Robust Nonnegative Matrix Factorization using L21-norm

Deguang Kong
Dept. of Computer Science and Engineering, University of Texas at Arlington,
doogkong@gmail.com

Chris Ding
Dept. of Computer Science and Engineering, University of Texas at Arlington,
CHQDing@uta.edu

Heng Huang
Dept. of Computer Science and Engineering, University of Texas at Arlington,
heng@uta.edu

ABSTRACT

Nonnegative matrix factorization (NMF) is widely used in data mining and machine learning fields. However, many data contain noises and outliers. Thus a robust version of NMF is needed. In this paper, we propose a robust formulation of NMF using $L_{2,1}$ norm loss function. We also derive a computational algorithm with rigorous convergence analysis. Our robust NMF approach, (1) can handle noises and outliers; (2) provides very efficient and elegant updating rules; (3) incurs almost the same computational cost as standard NMF, thus potentially to be used in more real world application tasks. Experiments on 10 datasets show that the robust NMF provides more faithful basis factors and consistently better clustering results as compared to standard NMF.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Theory

Keywords

NMF, L21 norm, Robust, Clustering

1. INTRODUCTION

Nonnegative Matrix Factorization (NMF) has been popularly studied in data mining and machine learning areas since the initial work of Lee and Seung [13]. Originally proposed as a method for finding matrix factors with parts-of-whole interpretations [13], NMF has been applied to a number of different areas, e.g., environmetrics [19], chemometrics [25], pattern recognition [14], multimedia data analysis [3], text mining [20] and DNA gene expression analysis [1]. Algorithmic extensions of NMF also have been developed to accommodate a variety of objective functions [4, 7] into different data analysis problems, including classification [21] collaborative filtering [22], and constrained clustering [15, 23]. It also can be extended by making a combination with Laplacian embedding [2, 11]. One of the key features of NMF is its clustering capabilities. It was shown [5, 8] that NMF essentially solves a matrix clustering problem.

Standard NMF uses the least square error function which is well-known to be unstable w.r.t. noises and outliers [24]. However, many real data in various applications probably contain noise and outliers. Potential applications in real world drive us to consider about a robust version of NMF. For this reason, a robust NMF model is studied in this paper.

To our knowledge, a robust NMF has not been studied so far. In this paper, we propose a novel robust formulation of NMF by using $L_{2,1}$-norm loss function[1]. The proposed method is termed as "robust" because it can accommodate outliers and noises in a better way than standard one. We derive the computational algorithm and provide rigorous analysis on its convergence and correctness. More importantly, the derived solution for robust NMF has very elegantly updating rules, with nearly the same computation cost as standard NMF, and also easy for implementation. We perform experiments on 10 datasets using both robust NMF and standard NMF. On all 10 datasets, robust NMF consistently outperforms standard NMF in terms of clustering results.

The merits of our Robust NMF are in threefold.

- Robust NMF can handle outliers and noises.
- Robust NMF provides very efficient and elegant updating rules.
- Robust NMF incurs almost the same computation cost as standard NMF, convenient for applications in different contexts.

The rest of the paper is organized as follows. In section 2 we propose the robust NMF formulation using $L_{2,1}$ norm, emphasize the advantages of our approach compared with standard NMF. In section 3, we present a rigorous convergence analysis of the algorithm. In section 4, we show that the converged solution satisfies the Karush-Kuhn-Tucker condition and thus is a correct optimal solution. In section 5 we present experimental results on 10 datasets. We show convergence properties, speed, and clustering results by making comparisons with standard NMF. Finally we discuss the extension of $L_{2,1}$ NMF to L_1 NMF followed by the conclusion.

2. ROBUST NMF USING $L_{2,1}$ NORM

In this section we first revisit standard NMF, show the assumption of the Gaussian noise leads to the formulation of standard NMF

[1]Robust NMF can also be formalized as L_1-NMF. We discuss the extension to L_1-NMF in section 6. In this paper, we focus on robust NMF with $L_{2,1}$-norm.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

by imposing constraints $F \geq 0,\ G \geq 0$; then we present our robust NMF formulation, show the assumption of the Laplacian noise leads to the formulation of $L_{2,1}$ NMF; next we give two illustrative examples both on toy data and real data to show the robustness and effectiveness of $L_{2,1}$ NMF; finally we present the computational algorithm for $L_{2,1}$ NMF, and highlight the main contribution of our paper.

2.1 Review on standard NMF

Given input data vectors $X = (x_1, x_2, ...x_n)$, where $x_i \in R^p$ represents a data point. Standard NMF is defined as,

$$\min_{F,G} \ \|X - FG\|_F^2, \ \ s.t. \ \ F \geq 0,\ G \geq 0, \tag{1}$$

where $\|X\|_F^2 = \sum_{ij} X_{ij}^2$ is the Frobenius form of a matrix. Above problem is usually solved by an iterative updating algorithm, where F and G are updated alternatively using

$$F_{jk} \Leftarrow F_{jk} \frac{(XG^T)_{jk}}{(FGG^T)_{jk}}, \tag{2}$$

$$G_{ki} \Leftarrow G_{ki} \frac{(F^T X)_{ki}}{(F^T FG)_{ki}}. \tag{3}$$

2.2 From Gaussian noise to standard NMF

In general, the input data x_i is a p-dimensional column vector contaminated by additional noise,

$$x_i = \theta_i + \varepsilon_i, \tag{4}$$

where θ_i is the unobservable true value of the observed x_i, and ε_i is the additive noise. θ_i can be viewed as a point in a k-dimensional subspace ($k < p$) such that,

$$\theta_i = Fg_i, \tag{5}$$

where g_i is the projection of x_i on the subspace defined by columns of F.

Suppose the noise ε_i follows zero-mean normal distribution with standard deviation of σ, thus $x_i \sim N(\theta_i, \sigma^2)$. Usually the elements of each vector x_i in X are independent, thus the probability distribution of x_i conditioned on θ_i is,

$$p(x_i|\theta_i) \sim \exp\{-\frac{||x_i - \theta_i||^2}{2\sigma^2}\} \tag{6}$$

The data log likelihood can be written as,

$$\log \prod_{i=1}^{n} p(x_i|\theta_i) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} ||x_i - \theta_i||^2 \tag{7}$$

To maximize the data log likelihood is equivalent to minimize the term $\sum_{i=1}^{n} ||x_i - \theta_i||^2$ in Eq.(7). Thus we have Eq. (8) by substituting θ_i with Fg_i by using Eq. (5),

$$\min_{\theta_i} \sum_{i=1}^{n} ||x_i - \theta_i||^2 = \min_{F,g_i} \sum_{i=1}^{n} ||x_i - Fg_i||^2 = \min_{F,G} ||X - FG||_F^2 \tag{8}$$

where g_i is the i-th column of G. This means the assumption of i.i.d Gaussian noise model transfers the maximum likelihood problem into a standard NMF problem by imposing constraints $F \geq 0, G \geq 0$.

