2012 IEEE 12th International Conference on Data Mining

Robust Nonnegative Matrix Factorization Via Half-Quadratic Minimization

Liang Du*†, Xuan Li*†, Yi-Dong Shen*
*State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
†Graduate University, Chinese Academy of Sciences, Beijing 100049, China
{duliang,lixuan,ydshen}@ios.ac.cn

***Abstract*—Nonnegative matrix factorization (NMF) is a popular technique for learning parts-based representation and data clustering. It usually uses the squared residuals to quantify the quality of factorization, which is optimal specifically to zero-mean, Gaussian noise and sensitive to outliers in general cases. In this paper, we propose a robust NMF method based on the correntropy induced metric, which is much more insensitive to outliers. A half-quadratic optimization algorithm is developed to solve the proposed problem efficiently. The proposed method is further extended to handle outlier rows by incorporating structural knowledge about the outliers. Experimental results on data sets with and without apparent outliers demonstrate the effectiveness of the proposed algorithms.**

***Keywords*-robust non-negative matrix factorization, half-quadratic optimization, correntropy induced metric**

I. Introduction

In many applications in data mining and machine learning, one if often confronted with high dimensional data. Matrix factorization, which usually seeks two or more lower dimensional matrices to approximate the original data, is popular for data analysis. The factorization results in a reduced representation of the original data. It can be seen either as a feature extraction or a dimensionality reduction technique. The popular matrix factorization methods include Principle Component Analysis (PCA), Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) [1].

Unlike PCA and SVD, NMF aims to find two non-negative matrices whose product provides a good approximation to the original matrix. Due to the non-negativity constraints, each data is represented as additive combination of basis factors. As a result, NMF can be interpreted as a part-based representation of the data. Recently, many variants of NMF with different forms of factorization and regularization have also been developed to improve original NMF from different perspectives, including [2], [3], [4], [5], [6]. The NMF technique has been applied to many applications in the fields of DNA gene expression analysis [7], object recognition [8], and clustering [9].

The original NMF and its many variants usually use the sum of squared error or the L_2 error function to measure the quality of approximation. Although it has nice mathematical properties and shown their effectiveness in many tasks, it is not always the best choice for data analysis [10]. It has been shown that the squared error is optimal for zero mean, Gaussian noise [11]. Actually, real world problems almost always involve data that do not conform to the assumptions made by the model. Previous studies show that the least-squares error measure is sensitive to outliers. Sometimes, even a single corrupted point can arbitrarily degrade the quality of the approximation. Recently, some variants have been proposed to improve the robustness of original NMF. The $L_1 - L_2$ function in [12] was used for the purpose of robust factorization. The optimization problem is solved by a general gradient descent scheme, which is computational expensive. $L_{2,1}$-NMF [11] was proposed to measure the quality of decomposition using the $L_{2,1}$-norm function, which assumes the fitting error follows Laplacian distribution. SR-NMF [13], [14] was proposed to perform standard NMF by subtracting an outlier matrix, which is assumed to be sparse.

In this paper, we propose to replace the quadratic form of residuals by less increasing functions to achieve robust factorization. Instead of directly minimizing the non-quadratic and possibly non-convex loss function, we develop an iterative algorithm relying on the half-quadratic minimization technique. At each iteration, the optimization problem is reduced to a weighted least square NMF, which can be solved in a similar way to standard NMF. In particular, we first propose a novel robust NMF method based on the correntropy induced metric, called CIM-NMF. The correntropy has been shown to obtain robust analysis in information theoretic learning and effectively handle non-Gaussian noise and large outliers [15]. We then extend CIM-NMF to handle outlier rows by incorporating structural knowledge about the outliers, which leads to rCIM-NMF. Due to the connection between the correntropy induced metric and the robust M-estimators, we further extend to use the Huber's function to measure the quality of approximation and have Huber-NMF. The optimization problems for CIM-NMF, rCIM-NMF, and Huber-NMF can be efficiently solved by the iterative algorithm. We also discuss the connections between our methods and existing robust NMFs and weighted NMFs. Experimental results on benchmark datasets with/without outliers demonstrate the effectiveness of the proposed methods.

The rest of the paper is organized as follows: in Section II, we give a brief overview of NMF. In Section III, we

1550-4786/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDM.2012.39

introduce the generic learning algorithm based on half quadratic minimization. In section IV, we introduce three robust variants of NMF, that is, CIM-NMF, rCIM-NMF, and Huber-NMF. In Section V, we discuss the relations to existing works. In Section VI, we present the experimental results. Finally, in Section VII, we conclude the paper with future works.

II. A Brief Review of NMF

Given a non-negative data matrix $X \in \mathbb{R}^{N\times M}$, whose rows correspond to data instances and columns to features. We use X_{i*} to denote the ith row and X_{*j} to denote the jth column in X. NMF aims to find two nonnegative matrices $U \in \mathbb{R}^{N\times K}$ and $V \in \mathbb{R}^{M\times K}$, whose product can well approximate the original matrix X:

$$X \approx UV^T. \tag{1}$$

There are different criteria to quantify the quality of the decomposition. Lee et al. [1], [16] proposed two objective functions: the square of the Euclidean distance and the Kullback-Leibler divergence. The standard NMF can be formulated as minimizing the following sum of squared residuals:

$$\min_{U,V} \sum_{i=1}^{N} ||X_{i*} - U_{i*}V^T||^2 = \sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - (UV)_{ij})^2 \tag{2}$$
$$\text{s.t.} \quad U \geq 0, V \geq 0,$$

Although the objective function in Eq. (2) is convex with respect to U only or V only, it is not convex in both variables together. It is proved that a local minimum can be found by the following iterative multiplicative update rules:

$$U_{ik} = U_{ik}\frac{(XV)_{ik}}{(UV^TV)_{ik}} \tag{3}$$

$$V_{jk} = V_{jk}\frac{(X^TU)_{jk}}{(VU^TU)_{jk}}. \tag{4}$$

III. A Generic Algorithm based on Half-Quadratic Minimization

In this paper, we will derive several robust variants of NMF by replacing the squared error with other functions. Here, we first introduce the generic robust NMF framework based on Half-Quadratic minimization technique. We will use this results repeatedly later.

A. The Half-Quadratic Minimization

Let the residual e be the difference between the actual value of the data and the value predicted by NMF model. Replacing the squared residual on each entry[1] with a generic function, which yields

$$\mathcal{J}(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{M}\ell(E_{ij}), \tag{5}$$

where $E_{ij} = X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk}$, and $\ell(\cdot)$ is chosen to be robust to outliers or gross errors.

