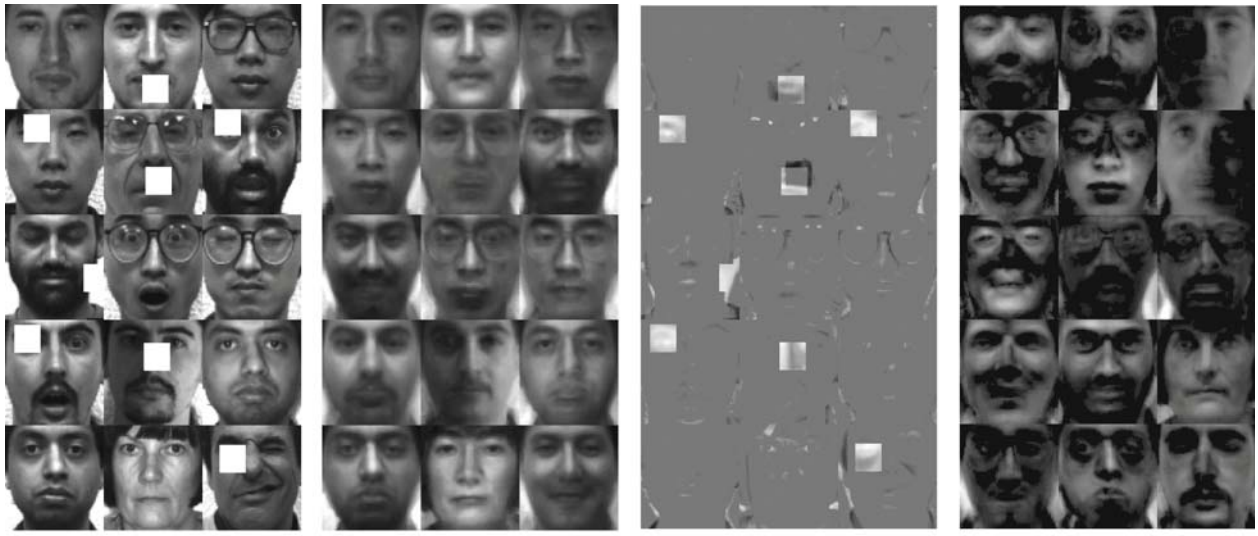


Fig. 1 Decomposition result of RNMF on the corrupted Yale face database. We apply RNMF to reducing the dimensionality of the faces from 4096 to 15, and the parameter λ is set to 0.3. (a) 15 sample images from the corrupted Yale face database; (b) reconstructed faces for the 15 sample images; (c) error vectors for the 15 sample images; (d) 15 basis vectors contained in $\boldsymbol{W}$

Fig. 2 Sample images from the AR face database

In the following, we refer to $\boldsymbol{W}\boldsymbol{h}_i$ as the reconstructed face, and $\mathbf{s}_i$ as the error vector. Figure 1 plots the decomposition result of RNMF with the parameter $\lambda = 0.3$. Because of limited space, we just show 15 sample faces (Fig. 1(a)), their corresponding reconstructed faces (Fig. 1(b)), their corresponding error vectors (Fig. 1(c)), and the 15 basis vectors contained in $\boldsymbol{W}$ (Fig. 1(d)). As can be seen, the faces reconstructed by RNMF are quite clear, and we can hardly find the white blocks. Furthermore, the man-made corruption is indeed captured by the error vectors.

5.3 Face clustering

In this subsection, we show the experimental results of face clustering.

5.3.1 Evaluation metrics

Two metrics, the accuracy (AC) and the normalized mutual information ($\overline{\text{MI}}$), are used to measure the clustering performance [8]. Given a face image $\boldsymbol{x}_i$, let p_i and q_i be the obtained cluster label and the label provided by the database, respectively. The AC is defined as follows:

$$\text{AC} = \frac{\sum_{i=1}^{m} \delta(q_i, \text{map}(p_i))}{m}, \tag{33}$$

where m is the total number of face images, $\delta(a,b)$ is the delta function that equals one if $a = b$ and equals zero otherwise, and $\text{map}(p_i)$ is the permutation mapping function that map each cluster label p_i to the equivalent label from the database. The best mapping can be found by using the Kuhn-Munkres algorithm [20].