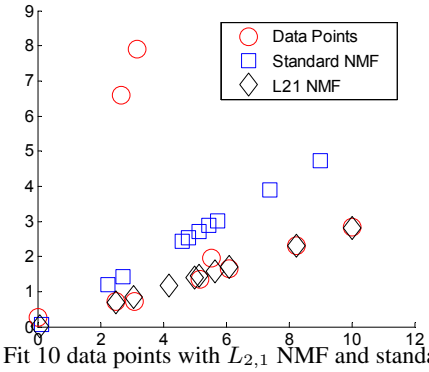

Figure 1: Fit 10 data points with $L_{2,1}$ NMF and standard NMF by using Fg_i. Two data points on upper-left are outliers.

2.3 Robust NMF using $L_{2,1}$ norm

One of the most important drawbacks of standard NMF is that it is prone to outliers. Let $X = (x_1, \cdots, x_n),\ G = (g_1, \cdots, g_n)$. The error function of standard NMF is

$$\|X - FG\|_F^2 = \sum_{i=1}^{n} \|x_i - Fg_i\|^2 \tag{9}$$

Here the error for each data point enters the objective function as **squared** residue error in the form of $\|x_i - Fg_i\|^2$. Thus a few outliers with large errors easily dominate the objection function because of the squared errors. Note in NMF, both matrices F, G are unknown, the impact of the outliers may be more complicate than the simpler convex case. Thus it is very necessary to present robust NMF formulation and discuss its properties.

The robust formulation of the error function is

$$\|X - FG\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} (X - FG)_{ji}^2} = \sum_{i=1}^{n} \|x_i - Fg_i\| \tag{10}$$

In this robust formulation, the error for each data point is $\|x_i - Fg_i\|$, which is not squared, and thus the large errors due to outliers do not dominate the objective function because they are not squared.

For this reason, in this paper, we propose robust NMF ($L_{2,1}$ NMF) formulated as

$$\min_{F,G} \ \|X - FG\|_{2,1} \ \ s.t. \ \ F \geq 0,\ G \geq 0 \tag{11}$$

$L_{2,1}$ norm of a matrix A is first introduced in [9] and is defined as

$$\|A\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{p} A_{ji}^2} = \sum_{i=1}^{n} \|a_i\|. \tag{12}$$

where a_i is the ith column of A. In [9] it is called rotational invariant L_1 norm, because the column vectors a_i contribute in the form of vector Euclidean norm which is rotational invariant. $\|\cdot\|_{2,1}$ is a valid *norm* because it satisfies the 3 conditions for a norm: (1) positive scalability: $\|\alpha A\|_{2,1} = |\alpha| \|A\|_{2,1}$ where α is a real scalar; (2) triangle inequality: $\|A + B\|_{2,1} \leq \|A\|_{2,1} + \|B\|_{2,1}$; (3) existence of a zero vector: if $\|A\|_{2,1} = 0$, then $A = 0$. These 3 properties can be easily proved.

Note that in general, the $L_{2,1}$ norm of Eq.(11) is harder to solve than the least squares of the standard NMF of Eq.(1). One main contribution of our paper is to derive an efficient algorithm to solve the $L_{2,1}$ formulation of Eq.(11). We will present the algorithm in section 2.7.

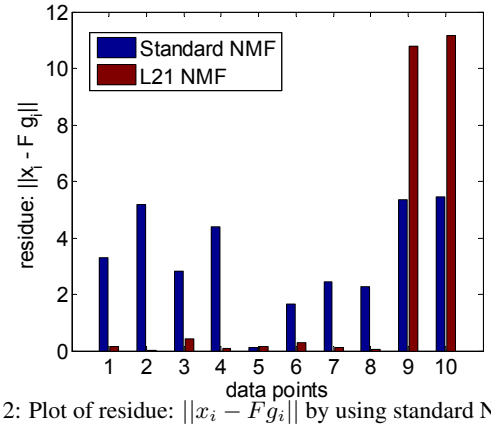

Figure 2: Plot of residue: $||x_i - Fg_i||$ by using standard NMF and $L_{2,1}$ NMF for each of 10 data points in Fig. 1. Data points #9 and #10 are outliers.

2.4 From Laplacian noise to $L_{2,1}$ NMF

Further, we show the motivation of $L_{2,1}$ NMF from probability point of view. Different from the normal distribution assumption in section 2.2, if we assume the noise ε_i follows the Laplacian distribution with zero mean, we have

$$p(x_i|\theta_i) \sim \exp\{-\frac{||x_i - \theta_i||}{\lambda}\}, \tag{13}$$

where λ is a scale parameter.

Following the strategy of maximizing the data log likelihood, we obtain,

$$\max_{\theta_i} \log \prod_{i=1}^{n} p(x_i|\theta_i) = \max_{\theta_i} -\frac{1}{\lambda} \sum_{i=1}^{n} ||x_i - \theta_i|| \tag{14}$$

To maximize the data log likelihood is equivalent to minimize the term $\sum_{i=1}^{n} ||x_i - \theta_i||$ in Eq.(14). Thus we have Eq. (15) by substituting θ_i with Fg_i,

$$\min_{\theta_i} \sum_{i=1}^{n} ||x_i - \theta_i|| = \min_{F,g_i} \sum_{i=1}^{n} ||x_i - Fg_i|| = \min_{F,G} ||X - FG||_{2,1} \tag{15}$$

This means the assumption of i.i.d Laplacian noise model transfers the maximum likelihood problem into a $L_{2,1}$ NMF problem by imposing constraints $F \geq 0, G \geq 0$.

2.5 Illustration of robust NMF on toy data

We use 2-dimensional toy data to show the robustness of $L_{2,1}$ NMF in Fig.(1). 10 original 2-dimensional data points are generated, two of which are outliers. For each data point, we use Fg_i to fit the original data point. Here we project all data into 1D subspace (i.e., $k = 1$). Both the fitting values from $L_{2,1}$ and standard NMF are shown in Fig. 1. $L_{2,1}$ NMF gives much better results while standard NMF is greatly influenced by two outliers. In Fig.(2), we also plot the residue $||x_i - Fg_i||$ corresponding to each data point of Fig.(1). Clearly, $L_{2,1}$ NMF gives much smaller errors compared with standard NMF.