Generally, the loss function is non-quadratic and possibly non-convex, and it is difficult to be minimized directly. Fortunately, the half-quadratic minimization technique has been developed to optimize those loss functions. By introducing additional auxiliary variable, it reformulates a non-quadratic loss function as an augmented objective function in an enlarged parameter space. It has been shown that the half-quadratic iterations is a quasi-Newton method and substantially faster than gradient based methods [17].

According to the conjugate function [18] and half-quadratic theory [17], for a fixed E_{ij}, the following equations holds

$$\ell(E_{ij}) = \min_{W_{ij}\in\mathbb{R}} Q(E_{ij}, W_{ij}) + \phi(W_{ij}), \tag{6}$$

where $\phi(W_{ij})$ is the conjugate function of $\ell(E_{ij})$, W_{ij} is the corresponded auxiliary variable, and $Q(\cdot,\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is a quadratic term for E_{ij} and W_{ij}. In this paper, we only consider the quadratic term of multiplicative form [19]

$$Q(E_{ij}, W_{ij}) = \frac{1}{2}W_{ij}E_{ij}^2. \tag{7}$$

Substituting Eq. (6) and Eq. (7) into Eq. (5), we have the augmented objective function

$$\min_{U,V}\left\{\mathcal{J}(U,V) = \sum_{i=1}^{n}\ell(E_{ij})\right\}$$
$$= \min_{U,V,W}\left\{\mathcal{J}(U,V,W) = \sum_{i=1}^{n}[\frac{1}{2}W_{ij}E_{ij}^2 + \phi(W_{ij})]\right\} \tag{8}$$

The loss function in Eq. (8) can be optimized by the following alternating minimization scheme:

- When U and V are fixed, the minimization of the objective function in Eq. (8) becomes convex with respect to W. The explicit optimum is given by
$$W_{ij} = \frac{\ell'(E_{ij})}{E_{ij}}, \tag{9}$$
which only depends on the loss function $\ell(\cdot)$. It is expected that outliers often cause large fitting errors and the corresponding weights W_{ij} should be small. For inliers with small errors, the weights W_{ij} should be large. Therefore, W_{ij} can be seen as an outlier mask. Some examples of these weight functions can be found in Figure 1(b).
- When W is fixed, the minimization of the objective function in Eq. (8) reduces to the weighted NMF

[1] Note that the residual e can also be defined on rows or columns instead of the entries, which is the case of the rCIM-NMF (see Section IV-C).

presented in Eq. (12). A local minimum can be found by the following update rules

$$U_{ik} = U_{ik}\frac{(W \otimes XV)_{ik}}{(W \otimes (UV^T)V)_{ik}} \tag{10}$$

$$V_{jk} = V_{jk}\frac{((W \otimes X)^T U)_{jk}}{((W \otimes (UV^T))^T U)_{jk}}. \tag{11}$$

More details about weighted NMF can be found in next subsection.

The update sequence generated by the above scheme will converges. The objective function in Eq. (8) is nonincreasing under the update rules in Eq. (9) and Eq. (10). It can be verified that the objective function is also bounded. A similar proof of the convergence of weighted NMF can be found in [20], [11].

B. Weighted NMF

By assigning a non-negative weight on each entry, the weighted NMF with squared error can be formulated as the following optimization problem [21], [22]

$$\begin{aligned} \min_{U,V} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M} W_{ij}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2 \\ \text{s.t.} \quad & U \geq 0, V \geq 0, \end{aligned} \tag{12}$$

where $W \in \mathbb{R}^{N\times M}$ is a non-negative matrix, which indicates the importance of entries in X.

Since the objective function in Eq. (12) is not convex with U and V jointly, we aim to find a local minimum by iteratively updating U and V in a similar way with the unweighted NMF in Eq. (2).

Computation of U: Given V, the optimization problem with respect to U is equivalent to minimizing

$$\mathcal{L}_U(U) = \sum_{i=1}^{N}(X_{i*} - U_{i*}V^T)A_i(X_{i*} - U_{i*}V^T)^T + \text{tr}(\Phi U),$$

where $A_i = \text{diag}(W_{i*}) \in \mathbb{R}^{M\times M}$, $\Phi = [\Phi_{ik}]$ is the Langrange multiplier for the nonnegative constraint $U_{ik} \geq 0$. The partial derivatives of $\mathcal{L}_U$ with respect to U_{ik} is:

$$\frac{\partial \mathcal{L}_U}{\partial U_{ik}} = -2(X_{i*}A_iV)_k + 2(U_{i*}V^TA_iV)_k + \Phi_{ik}. \tag{13}$$

Using the KKT conditions $\Phi_{ik}U_{ik} = 0$, we get the following equations

$$[-(X_{i*}A_iV)_k + (U_{i*}V^TA_iV)_k]U_{ik} = 0. \tag{14}$$

The above equation leads to the following update rule:

$$\begin{aligned} U_{ik} &= U_{ik}\frac{(X_{i*}A_iV)_k}{(U_{i*}V^TA_iV)_k} & (15)\\ &= U_{ik}\frac{(W \otimes XV)_{ik}}{(W \otimes (UV^T)V)_{ik}}, & (16) \end{aligned}$$

where $\otimes$ is the Hadamard product, i.e., element wise product between two matrices. Here we assume Hadamard product has higher operator precedence over regular matrix product, i.e., $AB \otimes CD = A(B \otimes C)D$.

Computation of V: Given U, the optimization problem with respect to V is equivalent to minimizing

$$\mathcal{L}_V(V) = \sum_{j=1}^{M}(X_{*j} - UV_{j*}^T)^T B_j(X_{*j} - UV_{j*}^T) + \text{tr}(\Psi V),$$

where $B_j = \text{diag}(W_{*j}) \in \mathbb{R}^{N\times N}$. Similarly, the optimal solution for V is given by the following updating rule

$$\begin{aligned} V_{jk} &= V_{jk}\frac{(X_{*j}B_jU)_k}{(V_{j*}U^TB_jU)_k} & (17)\\ &= V_{jk}\frac{((W \otimes X)^TU)_{jk}}{((W \otimes (UV^T))^TU)_{jk}}. & (18) \end{aligned}$$

IV. Robust Nonnegative Matrix Factorization

In this section, we will derive three robust NMFs, which use the Correntropy Induced Metric or the Huber M-estimator to measure the quality of matrix approximation.