Let C denote the set of clusters provided by the database and C' obtained from the clustering algorithm. Their mutual information metric $\text{MI}(C, C')$ is defined as follows:

$$\text{MI}(C, C') = \sum_{c_i \in C,\, c'_j \in C'} p(c_i, c'_j) \log_2 \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)}, \tag{34}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a face image arbitrarily selected from the database belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected face image belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{\text{MI}}$ as follows:

$$\overline{\text{MI}} = \frac{\text{MI}(C, C')}{\max(H(C), H(C'))} \tag{35}$$

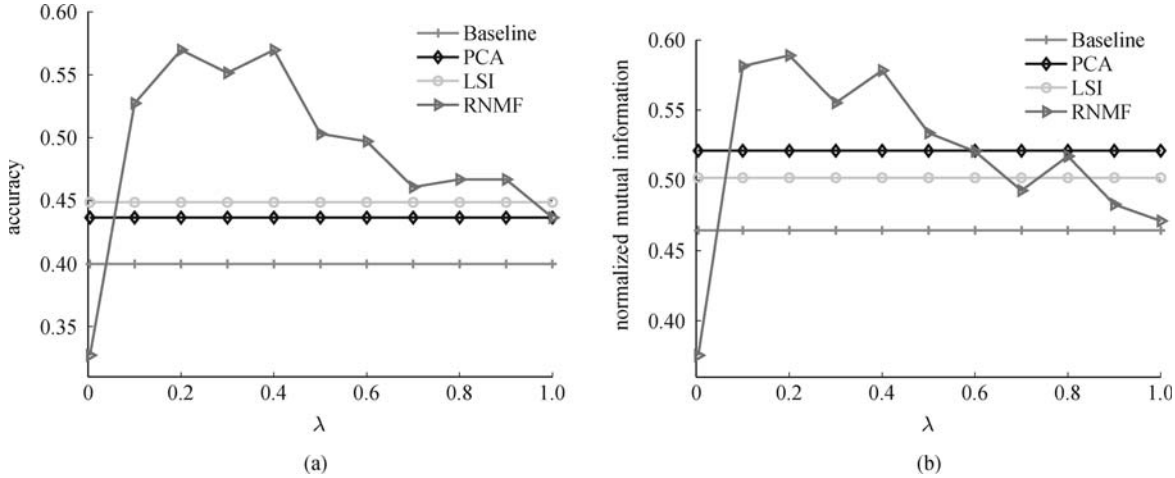


Fig. 3 Clustering results on the corrupted Yale face database. Each dimensionality reduction algorithm is applied to mapping the face images to a 15-dimensional subspace. Then, k-means is applied to partitioning these faces into 15 clusters. (a) Accuracy versus value of λ; (b) normalized mutual information versus the value of λ

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively.

5.3.2 Experimental results

The clustering experiments are performed on the corrupted Yale database. First, all the face images are mapped to a 15-dimensional subspace by each dimensionality reduction algorithm (PCA, LSI, and RNMF). Second, we use k-means to partition these faces into 15 clusters. The result of k-means in the original feature space is referred to as Baseline since the k-means algorithm can only find local minimum. In our experiments, we apply it 50 times with different start points and the best result in terms of its objective function is recorded.

For our RNMF, there is a regularization parameter λ, which controls the sparsity of $\boldsymbol{S}$. In Fig. 3, we show the clustering performance of RNMF versus the value of λ. As can be seen, RNMF can achieve significantly better performance than other methods over a large range of λ (0.1 to 0.6). As λ increases, the elements in $\boldsymbol{S}$ tend to be zero more often than not, and hence our RNMF actually converges to NMF. When λ is large enough, all the elements in $\boldsymbol{S}$ will be zero. Then, the performance of RNMF does not depend on λ any more. The performance of PCA and SVD is similar, and better than the Baseline.

5.4 Face recognition

We choose the first 10 subjects of the AR face database to form an image subset, which contains 260 face images. The face recognition experiments are performed on this subset. As before, we reduce the dimensionality of these images from 3008 to 15 using each dimensionality reduction algorithm, and do face recognition in the reduced subspace. Recognition in the original 3008-dimensional space is referred to as Baseline.

We select 3 images per subject as the training data, and the rest are used for testing. The 1-nearest neighbor (1-nn) classifier is used to classify the testing data. 50 training/testing splits are randomly generated and the average classification accuracy over these splits is used to evaluate the face recognition performance. Figure 4 plots the average classification accuracy versus the value of λ. The classification accuracy of RNMF is much better than the compared methods. On this database, the performance of RNMF levels off at $\lambda = 2$. Note that for face recognition, the performance of PCA and LSI is even worse than the Baseline.