2.6 Illustration of robust NMF on real data

To further illustrate the effectiveness of our $L_{2,1}$ NMF, we run $L_{2,1}$ NMF algorithm on real world data sets. Due to space limit, here we only show the results obtained from AT&T face image data set and Yale face data set in Figs. (3-4). On these two datasets, each x_i is corresponding to an image, which is linearized into a vector. The basis factor f_k in the computed $F = (f_1, f_2, ..., f_K)$ is thus an image. After NMF algorithms converge, we can reshape each f_k into its original size and show each f_i as an image.

Results on AT&T data and Yale data are shown in Fig.(3) and Fig.(4), respectively. Upper images of each row show the robust NMF results. Lower images of each row show the results of standard NMF. Better visual effects indicate better performance of the algorithm. Generally, $L_{2,1}$ NMF performs much better than standard NMF.

One example is the 6th upper image and the corresponding lower image in the top row of Fig.(3). You can see the clear differences between them. First of all, upper image is much clearer by keeping most of the pixel information. It is hard to tell "who is who" from the lower one. Next, most of the important pixels (e.g., pixels on eyes, month, nose) are preserved on upper image while some of them(e.g., eyes, month, cheek) are lost with noises of distorted pixels (e.g., anamorphic nose). Finally, compared to the particular category it comes from, the upper image resembles much more than the lower one.

Another example is images from Yale data set. See the 2nd face image in the upper part and the corresponding one in the lower part of top row of Fig.(4). The upper image is much better than the lower one. First of all, the lower image is fuzzy and shaded while the upper one is unshaded and vivid. Secondly, without any distortion and transformation, the upper image preserves most of the pixels while the lower one loses nearly half of the face pixels, and resembles more to the faces from the other categories (e.g., the 8th face in the 3rd row). Finally, the upper one is more similar to the particular category it is originated from.

From above results on real world data, we can see the superiority of robust NMF. Due to the noises and outliers existed in real data, it is reasonable for $L_{2,1}$ NMF to produce better results, which validates the effectiveness of our approach.

2.7 Computation Algorithm for $L_{2,1}$ NMF

The main contribution of this paper is to derive the following iteratively updating algorithm for Eq. (11),

$$F_{jk} \Leftarrow F_{jk} \frac{(XDG^T)_{jk}}{(FGDG^T)_{jk}} \tag{16}$$

$$G_{ki} \Leftarrow G_{ki} \frac{(F^T XD)_{ki}}{(F^T FGD)_{ki}} \tag{17}$$

where D is a diagonal matrix with the diagonal elements given by

$$D_{ii} = 1 \Big/ \sqrt{\sum_{j=1}^{p} (X - FG)_{ji}^2} = 1/\|x_i - Fg_i\|. \tag{18}$$

The computational algorithm for robust NMF is surprisingly simple. It has almost the same computational cost as standard NMF. It is easy to be adapted for various applications in different context. To our knowledge, this is the first study of the robust NMF with $L_{2,1}$ norm formulation.

We provide the proof of the convergence of the algorithm in section 3 and the correctness of the algorithm in section 4. We note that similar algorithmic approach has been used in $L_{2,1}$ based regression model [17] where no nonnegativity constraints are involved.

3. CONVERGENCE OF THE ALGORITHM

In this section, our main goal is to prove the convergence of the algorithm described in Theorem 1.

Figure 3: Computed $F = (f_1, \cdots f_K)$ on AT&T dataset ($K = 40$) shown as 4 rows. Upper images in each row are $L_{2,1}$ NMF results and lower images are standard NMF results.

Figure 4: Computed $F = (f_1, \cdots f_K)$ on Yale dataset ($K = 31$) shown as 3 rows. Upper images in each row are $L_{2,1}$ NMF results and lower images are standard NMF results.

THEOREM 1. *(A) Updating G using the rule of Eq.(17) while fixing F, the objective function of Eq.(11) monotonically decreases. (B) Updating F using the rule of Eq.(16) while fixing G, the objective function of Eq.(11) monotonically decreases.*

We prove (A,B) separately in next two subsections.

3.1 Updating G

We focus on updating G while fixing F. The proof of Theorem 1(A) requires the following two lemmas.

LEMMA 2. *Let G^t be the old G [on the RHS of Eq.(17)] and G^{t+1} be the new G [on the LHS of Eq.(17)]. Under the updating rule of Eq.(17), the following inequation holds*

$$\begin{aligned} &Tr(X - FG^{t+1})D(X - FG^{t+1})^T \\ &\leq Tr(X - FG^t)D(X - FG^t)^T, \end{aligned} \tag{19}$$

where $D_{ii} = 1/\|x_i - Fg_i^t\|$.

The proof of Lemma 2 is given in section 3.3.

LEMMA 3. *Under the updating rule of Eq.(17), the following inequation holds*

$$\|X - FG^{t+1}\|_{2,1} - \|X - FG^t\|_{2,1} \leq \tag{20}$$

$$\begin{aligned} &\frac{1}{2}\Big[Tr(X - FG^{t+1})D(X - FG^{t+1})^T \\ &\quad -Tr(X - FG^t)D(X - FG^t)^T\Big], \end{aligned} \tag{21}$$

where $D_{ii} = 1/\|x_i - Fg_i^t\|$.

The proof of Lemma 3 is given in section 3.4.

PROOF. (Theorem 1(A)). From Lemma 2, the value of expression inside $[\,\cdot\,]$ in Eq.(21) is negative or zero. Therefore

$$\|X - FG^{t+1}\|_{2,1} - \|X - FG^t\|_{2,1} \leq 0 \tag{22}$$

This proves that the objective function of Eq.(11) decreases monotonically. □

3.2 Updating F

We now focus on updating F while fixing G. Similarly, the proof of Theorem 1(B) also requires the following two lemmas.

LEMMA 4. *Let F^t be the old F [on the RHS of Eq.(16)] and F^{t+1} be the new F [on the LHS of Eq.(16)]. Under the updating rule of Eq.(16), the following inequation holds*

$$\begin{aligned} &Tr(X - F^{t+1}G)D(X - F^{t+1}G)^T \\ &\leq Tr(X - F^tG)D(X - F^tG)^T, \end{aligned} \tag{23}$$

where $D_{ii} = 1/\|x_i - F^t g_i\|$.