A. Correntropy Induced Metric

Recently, the concept of correntropy [15] was proposed in information-theoretic learning (ITL) to process non-Gaussian and impulsive noise. Based on the information potential of Renyi's quadratic entropy [23], the correntropy is defined as a generalized similarity between two arbitrary variables $\boldsymbol{x}$ and $\boldsymbol{y}$

$$V_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \text{Expectation}[k_\sigma(\boldsymbol{x} - \boldsymbol{y})], \tag{19}$$

where $k_\sigma(\cdot)$ is the kernel function. In practice, the joint probability density function is often unknown, and only a finite number of data $\{(x_i, y_i)\}_{i=1}^n$ are available, which leads to the following sample estimator of correntropy:

$$\hat{V}_\sigma(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} k_\sigma(x_i - y_i). \tag{20}$$

Based on the above correntropy, Liu et al. [15] further proposed the Correntropy Induced Metric (CIM) for any two vectors in the sample space as follows:

$$\text{CIM}(\boldsymbol{x}, \boldsymbol{y}) = (k(0) - \frac{1}{n}\sum_{i=1}^{n} k_\sigma(e_i))^{1/2}, \tag{21}$$

where e_i is defined as $e_i = x_i - y_i$, and we only consider the Gaussian kernel $g(e, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-e^2/2\sigma^2)$ in this paper.

Unlike the objective function in Eq. (2), which increases quadratically with the fitting error and amplifies the side effects of large errors, the value of CIM in Eq. (21) for large error is close to 1. Since the large errors are often induced by outliers, the effect of these outliers in CIM is limited and even insignificant. In other words, CIM is mainly determined by small errors, which correspond to inliers. Thus, CIM is very useful for cases when the measurement error is nonzero mean, non-Gaussian with large outliers. The CIM has shown its superiority in terms of robustness in signal processing [15], feature extraction [24], and face recognition [25]. Besides, due to the connection between correntropy and M-estimators [15], it is also practical to choose an appropriate kernel size.

B. CIM-NMF

Substituting the squared error on each entry in Eq. (2) with the squared CIM, we have the CIM-NMF by minimizing the following objective function:

$$\mathcal{J}(U,V) = 1 - \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{N} g(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk}, \sigma),$$

which is equivalent to solve the following optimization problem

$$\begin{aligned} \min_{U,V} \quad & \sum_{i=1}^{N}\sum_{j=1}^{M}(1 - g(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk}, \sigma)) \qquad (22) \\ \text{s.t.} \quad & U \geq 0, V \geq 0. \end{aligned}$$

According to Eq. (8), the above optimization problem of CIM-NMF is equivalent to minimizing the following augmented objective function in an enlarged parameter space

$$\min_{U,V,W} \sum_{i=1}^{N}\sum_{j=1}^{M}[W_{ij}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2 + \phi(W_{ij})], \quad (23)$$

which can be solved by the generic algorithm in Section 3. Concretely speaking, we optimize the objective function in Eq. (23) with respect to one variable while fixing the other variables. This procedure repeats until convergence.

Computation of W: When U and V are fixed, the optimization problem with respect to W can be solved separately, and the optimal value of W_{ij} is given by

$$\begin{aligned} W_{ij} &= \frac{\frac{d}{dE_{ij}}(1 - g(E_{ij}, \sigma))}{E_{ij}} \\ &\propto \exp(-\frac{(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2}{2\sigma^2}). \qquad (24) \end{aligned}$$

Computation of U and V: When W is given, the optimization problem in Eq. (23) is reduced to the weighted NMF in Eq. (12). Therefore, the multiplicative update rules in Eq. (15) and Eq. (17) can be directly applied.

Like any kernel method, the selection of kernel size will affect the performance of the proposed algorithm, and kernel size is often determined empirically. In this paper, the kernel size is computed as average reconstruction error, i.e.,

$$\sigma^2 = \frac{1}{2NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2. \quad (25)$$

The complete algorithm to solve CIM-NMF is summarized in algorithm 1.

Algorithm 1 CIM-NMF Algorithm Description

```
Input: The data matrix $X \in \mathbb{R}^{N \times M}$, the initial values of
  $U \in \mathbb{R}^{N \times K}$ and $V \in \mathbb{R}^{M \times K}$.
Output: $U$, $V$, and weight matrix $W$.
  repeat
    Update $W$ by $W_{ij} = \exp(-\frac{(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2}{2\sigma^2})$;
    Update $U$ by $U_{ik} = U_{ik}\frac{(W \otimes XV)_{ik}}{(W \otimes (UV^T)V)_{ik}}$;
    Update $V$ by $V_{jk} = V_{jk}\frac{((W \otimes X)^T U)_{jk}}{((W \otimes (UV^T))^T U)_{jk}}$;
    Update $\sigma^2$ by $\frac{1}{2NM}\sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2$;
  until Converges
```

C. rCIM-NMF

In many applications, we have additional knowledge on outliers. For example, in a microarray data set, if one record (row) is corrupted, then it is likely that the qualities of most of the entries in its corresponding row are low and it should be better to consider the entire row as outlier. To find such outlier pattern and decrease their contribution to the optimization problem automatically, we can measure the quality of matrix factorization by considering all entries in one row as a whole. In this way, rows of the data matrix are assigned to different weights, and entries in the same row are equally weighted.

Concretely, we substitute the squared residuals on each row (left equation in Eq. (2)) with the squared CIM, and get the row-based CIM-NMF (rCIM-NMF) by minimizing the following objective function:

$$J(U,V) = 1 - \frac{1}{N}\sum_{i=1}^{N} g(||X_{i*} - U_{i*}V^T||, \sigma), \quad (26)$$

which is equivalent to the following optimization problem

$$\begin{aligned} \min_{U,V} \quad & \sum_{i=1}^{N}(1 - g(||X_{i*} - U_{i*}V^T||, \sigma)) \qquad (27) \\ \text{s.t.} \quad & U \geq 0, V \geq 0. \end{aligned}$$

Similar to CIM-NMF in Eq. (22), we also solve the optimization problem via the half-quadratic minimization framework in Section 3. According to Eq. (8), the optimization problem in Eq. (27) is equivalent to minimizing the following objective function

$$\min_{U,V,\boldsymbol{w}} \quad \sum_{i=1}^{N}[w_i||X_{i*} - U_{i*}V^T||^2 + \phi(w_i)], \quad (28)$$

where w_i is the weight associated on row X_{i*}.

Computation of $\boldsymbol{w}$: Using the generic algorithm of Section 3, the optimal value of w_i is given by

$$w_i = \exp(-\frac{||X_{i*} - U_{i*}V^T||^2}{2\sigma^2}). \quad (29)$$

Computation of U and V: When $\boldsymbol{w}$ is given, the optimization problem with respect to U and V becomes the weighted NMF problem

$$\min_{U,V} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} W_{ij}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2, \tag{30}$$

where $W = \boldsymbol{w}\mathbf{1}_M^T$. Thus, the optimal value of U and V can also be obtained by the update rules in Eq. (15) and Eq. (17).

The kernel size for rCIM-NMF can also be estimated in a similar way to Eq. (25). The complete algorithm to solve rCIM-NMF is summarized in algorithm 2.

