TWO DIMENSIONAL NONNEGATIVE MATRIX FACTORIZATION

Quanquan Gu and Jie Zhou

State Key Laboratory on Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology(TNList)
Department of Automation, Tsinghua University, Beijing 100084, China
gqq03@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn

ABSTRACT

Nonnegative Matrix Factorization (NMF) has been widely used in computer vision and pattern recognition. It aims to find two nonnegative matrices whose product can well approximate the original matrix, which naturally leads to parts-based representation. In this paper, we propose a Two Dimensional Nonnegative Matrix Factorization (2DNMF), specifically for a sequence of matrices. In contrast to NMF which applies for only a single matrix, and finds only one base matrix, 2DNMF aims to find two base matrices to represent the input matrices in a low dimensional matrix subspace. It not only inherits the advantages of NMF, but also owns the properties low computational complexity, as well as high recognition accuracy. Experiments on benchmark image recognition data sets illustrate that the proposed method is very effective and efficient.

Index Terms— Two Dimensional, Nonnegative Matrix Factorization, Feature Extraction

1. INTRODUCTION

Dimensionality reduction is an important topic in computer vision and pattern recognition. In the past decades, many dimensionality reduction methods have been proposed, e.g. *Principal Component Analysis* (PCA) [1]. Recent years, nonnegative and sparse representations have witnessed increasing interest, in which *Nonnegative Matrix Factorization* (NMF) [2] is one of the most representative works. NMF aims to find two nonnegative matrices whose product can well approximate the original matrix, which naturally leads to parts-based representation.

The methods discussed above are all based on vector data. However, many real world data, e.g. image, is usually represented by matrix. Conventional treatment to this kind of data is to vectorize each matrix to a vector, and combine them column by column to form a single large matrix, then traditional dimensionality reduction is applied. However, when the data matrix is transformed into a vector, the data is usually represented in a very high dimensional feature space, which may result in the *curse of dimensionality*. Furthermore, the intrinsic spatial structure in the data matrices will be lost. Several works have been done to generalize PCA to apply for matrices [3] [4]. As to NMF, [5] proposed a nonnegative matrix set factorization (NMSF), which finds one common base matrix for all the input matrices.

In this paper, we propose a *Two Dimensional Nonnegative Matrices Factorization* (2DNMF), specifically for a sequence of matrices. In contrast to NMF which applies for only a single matrix, and finds only one base matrix, 2DNMF aims to find two base matrices to represent the input matrices in a low dimensional matrix subspace. It not only inherits the advantages of NMF, but also owns the properties of low computational complexity, as well as high recognition accuracy. Compared with NMSF which finds only one common base matrix, 2DNMF pursues two common base matrices, which is more natural and reasonable for matrices and can explicitly give the bases [4]. We will show that 2DNMF can be optimized in an iterative way, and its convergence is theoretically guaranteed. Both theoretical analysis and empirical experiments on benchmark image recognition data sets illustrate that the proposed method is very effective and efficient.

The remainder of this paper is organized as follows. In Section 2, we briefly review NMF. In Section 3, we present 2DNMF and its optimization algorithm, followed with theoretical analysis. The experiments on benchmark image recognition databases are demonstrated in Section 4. Finally, we draw a conclusion in Section 5.

This work was supported by Natural Science Foundation of China under grant 60673106 and 60573062.

2. A BRIEF REVIEW OF NMF

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}_+^{d \times N}$, each column of $\mathbf{X}$ is a data point. NMF aims to find two nonnegative matrices $\mathbf{U} \in \mathbb{R}_+^{d \times m}$ and $\mathbf{V} \in \mathbb{R}_+^{N \times m}$ which minimize the following objective

$$\begin{aligned} J_{NMF} &= ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||_F^2, \\ &\text{s.t.} \quad \mathbf{U} \geq 0, \mathbf{V} \geq 0, \end{aligned} \tag{1}$$

where $||\cdot||_F$ is Frobenius norm. We usually call $\mathbf{U}$ as base matrix, and $\mathbf{V}$ as coefficient matrix. To optimize the objective in Eq.(1), [6] presented an iterative update algorithm as follows

$$\begin{aligned} \mathbf{U}_{ij} &\leftarrow \mathbf{U}_{ij} \frac{(\mathbf{X}\mathbf{V})_{ij}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ij}} \\ \mathbf{V}_{ij} &\leftarrow \mathbf{V}_{ij} \frac{(\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ij}} \end{aligned} \tag{2}$$

3. TWO DIMENSIONAL NONNEGATIVE MATRIX FACTORIZATION

In this section, we will present 2DNMF for the data represented by a sequence of matrices.