The proof of Lemma 4 is similar to the proof of Lemma 2 and thus is skipped due to space limitations.

LEMMA 5. *Under the updating rule of Eq.(16), the following inequation holds*

$$\|X - F^{t+1}G\|_{2,1} - \|X - F^tG\|_{2,1} \leq \tag{24}$$

$$\begin{aligned} &\frac{1}{2}\Big[Tr(X - F^{t+1}G)D(X - F^{t+1}G)^T \\ &\quad -Tr(X - F^tG)D(X - F^tG)^T\Big], \end{aligned} \tag{25}$$

where $D_{ii} = 1/\|x_i - F^t g_i\|$.

The proof of Lemma 5 is similar to the proof of Lemma 3 and thus is skipped due to space limitations.

PROOF. (Theorem 1(B)). From Lemma 4, the value of expression inside $[\,\cdot\,]$ in line Eq.(25) is negative or zero. Therefore

$$\|X - F^{t+1}G\|_{2,1} - \|X - F^tG\|_{2,1} \leq 0 \tag{26}$$

This proves that the objective function of Eq.(11) decreases monotonically. □

3.3 Proof of Lemma 2

PROOF. Eq.(19) can be re-expressed as

$$J(G^{t+1}) \leq J(G^t) \tag{27}$$

where

$$J(G) = \text{Tr}(X - FG)D(X - FG)^T \tag{28}$$

Lemma 2 states that under updating rule of Eq.(17), $J(G)$ monotonically decreases.

We prove Lemma 2 using the auxiliary function approach [13]. If a function satisfies

$$J(G) \leq Z(G, G'), \forall G' \tag{29}$$

$$Z(G, G) = J(G) \tag{30}$$

we say $Z(G, G')$ is an auxiliary function of $J(G)$. We define

$$G^{(t+1)} = \arg\min_G Z(G, G^{(t)}) \tag{31}$$

Then, we have

$$J(G^{(t+1)}) = Z(G^{(t+1)}, G^{(t+1)}) \leq Z(G^{(t+1)}, G^{(t)}) \leq J(G^{(t)})$$

This proves that $J(G^{(t)})$ monotonically decreases.

The key steps in the remainder of the proof are: (1) find an appropriate auxiliary function; (2) find the global maxima of the auxiliary function.

Now we show that an auxiliary function of $J(G)$ of Eq.(27) is

$$\begin{aligned} Z(G, G') &= \text{Tr}(XDX^T - 2G^TF^TXD) \\ &\quad + \sum_{k=1}^{K}\sum_{i=1}^{n} \frac{(F^TFG'D)_{ki}G_{ki}^2}{G'_{ki}} \end{aligned} \tag{32}$$

First, $J(G)$ of Eq.(28) can be expressed as

$$J(G) = \text{Tr}\left(XDX^T - 2G^TF^TXD + FGDG^TF^T\right). \tag{33}$$

We make use of the following matrix inequality [6]

$$\text{Tr}\,(H^TAHB) \leq \sum_{ik}(AH'B)_{ik}\frac{H_{ik}^2}{H'_{ik}} \tag{34}$$

where A, B, H are nonnegative matrices with appropriate sizes and $A = A^T,\ B = B^T$. The equality holds when $H = H'$.

In the inequality Eq.(34), setting $A = F^TF, B = D, H = G, H' = G'$, then the 3rd term of Eq.(33) is always smaller than the 3rd term of Eq.(32). The equality holds when $G = G'$. Thus $Z(G, G')$ of Eq.(32) is an auxiliary function of $J(G)$ of Eq.(33).

Now we need to find the global minima of Eq.(32). Let $f(G) = Z(G, G')$. The gradient of $f(G)$ is

$$\frac{\partial f(G)}{\partial G_{ki}} = -2(F^TXD)_{ki} + 2\frac{(F^TFG'D)_{ki}G_{ki}}{G'_{ki}} \tag{35}$$

The 2nd order derivatives (Hessian matrix) is

$$\frac{\partial^2 f(G)}{\partial G_{ki}\partial G_{lj}} = \left(2\frac{(F^TFG'D)_{ki}}{G'_{ki}}\right)\delta_{ij}\delta_{kl}$$

Therefore, the Hessian matrix $\frac{\partial^2 f(G)}{\partial G_{ki}\partial G_{lj}}$ is semi-positive definite. This implies function $f(G)$ is a convex function and there is a unique global minima for $f(G)$.

The global minima is obtained by setting the gradient of $f(G)$ to zero and solve for G. Thus we set Eq.(35) to zero and obtain

$$G_{ki} = G'_{ki}\frac{(F^T XD)_{ki}}{(F^T FG'D)_{ki}} \tag{36}$$

Noting $G^{(t+1)} \leftarrow G$ and $G^{(t)} \leftarrow G'$, the above equation recovers the updating rule of Eq.(17). Therefore under this updating rule, the objective function $J(G)$ of Eq.(28) decreases monotonically. □

3.4 Proof of Lemma 3

PROOF. First we note that

$$\text{Tr}(X - FG^t)D(X - FG^t)^T = \sum_{j=1}^{p}\sum_{i=1}^{n}(X - FG^t)_{ki}^2 D_{ii} \tag{37}$$

$$= \sum_{i=1}^{n}\|x_i - Fg_i^t\|^2 D_{ii} \tag{38}$$

Similarly,

$$\text{Tr}(X - FG^{t+1})D(X - FG^{t+1})^T = \sum_{i=1}^{n}\|x_i - Fg_i^{t+1}\|^2 D_{ii} \tag{39}$$

Thus the right-hand-size (RHS) of Eq.(21) becomes

$$RHS = \frac{1}{2}\sum_{i=1}^{n}\Big(\|x_i - Fg_i^{t+1}\|^2 D_{ii} - \|x_i - Fg_i^t\|^2 D_{ii}\Big) \tag{40}$$

$$= \frac{1}{2}\sum_{i=1}^{n}\Big(\|x_i - Fg_i^{t+1}\|^2 D_{ii} - \frac{1}{D_{ii}}\Big) \tag{41}$$

by using the definition $D_{ii} = 1/\|x_i - Fg_i^t\|$.