Algorithm 2 rCIM-NMF Algorithm Description

```
Input: The data matrix $X \in \mathbb{R}^{N\times M}$, the initial values of
    $U \in \mathbb{R}^{N\times K}$ and $V \in \mathbb{R}^{M\times K}$.
Output: $U$, $V$, and weight vector $\boldsymbol{w}$.
  repeat
    Update $w_i$ by $w_i = \exp(-\frac{||X_{i*}-U_{i*}V^T||^2}{2\sigma^2}), W = \boldsymbol{w}\mathbf{1}_M^T$;
    Update $U$ by $U_{ik} = U_{ik}\frac{(W\otimes XV)_{ik}}{(W\otimes(UV^T)V)_{ik}}$;
    Update $V$ by $V_{jk} = V_{jk}\frac{((W\otimes X)^T U)_{jk}}{((W\otimes(UV^T))^T U)_{jk}}$;
    Update $\sigma^2$ by $\frac{1}{2N}\sum_{i=1}^{N} ||X_{i*} - U_{i*}V^T||^2$;
  until Converges
```

D. Extensions to M-estimators and Huber-NMF

In robust statistics, M-estimators [26] are generalized maximum likelihood estimation to the minimization of the sums of functions of the data. They have been widely used in machine learning and data minging for robust learning [25]. In robust regression, IRLS is often used to solve M-estimators. Another common used technique is the half-quadratic optimization [19]. By the multiplicative and the additive half-quadratic reformulation of M-estimator, the original problem is solved by the alternate minimization of an augmented objective function. Some popular M-estimators [27] include L_p function, $L_1 - L_2$ function, Huber's function, Cauchy's function, and Welsh function.

It seems difficult to select a proper M-estimator for general purpose. In this paper, we also take the Huber function in Eq. (31) to measure the quality of approximation by considering its connection to L_2-norm and L_1-norm.

$$\ell_{\text{huber}}(e) = \begin{cases} e^2 & \text{if } |e| \le c \\ 2c|e| - c^2 & \text{if } |e| \ge c \end{cases}, \tag{31}$$

where c is the cutoff parameter to tradeoff between L_2-norm and L_1-norm.

The Huber-NMF can be formulated as the following optimization problem:

$$\min_{U,V} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} \ell_{\text{huber}}(E_{ij}), \tag{32}$$

where $E_{ij} = X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk}$.

Again, the Huber-NMF can be solved by the generic algorithm in Section 3.

Computation of W: When U and V is given, the optimal value of W is given by

$$W_{ij} = \begin{cases} 1 & \text{if } |E_{ij}| \le c \\ \frac{c}{|E_{ij}|} & \text{otherwise} \end{cases} \tag{33}$$

Computation of U and V: Similarly, the minimization of Eq. (32) can also be achieved by the equations in Eq. (15) and Eq. (17).

As [28], the cutoff parameter c is set to the median of reconstruction errors, i.e.,

$$c = \text{median}(|E_{ij}|). \tag{34}$$

The complete algorithm to solve Huber-NMF is summarized in algorithm 3.

Algorithm 3 Huber-NMF Algorithm Description

```
Input: The data matrix $X \in \mathbb{R}^{N\times M}$, the initial values of
    $U \in \mathbb{R}^{N\times K}$ and $V \in \mathbb{R}^{M\times K}$.
Output: $U$, $V$, and weight matrix $W$.
  repeat
    Update $w_i$ by $W_{ij} = \begin{cases} 1 & \text{if } |E_{ij}| \le c \\ \frac{c}{|E_{ij}|} & \text{otherwise} \end{cases}$;
    Update $U$ by $U_{ik} = U_{ik}\frac{(W\otimes XV)_{ik}}{(W\otimes(UV^T)V)_{ik}}$;
    Update $V$ by $V_{jk} = V_{jk}\frac{((W\otimes X)^T U)_{jk}}{((W\otimes(UV^T))^T U)_{jk}}$;
    Update $c$ by $\text{median}(|E_{ij}|)$;
  until Converges
```

E. Discussion

To get a better understanding to the behaviors of these functions, we plot these loss functions and their corresponding weight functions of multiplicative half-quadratic form in Figure 1. The interesting observation is as follows: 1) compared with L_2-norm, other functions are less increasing and give less punishment to large fitting errors; 2) though L_1 norm is often used to pursue robustness, the corresponding weight $1/|e|$ is not upper bounded, so the objective function would be dominated by the data points with near-zero fitting errors which leads to the singularity problem [28]; 3) the Welsch function, which is equivalent to CIM in Eq. (22), behaves like an L_2 norm on small errors, like an L_1 norm on relative larger errors, and approaching L_0 norm with the further increase of errors. The weight is upper bounded by 1 for small error and lower bounded by 0 for large error. The different property is determined by a scale parameter. 4) the Huber function behaves like an L_2 norm on small errors and like L_1 norm on large errors, controlled by a cutoff parameter; 5) the Cauchy function is also insensitive with large errors, which has been used for robust embedding learning [29].

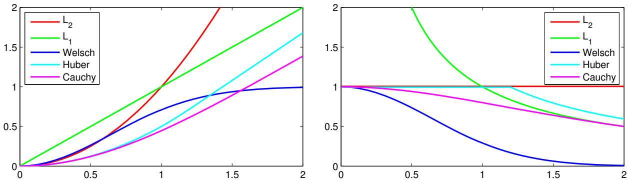


Figure 1: (a) Five popular error functions ℓ. (b) The corresponding weight funtions w.

V. Relations to Existing Works

In this section, we discuss some approaches which are closely related to our methods.

Clearly, the standard NMF can be seen as a special case of weighted NMF by setting $W = \mathbf{1}^{N \times M}$. In order to pursue robustness of NMF, Kong et.al. [11] proposed L_{21}-NMF to minimize L_1 error on rows, i.e.,

$$\min_{U,V} \sum_{i=1}^{N} ||X_{i*} - U_{i*}V^T||, \tag{35}$$

and L_1-NMF to minimize L_1 error on entries, i.e.,

$$\min_{U,V} \sum_{i=1}^{N}\sum_{j=1}^{M} |X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk}|, \tag{36}$$

Besides, the $L_1 - L_2$-NMF [12] was proposed to minimize the following objective function

$$\min_{U,V} \sqrt{1 + \sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2} - 1. \tag{37}$$

Actually, both the L_1 function and the $L_1 - L_2$ function belong to M-estimators, as we mentioned earlier. That is to say, $L_{2,1}$-NMF, L_1-NMF, and $L_1 - L_2$-NMF can also be solved by our generic half-quadratic minimization algorithm in Section 3.

It should be noteworthy that one special case of the weighted NMF in Section 3.2 has been presented in [21], [22] for the task of collaborative filtering. Previous work uses a binary weight matrix with its entries to indicate whether the rating is missing, however, we aim to improve the robustness of NMF by automatically adjust the weights of outliers and inliers.