3.1. Problem Formulation

Given a sequence of matrices, i.e. $\{\mathbf{X}_n\}_{n=1}^N \in \mathbb{R}_+^{r \times c}$, 2DNMF aims to find two nonnegative transformation matrices $\mathbf{U} \in \mathbb{R}_+^{r \times l_1}$ and $\mathbf{V} \in \mathbb{R}_+^{c \times l_2}$, and N nonnegative matrices $\mathbf{D}_n \in \mathbb{R}_+^{l_1 \times l_2}$, such that

$\mathbf{U}\mathbf{D}_n\mathbf{V}^T$ is a good approximation of $\mathbf{X}_n$, for all n. Mathematically, we can formulate this as the following minimization problem

$$\begin{aligned} J_{2DNMF} &= \sum_{n=1}^{N} ||\mathbf{X}_n - \mathbf{U}\mathbf{D}_n\mathbf{V}^T||_F^2, \\ &\text{s.t.} \quad \mathbf{U} \geq 0, \mathbf{D}_n \geq 0, \mathbf{V} \geq 0, \forall n, \end{aligned} \tag{3}$$

where $||\cdot||_F$ is Frobenius norm.

3.2. Optimization

In the following, we will give the solution to Eq.(3).

Since $\mathbf{U} \geq 0, \mathbf{V} \geq 0$ and $\mathbf{D}_n \geq 0$, we introduce the Lagrangian multiplier $\alpha \in \mathbb{R}^{r\times l1}, \beta \in \mathbb{R}^{c\times l2}$ and $\gamma \in \mathbb{R}^{l1\times l2}$, thus, the Lagrangian function is

$$\begin{aligned} L(\mathbf{U},\mathbf{V},\mathbf{D}_n) &= \sum_{n=1}^{N} ||\mathbf{X}_n - \mathbf{U}\mathbf{D}_n\mathbf{V}^T||_F^2 \\ &- \text{tr}(\alpha\mathbf{U}^T) - \text{tr}(\beta\mathbf{V}^T) - \text{tr}(\gamma\mathbf{D}_n^T) \end{aligned}$$

Setting $\frac{\partial L}{\partial \mathbf{U}} = 0$, $\frac{\partial L}{\partial \mathbf{V}} = 0$ and $\frac{\partial L}{\partial \mathbf{D}_n} = 0$, we obtain

$$\begin{aligned} \alpha &= \sum_{n=1}^{N}(-2\mathbf{X}_n\mathbf{V}\mathbf{D}_n^T + 2\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T) \\ \beta &= \sum_{n=1}^{N}(-2\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n + 2\mathbf{V}\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n) \\ \gamma &= -2\mathbf{U}^T\mathbf{X}_n\mathbf{V} + 2\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V} \end{aligned}$$

Using the Karush-Kuhn-Tucker condition [7], $\alpha_{ij}\mathbf{U}_{ij} = 0, \beta_{ij}\mathbf{V}_{ij} = 0$ and $\gamma_{ij}(\mathbf{D}_n)_{ij} = 0$, we get

$$\begin{aligned} [\sum_{n=1}^{N}(-\mathbf{X}_n\mathbf{V}\mathbf{D}_n^T + \mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T)]_{ij}\mathbf{U}_{ij} &= 0 \\ (\sum_{n=1}^{N}(-\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n + \mathbf{V}\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n))_{ij}\mathbf{U}_{ij} &= 0 \\ (-\mathbf{U}^T\mathbf{X}_n\mathbf{V} + \mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V})_{ij}\mathbf{U}_{ij} &= 0 \end{aligned} \tag{4}$$