The left-hand-size (LHS) of Eq.(21) becomes

$$LHS = \sum_{i=1}^{n}\Big(\|x_i - Fg_i^{t+1}\| - \|x_i - Fg_i^t\|\Big) \tag{42}$$

$$= \sum_{i=1}^{n}\Big(\|x_i - Fg_i^{t+1}\| - \frac{1}{D_{ii}}\Big) \tag{43}$$

Therefore, we have

$$\begin{aligned}
&LHS - RHS \\
&= \sum_{i=1}^{n}\Big(\|x_i - Fg_i^{t+1}\| - \frac{1}{2}\|x_i - Fg_i^{t+1}\|^2 D_{ii} - \frac{1}{2D_{ii}}\Big) \\
&= \sum_{i=1}^{n}\frac{D_{ii}}{2}\Big(\frac{2\|x_i - Fg_i^{t+1}\|}{D_{ii}} - \|x_i - Fg_i^{t+1}\|^2 - \frac{1}{D_{ii}^2}\Big) \\
&= \sum_{i=1}^{n}\frac{-D_{ii}}{2}\Big(\|x_i - Fg_i^{t+1}\|^2 - 2\|x_i - Fg_i^{t+1}\|\frac{1}{D_{ii}} + \frac{1}{D_{ii}^2}\Big) \\
&= \sum_{i=1}^{n}\frac{-D_{ii}}{2}\Big(\|x_i - Fg_i^{t+1}\| - \frac{1}{D_{ii}}\Big)^2 \\
&\le 0
\end{aligned} \tag{44}$$

This competes the proof. □

4. CORRECTNESS OF THE ALGORITHM

In previous section, we proved that the objective function decreases monotonically under updating of Eqs.(16,17). Here we prove that the converged solution is the correct optimal solution, i.e., the converged solution satisfies the Karush-Kohn-Tucker condition of the constrained optimization theory.

First we have Theorem 6 to prove the correctness of the algorithm w.r.t. F, and then we have Theorem 7 to prove the correctness of the algorithm w.r.t. G.

THEOREM 6. *At convergence, the converged solution F^* of the updating rule of Eq.(16) satisfies the KKT condition of the optimization theory.*

PROOF. The KKT condition for G with the constraints $F_{jk} \ge 0,\ j = 1\cdots p,\ k = 1\cdots K$, is

$$\frac{\partial J(F)}{\partial F_{jk}}F_{jk} = 0,\ \forall\, j,k \tag{45}$$

The derivative is

$$\begin{aligned}
&\frac{\partial J(F)}{\partial F_{jk}} \\
&= \sum_{i=1}^{n}\frac{1}{\sqrt{\sum_{j'}(X - FG)_{j'i}^2}}\sum_{j''=1}^{p}(X - FG)_{j''i}\frac{\partial(X - FG)_{j''i}}{\partial F_{jk}} \\
&= \sum_{i=1}^{n}\frac{-1}{\sqrt{\sum_{j'}(X - FG)_{j'i}^2}}(X - FG)_{ji}G_{ki} \\
&= \sum_{i=1}^{n} -(X - FG)_{ji}G_{ki}D_{ii} \\
&= -(XDG^T)_{jk} + (FGDG^T)_{jk}
\end{aligned} \tag{46}$$

Thus the KKT condition for F is

$$[-(XDG^T)_{jk} + (FGDG^T)_{jk}]F_{jk} = 0,\ \forall\, j,k \tag{47}$$

On the other hand, once F converges, according to the updating rule of Eq.(16), the converged solution F^* satisfies

$$F_{jk}^* = F_{jk}^*\frac{(XDG^T)_{jk}}{(F^*GDG^T)_{jk}} \tag{48}$$

which can be written as

$$[-(XDG^T)_{jk} + (F^*GDG^T)_{jk}]F_{jk}^* = 0, \tag{49}$$

This is identical to Eq.(47). Thus the converged solution satisfies the KKT condition. □

THEOREM 7. *At convergence, the converged solution G^* of the updating rule of Eq.(17) satisfies the KKT condition of the optimization theory.*

PROOF. The KKT condition for G with the constraints $G_{ki} \ge 0,\ k = 1\cdots K,\ i = 1\cdots n$, is

$$\frac{\partial J(G)}{\partial G_{ki}}G_{ki} = 0,\ \forall\, k,i \tag{50}$$

The derivative is

$$\begin{aligned}
&\frac{\partial J(F)}{\partial G_{ki}} \\
&= \sum_{i'=1}^{n} \frac{1}{\sqrt{\sum_{j'}(X-FG)_{j'i}^2}} \sum_{j=1}^{p}(X-FG)_{ji'} \frac{\partial (X-FG)_{ji'}}{\partial G_{ki}} \\
&= \sum_{i'=1}^{n} \frac{-1}{\sqrt{\sum_{j'}(X-FG)_{j'i'}^2}} \sum_{j=1}^{p}(X-FG)_{ji} F_{jk}\delta_{ii'} \\
&= -\frac{[F^T(X-FG)]_{ki}}{\sqrt{\sum_{j'}(X-FG)_{j'i}^2}} \\
&= -[F^T(X-FG)]_{ki} D_{ii} \\
&= -[F^T(X-FG)D]_{ki} \qquad (51)
\end{aligned}$$

Thus the KKT condition for G is

$$[-(F^TXD)_{ki} + (F^TFGD)_{ki}]G_{ki} = 0, \ \forall\, k, i \qquad (52)$$

On the other hand, once G converges according to the updating rule of Eq.(17), the converged solution G^* satisfies

$$G^*_{ki} \Leftarrow G^*_{ki} \frac{(F^TXD)_{ki}}{(F^TFG^*D)_{ki}} \qquad (53)$$

which can be written as

$$[-(F^TXD)_{ki} + (F^TFG^*D)_{ki}]G^*_{ki} = 0, \qquad (54)$$

This is identical to Eq.(52). Thus the converged solution G^* satisfies the KKT condition. □

5. EXPERIMENTS

In this section, we apply the proposed $L_{2,1}$ NMF clustering algorithm to compare its performance with standard NMF algorithm and k-means algorithm. Extensive experiments are made on ten well known data sets.

5.1 Dataset description

We use 6 widely used image data sets and also 4 UCI[1] data sets. Table 1 summarizes the characteristics of those data sets.

AT&T[2]. There are totally 400 images belonging to 40 different subjects. For each subject, the images are in great varieties because of different taking time with changing lighting variance and facial expressions. All the pictures are taken with dark homogeneous background. The size of each cropped image is 112x92 pixels. We resize each image to 56x46 pixels in our evaluation.

MNIST. This hand-written digits data set consists of 70,000 digital images, which are from digit "0" to "9" [12]. We centralize each image to 28x28 pixels according to the center of the mass of the pixel intensities. We random select 15 images from each class, and there are totally 150 images in our evaluation data set.