Finally, our methods also draw connections to recent proposed WFS-NMF [20], which assigns different weights on data points and features to indicate their importance and can be formulated as follows

$$\min_{U,V,W} \sum_{i=1}^{N}\sum_{j=1}^{M} W_{ij}(X_{ij} - \sum_{k=1}^{K} U_{ik}V_{jk})^2 \tag{38}$$

$$\text{s.t.} \quad U \geq 0, V \geq 0, W_{ij} = a_i b_j, \sum_{i=1}^{N} a_i^{\alpha} = 1, \sum_{j=1}^{M} b_j^{\beta} = 1.$$

Similar to L_1-norm, the weight of each entry derived from the L_p-norm on the simplex is also not upper bounded for near-zero errors, which may cause the singularity problem.

VI. Experiments Results

In this section, we present experiments to demonstrate the effectiveness of the proposed NMF variants on data sets with/without apparent outliers.

A. Compared Methods and Parameters Settings

We compare the performance of CIM-NMF, rCIM-NMF, and Huber-NMF with the following methods. (1) Kmeans: standard Kmeans algorithm; (2) PCA-Km: PCA is firstly applied to reduce the data dimension followed by the Kmeans clustering; (3) RPCA-Km: robust PCA [30] is firstly applied for subspace learning and followed by Kmeans; (4) Ncut [31]: a spectral clustering algorithm; (5) NMF [1]: standard NMF algorithm; (6) SR-NMF [13], [14]: a robust NMF under the assumption of sparse noise; (7) $L_{2,1}$-NMF [11]: a robust NMF which uses the $L_{2,1}$ to measure the quality of factorization; (8) WFS-NMF [20]: a weighted NMF where data points and features are assigned to different weights to indicate their importance.

To make a fair comparison, the parameters of these algorithms are set as follows. For PCA and RPCA, the reduced dimensionality is set to preserve 99% variance. For RPCA and SR-NMF, the regularization parameter is searched from the grid $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. For Ncut, the similarities are computed using the standard Gaussian kernel, and the kernel size is set in an automatic way using the method introduced in [32]. For all NMFs, we set the parameter K to be the number of clusters and initialize them using the results of Kmeans, which is often suggested in the literature [4], [11]. The power parameters α and β in WFS-NMF are set to 0.7, as suggested by [20].

B. Evaluation Metrics

To evaluate their performance, we compare the generated clusters with the ground truth by computing the following two performance measures.

Clustering accuracy (ACC). The first performance measure is the clustering accuracy, which discovers the one-to-one relationship between clusters and classes. Given a point $\boldsymbol{x}_i$, let p_i and q_i be the clustering result and the ground truth label, respectively. The ACC is defined as follows:

$$\text{ACC} = \frac{1}{n}\sum_{i=1}^{n} \delta(q_i, map(p_i)), \tag{39}$$

where n is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and $map(\cdot)$ is the permutation mapping function that maps each cluster index to a true class label. The best mapping can be found by using the Kuhn-Munkres algorithm [33].

Table I: Description of the data sets

Data sets	# instances	# classes	# features
COIL20	1440	1024	20
JAFFE	213	676	10
CSTR	476	1000	4
WebKB	827	4134	7

The greater clustering accuracy means the better clustering performance.

Normalized mutual information (NMI). Another evaluation metric that we adopt here is the normalized mutual information, which is widely used for determining the quality of clustering. Let $\mathcal{C}$ be the set of clusters from the ground truth and $\mathcal{C}'$ obtained from a clustering algorithm. Their mutual information $MI(\mathcal{C}, \mathcal{C}')$ is defined as follows:

$$MI(\mathcal{C}, \mathcal{C}') = \sum_{c_i \in \mathcal{C}, c'_j \in \mathcal{C}'} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i) p(c'_j)}, \qquad (40)$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a data point arbitrarily selected from the data set belongs to the cluster c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information as follows:

$$NMI(\mathcal{C}, \mathcal{C}') = \frac{MI(\mathcal{C}, \mathcal{C}')}{\max(H(\mathcal{C}), H(\mathcal{C}'))}, \qquad (41)$$

where $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of $\mathcal{C}$ and $\mathcal{C}'$, respectively. Again, a larger NMI indicates a better performance.

C. Clustering on Data Sets without Apparent Outliers

In this experiment we study the performance of different methods on data sets without apparent outliers. Each clustering algorithm is repeated 10 times and the average clustering result is recorded.

Data Sets Table I summarizes the characteristics of the data sets used in the experiments. Detailed descriptions of the data sets are as follows.

- COIL20. This dataset contains 1,440 grayscale images with black background for 20 objects with each object having 72 different images. Each image is processed to a size of 32 × 32 pixels.
- JAFFE. This database contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects.
- CSTR. This is the dataset of the abstracts of technical reports (TRs) published in the Department of Computer Science at University of Rochester from 1991 to 2002. The dataset contained 476 abstracts, which were divided into four research areas: Natural Language Processing(NLP), Robotics/Vision, Systems, and Theory. We select the top 1000 words by mutual information with class labels.
- WebKB[2]. It contains about 6,000 web pages collected from the web sites of computer science departments of four universities (Cornell, Texas, Washington, and Wisconsin). Each web page is labeled with one out of seven categories: student, professor, course, project, staff, department, and other. The subset from Cornell is used in our experiments.

Table II: Clustering Accuracy

Data set	COIL20	JAFFE	CSTR	WebKB
Kmeans	0.631	0.657	0.727	0.520
PCA-Km	0.638	0.780	0.749	0.527
RPCA-Km	0.652	0.753	0.703	0.615
Ncut	0.380	0.795	0.714	0.430
NMF	0.651	0.861	0.758	0.557
SR-NMF	**0.671**	0.839	0.777	0.603
$L_{2,1}$-NMF	0.658	0.893	0.771	0.614
WFS-NMF	0.649	0.879	**0.784**	**0.664**
Huber-NMF	0.661	**0.928**	0.765	0.617
rCIM-NMF	**0.695**	0.882	**0.798**	**0.675**
CIM-NMF	0.670	**0.927**	0.761	0.652

Table III: Clustering NMI

Data set	COIL20	JAFFE	CSTR	WebKB
Kmeans	0.743	0.745	0.651	0.042
PCA-Km	0.743	0.821	0.663	0.056
RPCA-Km	0.755	0.831	0.603	0.022
Ncut	0.578	0.833	0.638	0.151
NMF	0.679	0.859	0.665	0.155
SR-NMF	**0.758**	0.862	0.687	**0.176**
$L_{2,1}$-NMF	0.713	0.908	0.681	0.157
WFS-NMF	0.740	0.872	**0.687**	0.013
Huber-NMF	0.743	**0.932**	0.675	0.172
rCIM-NMF	**0.755**	0.897	**0.691**	**0.177**
CIM-NMF	0.753	**0.942**	0.668	0.153