Eq.(4) leads to the following updating formulas

$$\begin{aligned} \mathbf{U}_{ij} &\leftarrow \mathbf{U}_{ij}\sqrt{\frac{[\sum_{n=1}^{N}(\mathbf{X}_n\mathbf{V}\mathbf{D}_n^T)]_{ij}}{[\sum_{n=1}^{N}(\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T)]_{ij}}} \\ \mathbf{V}_{ij} &\leftarrow \mathbf{V}_{ij}\sqrt{\frac{[\sum_{n=1}^{N}(\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n)]_{ij}}{[\sum_{n=1}^{N}(\mathbf{V}\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n)]_{ij}}} \\ (\mathbf{D}_n)_{ij} &\leftarrow (\mathbf{D}_n)_{ij}\sqrt{\frac{[\mathbf{U}^T\mathbf{X}_n\mathbf{V}]_{ij}}{[\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}]_{ij}}} \end{aligned} \tag{5}$$

3.3. Convergence Analysis

In this subsection, we will investigate the convergence of the updating formulas in Eq.(5). We use the auxiliary function approach [6] to prove the convergence. Here we first introduce the definition of auxiliary function [6].

Definition 3.1. *[6] $Z(h,h')$ is an auxiliary function for $F(h)$ if the conditions*

$$Z(h,h') \geq F(h), Z(h,h) = F(h),$$

are satisfied.

Lemma 3.2. *[6] If Z is an auxiliary function for F, then F is non-increasing under the update*

$$h^{(t+1)} = \arg\min_h Z(h, h^{(t)})$$

Proof. $F(h^{(t+1)}) \leq Z(h^{(t+1)}, h^{(t)}) \leq Z(h^{(t)}, h^{(t)}) = F(h^{(t)})$ □

Theorem 3.3. *Let*

$$\begin{aligned} J(\mathbf{U}) &= \sum_{n=1}^{N} tr(-2\mathbf{D}_n\mathbf{V}^T\mathbf{X}_n^T\mathbf{U} + \mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T\mathbf{U}^T) \\ &= tr(-2\mathbf{E}\mathbf{U} + \mathbf{U}\mathbf{F}\mathbf{U}^T) \end{aligned}$$

where $\mathbf{E} = \sum_{n=1}^{N}(\mathbf{D}_n\mathbf{V}^T\mathbf{X}_n^T)$ and $\mathbf{F} = \sum_{n=1}^{N}(\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T)$. Then the following function

$$Z(\mathbf{U},\mathbf{U}') = -2\sum_{ij}(\mathbf{E}^T)_{ij}\mathbf{U}'_{ij}(1 + \log\frac{\mathbf{U}_{ij}}{\mathbf{U}'_{ij}}) + \sum_{ij}\frac{(\mathbf{U}'\mathbf{F})_{ij}\mathbf{U}_{ij}^2}{\mathbf{U}'_{ij}}$$

is an auxiliary function for $J(\mathbf{U})$. Furthermore, it is a convex function in $\mathbf{U}$ and its global minimum is

$$\mathbf{U}_{ij} = \mathbf{U}_{ij}\sqrt{\frac{[\sum_{n=1}^{N}(\mathbf{X}_n\mathbf{V}\mathbf{D}_n^T)]_{ij}}{[\sum_{n=1}^{N}(\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}\mathbf{D}_n^T)]_{ij}}} \tag{6}$$

Proof. For the limit of space, we omit it here. □

Theorem 3.4. *Updating $\mathbf{U}$ using Eq.(5) will monotonically decrease the value of the objective in Eq.(3), hence it converges.*

Proof. By Lemma 3.2 and Theorem 3.3, we can get that $J(\mathbf{U}^0) = Z(\mathbf{U}^0,\mathbf{U}^0) \geq Z(\mathbf{U}^1,\mathbf{U}^0) \geq J(\mathbf{U}^1) \geq \ldots$ So $J(\mathbf{U})$ is monotonically decreasing. Since $J(\mathbf{U})$ is obviously bounded below, we prove this theorem. □