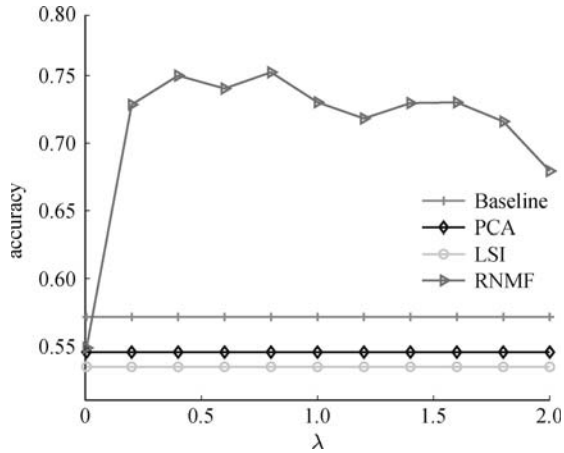


Fig. 4 Average classification accuracy versus value of λ on the AR face database

6 Conclusions

In this paper, we propose a novel dimensionality reduction method named Robust Non-negative Matrix Factorization (RNMF). RNMF allows some entries of the data matrix to be grossly corrupted, but the corruption need to be sparse. Experimental results on two standard face

databases show that RNMF can significantly improve the performance of face clustering and recognition.

There is still one open question that needs to be addressed. That is, under what conditions can RNMF recover the optimal $\boldsymbol{W}$ and $\boldsymbol{H}$ from the corrupted data matrix? We will investigate this in the future.

Acknowledgements This work was supported by the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education, and the National Natural Science Foundation of China (Grant No. 60875044).

References

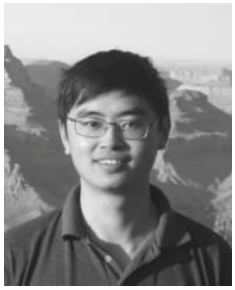
1. Duda R O, Hart P E, Stork D G. Pattern Classification. New York: Wiley-Interscience Publication, 2000
2. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Springer Series in Statistics. New York: Springer, 2009
3. Fodor I K. A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Technical report, 2002
4. Bishop C M. Pattern Recognition and Machine Learning (Information Science and Statistics). New York: Springer, 2007
5. Deerwester S, Dumais S T, Furnas G W, Landauer T K, Harshman R. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 1990, 41: 391–407
6. Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization. Nature, 1999, 401(6755): 788–791
7. Kalman D. A singularly valuable decomposition: the svd of a matrix. The College Mathematics Journal, 1996, 27(1): 2–23
8. Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR Conference on Research and development in informaion retrieval. 2003, 267–273
9. Cai D, He X, Wu X, Han J. Non-negative matrix factorization on manifold. In: Proceedings of the 8th IEEE International Conference on Data Mining. 2008, 63–72
10. Guillamet D, Vitrià J. Non-negative matrix factorization for face recognition. In: Escrig M, Toledo F, Golobardes E, eds. Topics in Artificial Intelligence. Lecture Notes in Computer Science, 2002, 2504: 336–344
11. Brunet J P, Tamayo P, Golub T R, Mesirov J P. Metagenes and molecular pattern discovery using matrix factorization. Proceedings of the National Academy of Sciences of the United States of America, 2004, 101(12): 4164–4169
12. Carmona-Saez P, Pascual-Marqui R D, Tirado F, Carazo J, Pascual-Montano A. Biclustering of gene expression data by non-smooth non-negative matrix factorization. BMC Bioinformatics, 2006, 7(1): 78
13. Lee D D, Seung H S. Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems, 2001, 13(2): 556–562
14. Cichocki A, Zdunek R, Amari S I. Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In: Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation. Lecture Notes in Computer Science. Charleston: Springer, 2006, 3889: 32–39
15. Wright J, Ganesh A, Rao S, Peng Y, Ma Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. Advances in Neural Information Processing Systems, 2009, 22: 2080–2088
16. Sha F, Lin Y, Saul L K, Lee D D. Multiplicative updates for nonnegative quadratic programming. Neural Computation, 2007, 19(8): 2004–2031
17. Hale E T, Yin W, Zhang Y. Fixed-point continuation for ℓ_1-minimization: methodology and convergence. SIAM Journal on Optimization, 2008, 19(3): 1107–1130
18. Martínez A M, Benavente R. The ar face database. Computer Vision Center. Technical Report 24, 1998
19. Martínez A M, Kak A C. Pca versus lda. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2): 228 –233
20. Lováz L, Plummern M D. Matching Theory. Amsterdam: North-Holland, 1986

Lijun ZHANG received the BS degree in Computer Science from Zhejiang University, China, in 2007. He is currently a candidate for a Ph.D. degree in Computer Science at Zhejiang University. His research interests include machine learning, information retrieval, and data mining.

Zhengguang CHEN received the BS degree in Computer Science from Zhejiang University, China, in 2009. He is currently a candidate for a MS degree in Computer Science at Zhejiang University. His research interests include computer vision, machine learning, and data mining.

Miao ZHENG received the BS degree in Computer Science from Zhejiang University, China, in 2008. He is currently a candidate for a Ph.D. degree in Computer Science at Zhejiang University. His research interests include machine learning, informa-

tion retrieval, and data mining.



Xiaofei HE received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University in 2007, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of IEEE.