Clustering Results: Tables II and III show the clustering accuracy and NMI results on these data sets. The best two results are shown in bold. From the experimental comparisons, we observe that: 1) the weighted variants of NMF, i.e., the last six methods, usually outperform the standard NMF since they decrease the weights of entries, rows, or columns with large fitting errors. This may indicate that, though these data sets are usually used for non-robust learning, a careful weighting scheme can still improve the performance of unweighted NMF; 2) our proposed methods, CIM-NMF, rCIM-NMF and Huber NMF, often outperform other algorithms on these datasets. For image data sets, CIM-NMF and Huber-NMF perform better that others, one plausible reason is that the possible outliers are scattered among the whole matrix. While rCIM-NMF performs better on text data sets, which is consistent with the normalized cut weighting (NCW) scheme used in [9].

[2] http://www.nec-labs.com/~zsh/files/link-fact-data.zip

Table IV: Clustering Accuracy on the ORL face database (mean% ± std)

$r(\%)$	Kmeans	PCA-Km	RPCA-Km	Ncut	NMF	SR-NMF	$L_{2,1}$-NMF	WFS-NMF	Huber-NMF	rCIM-NMF	CIM-NMF
5	51.1±2.9	50.6±1.2	52.1±2.5	44.2±2.9	54.7±3.2	58.2±2.1	56.0±3.4	56.5±3.0	57.1±2.9	56.7±3.3	**61.0±2.1**
10	48.7±2.5	48.4±2.6	48.8±1.9	34.6±1.5	51.5±3.0	54.3±2.7	52.9±2.8	53.6±2.7	55.4±2.2	53.9±2.7	**60.7±3.5**
15	45.5±2.6	45.4±2.0	45.7±2.0	33.8±2.4	49.6±1.7	50.5±3.0	49.9±2.6	50.5±2.5	51.8±3.0	50.6±2.8	**58.8±3.3**
20	43.3±2.4	43.7±2.4	43.8±2.6	35.0±2.2	45.9±2.5	48.0±2.2	47.1±2.5	47.8±2.1	50.2±2.5	47.8±2.3	**57.2±2.6**
25	40.5±2.3	41.5±2.3	41.5±1.7	35.4±2.0	44.2±2.8	46.1±1.8	44.9±2.7	45.5±3.4	47.6±1.8	44.6±2.5	**54.8±3.5**
30	39.1±2.2	39.8±1.7	39.2±1.7	35.1±1.6	42.2±2.8	42.7±2.7	42.5±2.2	42.1±2.8	44.5±2.7	42.1±2.5	**51.6±3.2**
35	37.1±1.9	37.3±1.7	38.1±2.1	34.4±1.8	38.7±1.9	40.6±2.0	39.4±2.1	39.6±2.4	41.3±2.3	40.6±2.0	**47.1±2.4**
40	34.6±1.6	35.0±1.7	35.8±1.7	32.8±1.5	36.9±2.2	38.0±2.0	37.1±1.7	38.7±2.5	38.4±1.6	37.0±2.1	**43.0±2.7**
45	34.1±1.5	34.0±1.4	34.5±1.5	33.3±1.3	36.2±2.1	37.1±1.3	37.9±2.5	36.1±3.1	37.6±2.2	36.8±1.7	**42.0±2.8**
50	32.1±1.6	32.8±1.5	33.2±1.3	32.3±1.7	34.4±1.8	35.4±2.0	35.0±2.1	35.4±2.4	35.9±2.0	35.0±2.1	**39.7±2.1**
Avg.	40.6	40.8	41.3	35.1	43.4	45.1	44.3	44.6	46.0	44.5	**51.6**

Table V: Clustering NMI on the ORL face database (mean% ± std)

$r(\%)$	Kmeans	PCA-Km	RPCA-Km	Ncut	NMF	SR-NMF	$L_{2,1}$-NMF	WFS-NMF	Huber-NMF	rCIM-NMF	CIM-NMF
5	70.8±1.4	70.6±1.1	72.0±1.2	64.4±2.9	72.1±1.5	74.8±1.0	73.4±1.9	73.6±2.2	74.3±2.0	74.2±1.9	**77.6±1.6**
10	68.4±1.1	67.9±1.5	69.3±1.0	54.9±1.6	69.5±1.5	71.9±1.5	70.7±1.5	71.0±1.8	72.4±1.2	70.9±2.0	**77.4±1.6**
15	65.5±1.5	65.4±1.3	66.7±0.9	54.8±2.6	67.7±1.2	68.3±1.3	68.1±1.6	68.6±1.8	69.6±1.4	68.2±1.8	**75.7±2.1**
20	63.8±1.3	64.0±1.4	64.6±1.3	56.5±2.0	65.3±1.7	66.6±1.7	65.5±1.4	66.1±1.3	67.0±1.4	66.6±1.5	**74.4±1.4**
25	61.4±1.4	62.0±1.1	63.1±1.3	57.8±2.0	63.4±1.8	65.0±1.4	63.9±1.6	64.0±1.6	64.9±1.5	63.5±1.8	**72.1±1.9**
30	59.8±1.6	60.4±1.0	61.2±1.0	57.3±1.8	61.4±2.1	62.2±1.8	61.7±1.5	62.1±1.5	62.8±1.6	61.6±1.3	**69.9±2.3**
35	57.8±1.2	58.6±1.4	60.0±1.2	56.5±1.9	59.5±1.3	60.7±1.5	59.8±1.8	59.3±1.4	60.0±1.9	60.9±1.5	**66.6±1.7**
40	56.6±1.0	56.8±1.3	57.9±1.2	55.7±1.5	57.5±1.3	59.0±1.5	58.0±1.0	58.8±1.2	59.1±1.6	57.5±1.3	**63.8±2.0**
45	56.2±1.6	56.1±1.1	57.0±1.4	56.3±1.1	57.2±1.3	58.1±1.0	58.1±1.3	58.1±1.4	58.7±1.5	57.6±1.4	**62.7±1.5**
50	54.4±1.4	55.1±1.0	56.0±1.3	55.2±1.6	55.8±1.7	56.9±1.5	56.4±1.3	56.6±1.1	56.5±1.6	56.9±1.4	**60.1±1.3**
Avg.	61.5	61.7	62.8	56.9	63.0	64.4	63.5	63.9	64.5	63.8	**70.0**

D. Experiments on Face Database with Malicious Occlusion

In this experiment, we aim to test the performance of the compared algorithms when data sets are contaminated with outliers. The ORL face database is used in this experiment. It contains 400 gray scale images of 40 individuals. The images are captured at different times, under different lighting conditions, with different facial expression and with/without glasses. All the face images are manually aligned and cropped. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Thus, each image is represented as a 1024-dimensional vector.