Theorem 3.5. *Let*

$$\begin{aligned} J(\mathbf{V}) &= \sum_{n=1}^{N} tr(-2\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T + \mathbf{V}\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T) \\ &= tr(-2\mathbf{G}\mathbf{V}^T + \mathbf{V}\mathbf{H}\mathbf{V}^T) \end{aligned}$$

where $\mathbf{G} = \sum_{n=1}^{N}(\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n)$ and $\mathbf{H} = \sum_{n=1}^{N}(\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n)$ Then the following function

$$Z(\mathbf{V},\mathbf{V}') = -2\sum_{ij}\mathbf{G}_{ij}\mathbf{V}'_{ij}(1 + \log\frac{\mathbf{V}_{ij}}{\mathbf{V}'_{ij}}) + \sum_{ij}\frac{(\mathbf{V}'\mathbf{H})_{ij}\mathbf{V}_{ij}^2}{\mathbf{V}'_{ij}}$$

is an auxiliary function for $J(\mathbf{V})$. Furthermore, it is a convex function in $\mathbf{V}$ and its global minimum is

$$\mathbf{V}_{ij} = \mathbf{V}_{ij}\sqrt{\frac{[\sum_{n=1}^{N}(\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n)]_{ij}}{[\sum_{n=1}^{N}(\mathbf{V}\mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n)]_{ij}}} \tag{7}$$

Proof. For the limit of space, we omit it here. □

Theorem 3.6. *Updating $\mathbf{V}$ using Eq.(5) will monotonically decrease the value of the objective in Eq.(3), hence it converges.*

Proof. It is easy to prove by Lemma 3.2 and Theorem 3.5, hence we omit it here. □

Theorem 3.7. *Let*

$$J(\mathbf{D}_n) = tr(-2\mathbf{V}^T\mathbf{X}_n^T\mathbf{U}\mathbf{D}_n + \mathbf{D}_n^T\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V}) \quad (8)$$

Then the following function

$$\begin{aligned} Z(\mathbf{D}_n, \mathbf{D}'_n) &= -2\sum_{ij}(\mathbf{U}^T\mathbf{X}_n\mathbf{V})_{ij}(\mathbf{D}_n)'_{ij}(1+\log\frac{(\mathbf{D}_n)_{ij}}{(\mathbf{D}_n)'_{ij}}) \\ &+ \sum_{ij}\frac{(\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V})_{ij}(\mathbf{D}_n)^2_{ij}}{(\mathbf{D}_n)'_{ij}} \end{aligned}$$

is an auxiliary function for $J(\mathbf{D}_n)$. Furthermore, it is a convex function in $\mathbf{D}_n$ and its global minimum is

$$(\mathbf{D}_n)_{ij} = (\mathbf{D}_n)_{ij}\sqrt{\frac{(\mathbf{U}^T\mathbf{X}_n\mathbf{V})_{ij}}{(\mathbf{U}^T\mathbf{U}\mathbf{D}_n\mathbf{V}^T\mathbf{V})_{ij}}} \quad (9)$$

Proof. For the limit of space, we omit it here. □

Theorem 3.8. *Updating $\mathbf{D}_n$ using Eq.(5) will monotonically decrease the value of the objective in Eq.(3), hence it converges.*

Proof. It is easy to prove by Lemma 3.2 and Theorem 3.7, hence we omit it here. □

Note that there is no guarantee that Updating $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{D}_n$ using Eq.(5) will converge to global optima.

3.4. Computational Complexity Analysis

In this subsection, we will analyze the computational complexity of 2DNMF, compared with NMF.

For 2DNMF, the total time complexity is $O(t(3Nrc+5Ncl_1+(5N+3)cl_2+(6N+3)rl_1+2Nrl_2+7Nl_1l_2+Nl_1^2+Nl_2^2))$ where t is the number of iterations.

In contrast, the total time complexity of NMF is $O(t(7rcm+7Nm+4Nrc))$.

To give a concrete case study, we take the ORL data set with $p = 3$ images for each individual for example, hence $N = 120$. We set $l_1 = l_2 = 20$ for 2DNMF, and $m = 400$ for NMF. Then the time complexity of 2DNMF is $O(2,186,880t)$, while the time complexity of NMF is $O(3,694,720t)$. As a result, 2DNMF is more efficient than NMF.