UMIST. It is a face-recognition data set frequently used in computer vision field. It is challenge to recognize the faces in this data set because the variations for the face from the same class are larger than those in other face recognition problems. Each image is cropped to 28x23 pixels. There are totally 360 images with 20 different classes.

CMU PIE. It is a face data set containing 68 subjects with 41,368 face images. In the preprocessing step, we normalize the images(in scale and direction) to keep the two eyes are aligned at the same position. Each image is resized into 32x32 pixels. In our experiment, we random select 10 images from each class with different combinations of pose, face expression and illumination condition.

Yale. It is a data set obtained from the combinations of the original and extended Yale database [10]. There are totally 38 classes (10 subjects in original database with 28 subjects in the extended database) under 576 viewing conditions (9 poses with 64 different illumination conditions). We shrink each image by a factor of 0.25 to size 48x42. We select 64 images in different illumination conditions from 31 classes, and therefore there are totally 1984 images.

Bin-alpha. This data set is composed of 1404 binary images of handwritten digits from "0" to "9" and also characters from "A" to "Z", totally 36 classes. The resolution of each image is 20x16 pixels.

For all the image data sets, we use the same original space without making any changes and also the raw gray level values as features. The other four non-image data sets German, Car, Lenses, Vehicle are randomly selected from the UCI Repository. All of them only have non-negative values as features. Robust NMF can be applied to text, web and social network datasets for various applications.

5.2 Measurement

The evaluation metrics [16, 18] we used here are clustering accuracy, normalized mutual information and purity. These measurement are widely used in the evaluation of different clustering approaches.

Clustering accuracy(ACC) is defined as,

$$ACC = \frac{\sum_{i=1}^{n} \delta(l_i, map(c_i))}{n}, \qquad (55)$$

where l_i is the true class label and c_i is the obtained cluster label of x_i, map(.) is the best mapping function, $\delta(x, y)$ is the delta function where $\delta(x, y) = 1$ if $x = y$, and $\delta(x, y) = 0$ otherwise. The mapping function map(.) matches the true class label and the obtained clustering label, where the best mapping is solved by Hungarian algorithm. A large ACC value indicates a better clustering performance.

Normalized mutual information (NMI) is used to evaluate the clustering quality from information point, and defined by normalization on the mutual information between the cluster assignments and the pre-existing input labeling of the classes. The normalization used is the average of the entropy of the cluster assignment and that of pre-existing input labeling. More formally,

$$NMI = \frac{I(S, C)}{(H(S) + H(C))/2}, \qquad (56)$$

where $C = \{c_1, c_2, ..., c_k\}$ is the pre-existing classes, $S = \{S_1, S_2, ..., S_k\}$ is a particular clustering result, $I(S, C)$ is the mutual information of clustering assignment with pre-existing class labels, and $H(S)$ is the entropy for the clustering assignment. A larger NMI valuealso indicates a better clustering solution.

Purity measures the extent to which each cluster contained data points from primarily one class. The purity of a clustering is obtained by the weighted sum of individual cluster purity values, given as,

$$Purity = \sum\nolimits_{i=1}^{k} \frac{n_i}{n} P(S_i), P(S_i) = \frac{1}{n_i} \max\nolimits_j (n_i^j). \qquad (57)$$

where S_i is a particular cluster size of n_i, n_i^j is the number of the i-th input class that were assigned to the j-th cluster. k is the number of the clusters and n is the total number of the data points. A large Purity value indicates a good clustering solution.

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

(a) UMIST (b) AT&T

Figure 5: Demonstration of the object function and the number of iterations needed before convergence of $L_{2,1}$ NMF and NMF on data sets UMIST and AT&T. The object functions we demonstrate are $log(||X - FG||_{2,1})$ and $log(||X - FG||_F)$. The convergence criteria is described in Eq. (58).

Table 1: Description of dataset

Dataset	#Size	#Dimensions	#Class
AT&T	400	2576	40
MNIST	150	784	10
CMUPIE	680	1024	68
UMIST	360	644	20
YALE	1984	2016	31
Bin-alph	1404	320	36
German	1000	20	2
Cars	392	8	3
WINE	178	13	3
Vehicle	846	18	4

Table 2: Comparison on the object functions and time cost on data sets, UMIST and AT&T. We present the normalized convergence error (NCE), i.e., $\frac{||X-FG||_{2,1}}{||X||_{2,1}}$ and $\frac{||X-FG||_F}{||X||_F}$. *Iteration* represents the time of iterations needed before converges according to the convergence criteria of Eq.(58).

Dataset	Metric	$L_{2,1}$NMF	NMF
UMIST	NCE	0.1598	0.1721
	Time(sec.)	73.6396	52.6239
	Iteration	3906	4932
AT&T	NCE	0.1185	0.1375
	Time(sec.)	121.5217	96.0741
	Iteration	4029	4783

5.3 Convergence analysis for $L_{2,1}$ NMF

We make convergence analysis on real world image data sets. We reshape each image into one vector, and then form a large matrix X, where each column is a linearized vector from an image. For example, AT&T data set has 400 images, where the size of each image is 56x46 pixels, thus X is constructed with size 2576x400. With the same input data X, we random generate the same F and G to feed into NMF model and $L_{2,1}$ NMF model. We compare the convergence properties (e.g., speed, object function values) of $L_{2,1}$ NMF with standard NMF.

We test convergence in each iteration by computing the object function values. The convergence criterion we used is

$$\frac{J_{t+1} - J_t}{J_t} < 10^{-7}, \tag{58}$$

where J_t is the object function values ($||X - FG||_F$ for NMF and $||X - FG||_{2,1}$ for robust NMF) in the t-th iteration in the convergence tests. For the fairness of comparisons, we use the non-squared object function $||X - FG||_F$ in standard NMF instead of the squared object function $||X - FG||^2$.

We implement our algorithm in Matlab 7.0. All the experiments are done on a AMD Phenom(tm) 2.80GHZ machine with 6GB memory running Windows 7. The initial object functions are computed through the random guess of F and G. Fig.(5) shows the log object functions and also the number of iterations needed for the convergence of NMF and $L_{2,1}$ NMF on the data sets UMIST and AT&T. Because data matrix X can be very large, for the convenience of demonstration, here we show the log values of the object functions, i.e., $log(||X - FG||_F)$ and $log(||X - FG||_{2,1})$.