To simulate outliers, different percents of images ($r = 5\%, 10\%, \ldots, 50\%$) are randomly selected and partially occluded on some key facial features (eyes or mouth) according to [24]. To reduce statistical variety, 20 tests were conducted on different randomly chosen percent of outliers, and the average performance as well as the standard derivation is reported.

Clustering Results: Tables IV and V show the detailed clustering results measured by Accuracy and NMI, respectively. As can be seen, our CIM-NMF algorithm significantly outperforms the other algorithms in all the cases. Compared with the second best algorithm, our method, CIM-NMF, achieves 12.1% relative improvement in clustering accuracy. For mutual information, it achieves 8.5% improvement over the second best algorithm.

For all the compared algorithms, their clustering performances decrease with the increase of outliers. Interestingly, when the number of outliers is relatively small (e.g., $r = 5\%, 10\%, 15\%$), we observe that: 1) the results of CIM-NMF is relatively stable while other NMF variants degrade quickly; 2) all other variants of NMF outperform standard NMF, which may indicate these variants are robust to outliers in some extent; 3) CIM-NMF achieves similar results on these occlusion, that is to say, the performance of CIM-NMF is almost not affected by these outliers. When r becomes larger, CIM-NMF still performs best and all other NMF variants achieve similar results to standard NMF. All these results show that CIM-NMF is more robust over a large range of outliers.

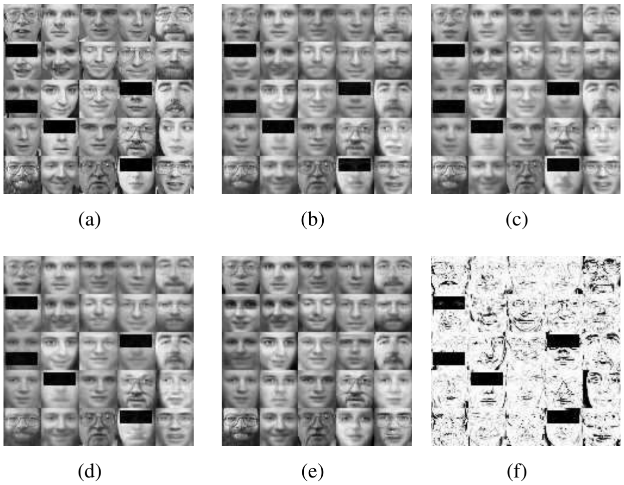
Classification Results: To further investigate the performance of these compared algorithms on corrupted data, we perform nearest neighbor classification based on the learned low dimensional representation. We select 3 images per subject as the training data and the rest are used for testing. For each differently chosen percentage of outliers, 50 training/testing splits are randomly generated and thus 20×50 classification accuracies are recorded.

Table VI shows the detailed classification accuracies of compared algorithms. Here, we use all features without dimension reduction as a weak baseline. Ncut is used to compute the graph embedding based representation. Again, CIM-NMF significantly outperforms all other algorithms in all cases. It achieves 15.4% relative improvement against the second best algorithm.

Visualization of Reconstructed Faces: To get a better understanding of our approach, Figure 2 shows the reconstructed images of several NMFs and the weight matrix of CIM-NMF. Here, we randomly select 25 face images

Table VI: Classification Accuracy on the ORL face database (mean% ± std)

$r(\%)$	All Features	PCA	RPCA	Ncut	NMF	SR-NMF	$L_{2,1}$-NMF	WFS-NMF	Huber-NMF	rCIM-NMF	CIM-NMF
5	73.3±2.4	73.6±2.3	73.4±2.4	69.9±2.7	73.4±2.2	77.2±2.3	75.8±2.3	75.4±2.6	77.8±2.4	75.4±2.5	**81.1±2.5**
10	68.8±2.5	69.5±2.4	68.9±2.5	56.9±2.8	70.2±2.5	72.5±2.4	71.3±2.5	71.9±2.7	73.0±2.6	70.7±2.6	**81.0±2.4**
15	64.6±2.6	65.3±2.7	64.6±2.6	54.0±2.8	65.5±2.6	67.8±2.6	67.0±2.5	66.8±2.5	67.8±2.6	66.2±2.7	**79.2±2.8**
20	60.8±2.8	61.1±2.6	60.8±2.7	51.4±3.3	61.3±2.7	63.1±2.6	62.2±2.7	62.4±2.4	63.3±2.7	61.9±2.6	**76.2±2.8**
25	58.1±2.7	58.7±2.8	57.9±2.7	50.0±3.0	59.1±2.7	60.0±2.8	59.1±2.8	59.7±2.6	60.6±2.9	58.5±2.7	**72.6±2.9**
30	55.0±2.9	55.3±3.0	54.8±2.8	48.1±2.9	55.0±3.1	56.2±3.2	55.4±3.1	55.8±2.1	56.8±2.9	54.5±3.4	**68.3±3.1**
35	52.2±2.9	52.3±3.1	51.8±2.9	45.9±3.2	52.1±3.3	53.0±3.0	52.2±3.0	52.3±2.6	53.1±3.1	51.9±2.9	**63.1±3.2**
40	49.3±2.9	49.7±3.0	48.7±2.9	43.8±3.3	49.3±3.1	50.1±2.9	49.2±3.0	49.6±2.0	50.1±3.3	48.6±3.1	**59.3±3.2**
45	48.2±3.0	48.4±3.0	47.7±3.0	43.8±3.0	47.5±3.2	48.1±3.1	47.5±3.4	48.2±2.2	48.7±3.2	47.5±3.0	**56.0±3.3**
50	45.7±3.1	45.6±3.1	45.1±3.1	41.9±3.0	44.7±3.1	45.8±3.2	45.0±3.2	45.4±2.8	46.0±3.3	44.4±3.1	**51.7±3.4**
Avg.	57.6	58.0	57.4	50.6	57.8	59.4	58.5	58.8	59.7	58.0	**68.9**


(a) (b) (c)

(d) (e) (f)

Figure 2: (a) Original images corrupted by occlusion. (b), (c), (d), (e) Reconstructed images by NMF, SR-NMF, $L_{2,1}$-NMF, and CIM-NMF. (f) Weight matrix learned by CIM-NMF.

from one test with $r = 20\%$ corrupted samples. As can be seen, the faces reconstructed by CIM-NMF are much clearer than compared algorithms and the weight matrix of CIM indeed assign small weights to obvious outliers (continuous occlusion). The SR-NMF and $L_{2,1}$-NMF can hardly identify the entries or samples which contain outliers.