4. EXPERIMENTS

In this section, we will investigate the performance of the proposed method for image recognition. We compare 2DNMF with PCA, GLRMA [4], NMF and NMSF [5].

4.1. Data Sets

In our experiments, we use three standard image recognition databases which are widely used as bench mark data sets in feature extraction literature.

The ORL face database[1]. There are ten images for each of the 40 human subjects, which were taken at different times, varying the lightings, facial expressions and facial details. The original images (with 256 gray levels) have size 92×112, which are resized to 32×32 for efficiency;

The UMIST face database is a multiview database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views. In our experiments, the images were also resized to 32×32;

The Coil20 data set[2] contains 32×32 gray scale images of 20 3D objects viewed from varying angles, at the interval of 5 degrees, resulting 72 images per object. The original images are resized to 32×32 for efficiency.

4.2. Parameter Settings

For each data set, we randomly divide it into training and testing sets. In detail, for each individual in the ORL and UMIST data sets, $p = 2, 3, 4$ images were randomly selected as training samples, and the rest were used for testing, while for each individual in the Coil20 data set, $p = 4, 6, 8$ images were randomly selected as training samples. We use the images in the training set to learn a subspace, and the recognition was performed in the subspace by Nearest Neighbor (NN) Classifier. Since the training set was randomly chosen, we repeated each experiment 20 times and calculated the average recognition accuracy. In general, the recognition rate varies with the dimensionality of the subspace. The best average performance obtained as well as the corresponding dimensionality is reported.

The parameter l_1 and l_2 in GLRAM and 2DNMF as well as the dimensionality in NMSF are set to the same value, denoted by l, in all experiments, for simplicity, which is set by searching the grid $\{1, 2, \ldots, 20\}$. Correspondingly, the parameter m in PCA and NMF is set by the grid $\{1^2, 2^2, \ldots, 20^2\}$ to obtain the same size of base image.

4.3. Convergence

In this subsection, we will examine the convergence of 2DNMF. In Fig. 1, we plot the objective function value in Eq.(3) with respect to the number of iterations on the three data sets.

From Fig. 1, we can empirically verify that the updating formulas in Eq.(5) indeed converge, which is consistent with the theoretical analysis in 3.3.

4.4. Recognition Capability

Given a testing image $\mathbf{X}_t \in \mathbb{R}^{r\times c}$, the projection is computed as follows.

For NMF, the projection is computed as $\mathbf{U}^\dagger \text{vec}(\mathbf{X}_t)$ where $\mathbf{U}^\dagger = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T$ and vec$(\cdot)$ is the vectorization function.

For NMSF, the projection is computed as $\mathbf{U}^\dagger\mathbf{X}_t$.

For 2DNMF, the projection is computed as $\mathbf{U}^\dagger\mathbf{X}_t\mathbf{V}$.

Table 1 shows the experimental results of all the methods on the three databases, where the value in each entry represents the average recognition accuracy of 20 independent trials, and the number in brackets is the corresponding l for GLRAM, NMSF and 2DNMF, and $\sqrt{m}$ for PCA, NMF.

It is clear that our method outperforms the other dimensionality reduction methods significantly on all the three data sets.

4.5. Computational Time

In this subsection, we compare the computational efficiency of NMF and 2DNMF. We plot the average training time of 20 independent trials of NMF and 2DNMF on the three data sets with incremental training samples in Fig. 2.

[1] http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data

[2] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

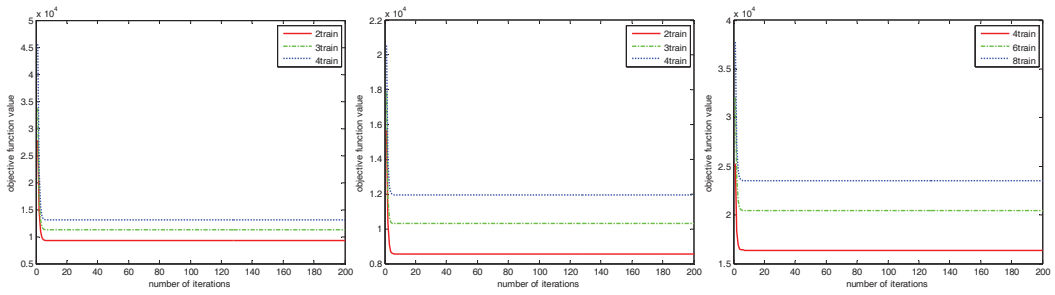

Fig. 1. The objective function value with respect to the number of iterations on the three data sets.