Our experiment results show that on data set UMIST, $L_{2,1}$ NMF needs 3906 iterations before convergence while NMF needs 4932 iterations before convergence. However, the computation time cost on $L_{2,1}$ NMF is higher than NMF due to the updating rule on the diagonal matrix D. As is shown in Table 2, both on UMIST and AT&T data sets, $L_{2,1}$ NMF takes less iterations to converge than NMF, yet with more computation time.

Also, the object function of $L_{2,1}$ NMF is always larger than non-squared NMF object function due to the formulation of $L_{2,1}$ norm defined in Eq.(10). It is worth mentioning that the larger object function values do not necessarily mean the worse matrix factorization results. As is shown in Table 2, $L_{2,1}$ NMF has lower normalized convergence error compared with NMF on those two testing data, which shows the effectiveness of $L_{2,1}$ NMF.

5.4 Clustering Results

We report clustering results by making comparisons with K-means clustering and standard NMF clustering approach. For the initializations of F and G, we did not use the random generated matrices. Firstly, we use the principal component analysis to get a subspace with r-dimension. After this, k-means clustering approach is employed on the projection data to get clustering results. We use above clustering results G' to initialize $G = G' + 0.3$, and then F are

Table 3: Clustering quality comparison of $L_{2,1}$ NMF with NMF and k-means on 10 data sets.

Dataset	Metric	Approaches		
		$L_{2,1}$-NMF	NMF	k-means
AT&T	ACC	**0.6808**	0.6496	0.6519
	NMI	**0.8206**	0.7945	0.8134
	PUR	**0.7210**	0.6822	0.7021
MNIST	ACC	**0.7719**	0.7297	0.6872
	NMI	**0.7429**	0.6966	0.6788
	PUR	**0.7849**	0.7461	0.7068
UMIST	ACC	**0.5176**	0.4861	0.4744
	NMI	**0.6174**	0.5869	0.6030
	PUR	**0.5306**	0.5029	0.5185
CMUPIE	ACC	**0.4321**	0.4138	0.2227
	NMI	**0.6782**	0.6557	0.5386
	PUR	**0.4562**	0.4377	0.2429
YALE	ACC	**0.2117**	0.1950	0.0870
	NMI	**0.3114**	0.2882	0.0933
	PUR	**0.2238**	0.2082	0.0943
Bin-alpha	ACC	**0.3594**	0.3283	0.3342
	NMI	**0.5621**	0.4987	0.5072
	PUR	**0.4138**	0.3791	0.3897
German	ACC	**0.6670**	0.6431	0.6308
	NMI	**0.4621**	0.4306	0.4275
	PUR	**0.7000**	0.7000	0.7000
Cars	ACC	**0.6171**	0.5436	0.4617
	NMI	**0.4394**	0.3837	0.2989
	PUR	**0.6720**	0.6406	0.6356
WINE	ACC	**0.8764**	0.8371	0.7138
	NMI	**0.6373**	0.5619	0.4268
	PUR	**0.8764**	0.8371	0.7138
Vehicle	ACC	**0.4812**	0.4697	0.4456
	NMI	**0.3987**	0.3767	0.3429
	PUR	**0.4934**	0.4719	0.4456

obtained by computing the clustering centroid for each category. Empirically, we run k-means 10 times during initializations of G.

Clustering Analysis on Confusion Matrices. Firstly, we show the confusion matrices constructed from the clustering analysis. The diagonals of the matrices show the number of data points that are clustered into the default subject clusters. The number of data with correct clustering labels can be directly computed by summing over all the diagonals in each matrix. Due to space limit, we did not show all the confusion matrices on all data sets. Figs. (6-9) show the confusion matrices on data sets UMIST, YALE, MNIST, PIE. It is easy to see that the diagonals of confusion matrices in $L_{2,1}$ NMF are much stronger than those of standard NMF approach.

Clustering Results Analysis. Table 3 summarizes the clustering results of our approach by making comparisons with standard NMF and k-means clustering. The metrics we used here are clustering accuracy(ACC), normalized mutual information(NMI) and purity(PUR). Extensive experiments are made on 10 data sets, 6 of which are image data sets and the other 4 are from UCI data sets. The experiments can also be easily conducted on other text/web data sets. On each dataset, the clustering number is set to the real number of classes in ground truth (e.g., for data set AT&T, K = 40; for data set MNIST, K = 10). For fairness, we use the same generated F and G to compare the performance of NMF and $L_{2,1}$ NMF in each round. k-means algorithm is run 10 times to get the clustering results with the least object function values. From Table 3, we can see, (1). $L_{2,1}$ NMF approach performs better than standard NMF approach on all data sets. (2). $L_{2,1}$ NMF outperforms k-means clustering on all data sets.

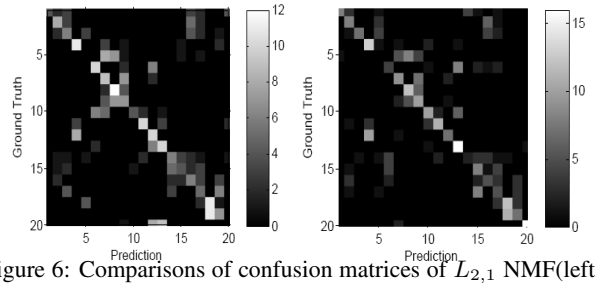

Figure 6: Comparisons of confusion matrices of $L_{2,1}$ NMF(left in each panel) and NMF(right in each panel) on dataset UMIST. Each column of a matrix represents the instances in a predicted class, and each row represents the instances in an actual class.

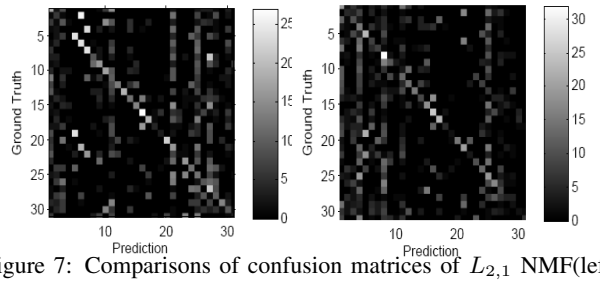

Figure 7: Comparisons of confusion matrices of $L_{2,1}$ NMF(left) and NMF(right) on dataset Yale.

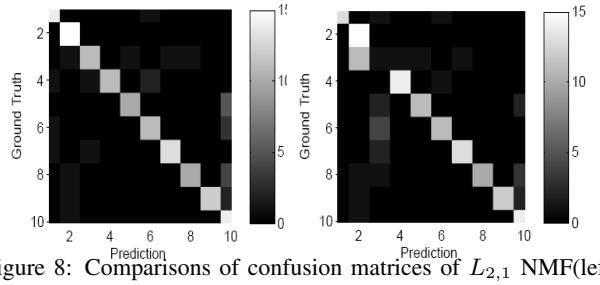

Figure 8: Comparisons of confusion matrices of $L_{2,1}$ NMF(left) and NMF(right) on dataset MNIST.