Visualization of Basis Vectors: In this test, we randomly choose one test with $r = 20\%$ corrupted samples and show the basis vectors obtained by NMF, and CIM-NMF in Figure 3. As can be seen, 10 basis vectors in NMF are partially occluded, while only 4 subjects are occluded in CIM-NMF. Comparing the basis vectors obtained by CIM-NMF with the results of NMF, we find that our approach CIM-NMF can indeed generate much clear basis vectors which are less affected

VII. Conclusion

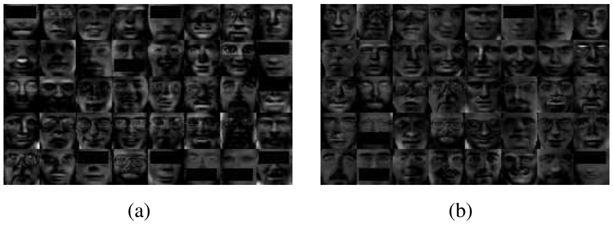

(a) (b)

Figure 3: (a) Basis vectors learned by NMF. (b) Basis vectors learned by CIM-NMF.

In this paper, we propose a novel robust NMF method based on the correntropy induced metric, called CIM-NMF. To deal with the minimization of non-quadratic and non-convex functions efficiently, we develop an iterative algorithm based on the half-quadratic minimization technique. At each iteration, the optimization problem is reduced to a weighted least square NMF, which can be solved in a similar way to standard NMF. The proposed method is further extended to handle outlier rows by incorporating structural knowledge about the outliers, which leads to rCIM-NMF. Due to the connection between CIM and robust M-estimators, we also extend to use the Huber's function to measure the quality of approximation and have Huber-NMF. The optimization problems for both rCIM-NMF and Huber-NMF are also solved by the algorithm developed for CIM-NMF. Experimental results on benchmark datasets with/without outliers demonstrate the effectiveness of the proposed methods.

Several questions remain to be investigated in our future work. There are many different robust loss functions, it remains unclear how to choose these functions theoretically. The robustness of the proposed algorithms depends on a free parameter, such as scale parameter σ and cutoff parameter c. Though an empirical strategy is provided, we plan to further investigate other parameter selection methods suggested in [25], [15], [28]. The robustness is pursued on stand NMF, it is also interested to improve the robustness of other NMF variants such as NMTF [4], and SymNMF [34].

VIII. Acknowledgments

We would like to thank all anonymous reviewers for their helpful comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) grants 60970045 and 60833001.

References

[1] D. Lee, H. Seung *et al.*, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[2] P. Hoyer, "Non-negative matrix factorization with sparseness constraints," *JMLR*, vol. 5, pp. 1457–1469, 2004.

[3] A. Pascual-Montano, J. Carazo, K. Kochi, D. Lehmann, and R. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsnmf)," *PAMI*, vol. 28, no. 3, pp. 403–415, 2006.

[4] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD*, 2006, pp. 126–135.

[5] D. Cai, X. He, J. Han, and T. Huang, "Graph regularized non-negative matrix factorization for data representation," *PAMI*, vol. 33, no. 8, pp. 1548–1560, 2011.

[6] C. Ding, T. Li, and M. Jordan, "Convex and semi-nonnegative matrix factorizations," *PAMI*, vol. 32, no. 1, pp. 45–55, 2010.

[7] J. Brunet, P. Tamayo, T. Golub, and J. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *PNAS*, vol. 101, no. 12, pp. 4164–4169, 2004.

[8] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of the 2001 IEEE CVPR*, vol. 1, 2001, pp. I–207.

[9] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th ACM SIGIR*, 2003, pp. 267–273.

[10] R. Sandler and M. Lindenbaum, "Nonnegative matrix factorization with earth mover's distance metric for image analysis," *PAMI*, vol. 33, no. 8, pp. 1590–1602, 2011.

[11] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using ℓ_{21}-norm," in *Proceedings of the 20th ACM CIKM*, 2011, pp. 673–682.

[12] A. Hamza and D. Brady, "Reconstruction of reflectance spectra using robust nonnegative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3637–3642, 2006.

[13] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust non-negative matrix factorization," *Frontiers of Electrical and Electronic Engineering in China*, pp. 1–9, 2011.

[14] B. Shen, L. Si, R. Ji, and B. Liu, "Robust nonnegative matrix factorization via l_1 norm regularization," *Arxiv preprint arXiv:1204.2311*, 2012.

[15] W. Liu, P. P. Pokharel., and J. C. Principe, "Correntropy: properties and applications in non-gaussian signal processing," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5286–5298, 2007.

[16] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *NIPS*, vol. 13, pp. 556–562, 2001.

[17] M. Nikolova and R. Chan, "The equivalence of half-quadratic minimization and the gradient linearization iteration," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1623–1627, 2007.

[18] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.

[19] M. Nikolova and M. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM Journal on Scientific computing*, vol. 27, no. 3, pp. 937–966, 2006.

[20] D. Wang, T. Li, and C. Ding, "Weighted feature subset non-negative matrix factorization and its applications to document understanding," in *2010 IEEE ICDM*, 2010, pp. 541–550.

[21] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," *SIAM SDM*, 2006.

[22] Q. Gu, J. Zhou, and C. Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *SIAM SDM*, 2010, pp. 199–210.

[23] J. Principe, D. Xu, and J. Fisher, "Information theoretic learning," *Unsupervised Adaptive Filtering*, vol. 1, pp. 265–319, 2000.

[24] X. Yuan and B. Hu, "Robust feature extraction via information theoretic learning," in *ICML*, 2009, pp. 1193–1200.

[25] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *PAMI*, vol. 33, no. 8, pp. 1561–1576, 2011.

[26] P. Huber, E. Ronchetti, and MyiLibrary, *Robust statistics*, 1981, vol. 1.

[27] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[28] C. Ding, D. Zhou, X. He, and H. Zha, "R1-pca: rotational invariant l1-norm principal component analysis for robust subspace factorization," in *ICML*, 2006, pp. 281–288.

[29] D. Luo, C. Ding, F. Nie, and H. Huang, "Cauchy graph embedding," in *ICML*, 2011, pp. 553–560.

[30] E. CANDES, Y. MA, J. WRIGHT *et al.*, "Robust principal component analysis?" *Journal of the Association for Computing Machinery*, vol. 58, no. 3, 2011.

[31] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *NIPS*, vol. 2, pp. 849–856, 2002.

[32] P. Perona and L. Zelnik-Manor, "Self-tuning spectral clustering," *NIPS*, vol. 17, pp. 1601–1608, 2004.

[33] L. Lovász and M. Plummer, *Matching theory*, 1986, no. 121.

[34] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *SIAM SDM*, 2012, pp. 106–117.