Table 1. Image Recognition accuracy of different algorithms on the three data sets. The number in brackets is the corresponding l for GLRAM, NMSF and 2DNMF, and $\sqrt{m}$ for PCA, NMF.

Data Set	ORL			UMIST			Coil20		
Method	2 Train	3 Train	4 Train	2 Train	3 Train	4 Train	4 Train	6 Train	8 Train
PCA	70.67(9)	78.88(11)	84.12(13)	61.40(7)	73.05(8)	77.96(9)	82.12(4)	86.78(4)	89.09(4)
GLRAM	71.28(20)	79.79(11)	84.77(16)	64.96(4)	77.47(4)	82.26(4)	82.39(6)	87.08(6)	89.59(6)
NMF	69.78(19)	78.27(6)	83.85(6)	57.05(9)	67.54(6)	74.03(6)	79.01(4)	85.09(4)	88.29(4)
NMSF	70.55(5)	80.89(7)	84.48(7)	61.11(8)	71.78(3)	77.57 (3)	81.26(5)	85.95(5)	88.43(5)
2DNMF	73.64(12)	82.11(11)	85.35(12)	71.55(7)	81.84(5)	85.94(7)	83.76(10)	87.70(7)	90.13(10)

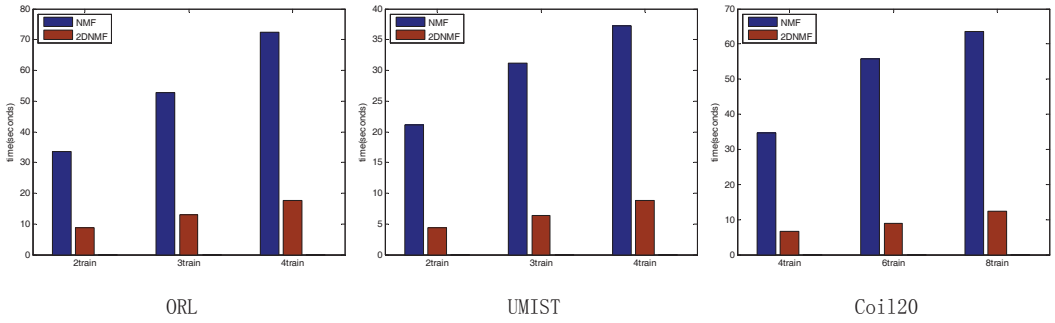

Fig. 2. Training time on ORL (left), UMIST (middle) and Coil20 (right) data sets.

We can see that the computational cost of 2DNMF is less than that of NMF, which is consistent with the theoretical analysis in 3.4.

5. CONCLUSIONS

In this paper, we propose a 2DNMF, specifically for a sequence of matrices. It aims to find two base matrices to represent the input matrices in a low dimensional matrix subspace. It not only inherits the advantages of NMF, but also owns the properties of low computational complexity as well as high recognition accuracy. Both theoretical analysis and empirical experiments on benchmark image recognition data sets illustrate that the proposed method is very effective and efficient.

6. REFERENCES

[1] I. T. Jolliffe, *Principal Component Analysis*, Series in Statistics. Springer Verlag, 2002.

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

[3] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-Yu Yang, "Two-dimensional pca: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, 2004.

[4] Jieping Ye, "Generalized low rank approximations of matrices," in *ICML*, 2004.

[5] Le Li and Yu-Jin Zhang, "Non-negative matrix-set factorization," in *ICIG '07: Proceedings of the Fourth International Conference on Image and Graphics*, Washington, DC, USA, 2007, pp. 564–569, IEEE Computer Society.

[6] Daniel D. Lee and H. Sebastian Seung, "Algorithms for non-negative matrix factorization," in *NIPS*, 2000, pp. 556–562.

[7] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, 2004.