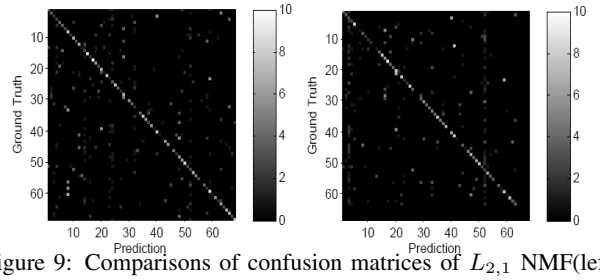

Figure 9: Comparisons of confusion matrices of $L_{2,1}$ NMF(left) and NMF(right) on dataset PIE.

6. EXTENSION TO L_1-NMF

Here we briefly discuss the extension of NMF to another form of robust NMF: L_1-NMF. The most robust formulation of the error function is

$$\|X - FG\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{p}|(X - FG)_{ji}|. \tag{59}$$

For computational reasons, we replace $|(X - FG)_{ij}|$ by $\left((X - FG)_{ij}^2 + \epsilon^2\right)^{1/2}$. Here ϵ is set to a very small number. Thus we minimize the objective function

$$J_1 = \|X - FG\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{p}((X - FG)_{ji}^2 + \epsilon^2)^{1/2}. \quad (60)$$

Formally, L_1-NMF is formulated as

$$\min_{F,G} \; J_1(F,G) \;\; s.t. \;\; F \geq 0, \; G \geq 0. \quad (61)$$

Extending our approach in deriving the computational algorithm for $L_{2,1}$-NMF, we can derive the following updating algorithms for L_1-NMF:

$$F_{jk} \Leftarrow F_{jk}\frac{[X \circ WG^T]_{jk}}{[(FG) \circ WG^T]_{jk}}, \quad (62)$$

$$G_{ki} \Leftarrow G_{ki}\frac{[F^T X \circ W]_{ki}}{[F^T(FG) \circ W]_{ki}}, \quad (63)$$

where W is a matrix given by $W_{ij} = \left((X - FG)_{ij}^2 + \epsilon^2\right)^{-1/2}$, and $\circ$ is the Hadamard product, i.e., elementwise product between two matrices. Here we assume Hadamard product has higher operator precedence over regular matrix product, i.e., $AB \circ CD = A(B \circ C)D$. Convergence property and correctness analysis can be similarly established. Details will be published in a forthcoming paper.

A striking feature of the L_1-NMF updating algorithm [Eqs.(62-63)] is its nearly-identical forms the $L_{2,1}$-NMF algorithm [Eqs.(16,17)] if we replace D by W; they are also nearly-identical to the algorithm [Eqs.(2,3)] of standard NMF. Analysis of this unified NMF algorithmic framework will be presented in the forthcoming paper.

7. CONCLUSION

In this paper, we propose a robust formulation of NMF using L21-norm. We also derive a computational algorithm with rigorous convergence analysis. Experiments on 10 datasets show that the robust NMF provides more faithful basis factors and consistently better clustering results as compared to standard NMF.

Acknowledgements. This work is partially supported by NSF-CCF-0939187, NSF-CCF-0917274, NSF-DMS-15228.

8. REFERENCES

[1] J.-P. Brunet, P. Tamayo, T. Golub, and J. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proc. Nat'l Academy of Sciences USA*, 102(12):4164–4169, 2004.

[2] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *ICDM*, pages 63–72, 2008.

[3] M. Cooper and J. Foote. Summarizing video using non-negative similarity matrix factorization. In *Proc. IEEE Workshop on Multimedia Signal Processing*, pages 25–28, 2002.

[4] I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in Neural Information Processing Systems 17*, Cambridge, MA, 2005. MIT Press.

[5] C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. *Proc. SIAM Data Mining Conf*, 2005.

[6] C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2010. (LBNL Tech Report 60428, 2006).

[7] C. Ding, T. Li, and W. Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method. *Proc. National Conf. Artificial Intelligence*, 2006.

[8] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of ACM SIGKDD*, pages 126–135, 2006.

[9] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization (pdf file). *Proc. Int'l Conf. Machine Learning (ICML)*, June 2006.

[10] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *PAMI*, 23:643–660, 2001.

[11] Q. Gu and J. Zhou. Co-clustering on manifolds. In *KDD*, pages 359–368, 2009.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[13] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2000.

[14] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *CVPR*, pages 207–212, 2001.

[15] T. Li, C. Ding, and M. I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *ICDM*, pages 577–582, 2007.

[16] F. Nie, C. H. Q. Ding, D. Luo, and H. Huang. Improved minmax cut graph clustering with nonnegative relaxation. In *ECML/PKDD*, pages 451–466, 2010.

[17] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l2,1-norms minimization. *NIPS*, 2010.

[18] F. Nie, D. Xu, I. W. Tsang, and C. Zhang. Spectral embedded clustering. In *IJCAI*, pages 1181–1186, 2009.

[19] P. Paatero and U. Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111–126, 1994.

[20] V. P. Pauca, F. Shahnaz, M. Berry, and R. Plemmons. Text mining using non-negative matrix factorization. In *Proc. SIAM Int'l conf on Data Mining*, pages 452–456, 2004.

[21] F. Sha, L. K. Saul, and D. D. Lee. Multiplicative updates for nonnegative quadratic programming in support vector machines. In *Advances in Neural Information Processing Systems 15*. MIT Press, Cambridge, MA, 2003.

[22] N. Srebro, J. Rennie, and T. Jaakkola. Maximum margin matrix factorization. In *Advances in Neural Information Processing Systems*, Cambridge, MA, 2005. MIT Press.

[23] F. Wang, T. Li, and C. Zhang. Semi-supervised clustering via matrix factorization. In *SDM*, pages 1–12, 2008.

[24] N. Z. Weixiang Liu and Q. You. Nonnegative matrix factorization and its applications in pattern recognition. *Chinese Science Bulletin*, 51(1):7–18, 2006.

[25] Y.-L. Xie, P. Hopke, and P. Paatero. Positive matrix factorization applied to a curve resolution problem. *Journal of Chemometrics*, 12(6):357–364, 1999.