WEIGHTED NONNEGATIVE MATRIX FACTORIZATION

Yong-Deok Kim and Seungjin Choi

Department of Computer Science, POSTECH, Korea
{*karma13,seungjin*}*@postech.ac.kr*

ABSTRACT

Nonnegative matrix factorization (NMF) is a widely-used method for low-rank approximation (LRA) of a nonnegative matrix (matrix with only nonnegative entries), where nonnegativity constraints are imposed on factor matrices in the decomposition. A large body of past work on NMF has focused on the case where the data matrix is complete. In practice, however, we often encounter with an incomplete data matrix where some entries are missing (e.g., a user-rating matrix). Weighted low-rank approximation (WLRA) has been studied to handle incomplete data matrix. However, there is only few work on weighted nonnegative matrix factorization (WNMF) that is WLRA with nonnegativity constraints. Existing WNMF methods are limited to a direct extension of NMF multiplicative updates, which suffer from slow convergence while the implementation is easy. In this paper we develop relatively fast and scalable algorithms for WNMF, borrowed from well-studied optimization techniques: (1) alternating nonnegative least squares; (2) generalized expectation maximization. Numerical experiments on MovieLens and Netflix prize datasets confirm the useful behavior of our methods, in a task of collaborative prediction.

Index Terms— Alternating nonnegative least squares, collaborative prediction, generalized EM, nonnegative matrix factorization, weighted low-rank approximation

1. INTRODUCTION

Low-rank approximation (LRA), such as factor analysis and singular value decomposition (SVD), is a fundamental tool in handling multivariate data or tabulated data. The goal of LRA is to seek a parsimonious representation, assuming there are only a small number of factors influencing a set of observed data samples. Various applications of LRA include dimensionality reduction, feature extraction/selection, clustering, and pre-processing for more sophisticated exploratory data analysis.

LRA is often formulated as a matrix factorization problem, the goal of which is to approximate a target matrix (data matrix) by a product of two or three low-rank factor matrices. For example, given a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$, the rank-$r$ approximation involves determining two factor matrices $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times r}$ ($r \leq \min(m, n)$) such that $\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|^2$ is minimized, where $\|\cdot\|$ denotes the Frobenious norm.

In various applications, all entries in the data matrix are nonnegative ($\boldsymbol{X} \geq 0$). Examples include images, documents, spectrograms, and user-ratings. In such a case where the data matrix is nonnegative, nonnegative matrix factorization (NMF) [9] was shown to be useful in seeking more fruitful o r better-interpretable representation, compared to classical LRA such as SVD. Multiplicative updates proposed by Lee and Seung [9] popularized NMF in diverse applications, including face recognition, audio and sound processing, medical imaging, EEG classification for brain computer interface, gene expression data analysis, document clustering, and so on.

A large body of past work on NMF has focused on the case of complete data matrix where all entries are observed without missing values. In practice, however, the data matrix is often incomplete with some of entries are missing or unobserved. For instance, most entries in a user-rating matrix are zeros (unobserved), so that matrix completion is necessary to predict unobserved ratings, which recently becomes a popular approach to *collaborative prediction* [13, 12, 15].

In this paper, we consider a problem of WLRA with nonnegativity constraints imposed on factor matrices, referred to as *weighted nonnegative matrix factorization* (WNMF). There is only a few study on WNMF, while WLRA and NMF have been extensively studied. To our best knowledge, WNMF was first used to deal with missing values in a distance matrix for predicting distances in large-scale networks [11], where a direct extension of NMF was exploited, incorporating binary weights into NMF multiplicative updates. Expectation-maximization (EM) optimization was employed to solve WNMF in [15], where missing entries are replaced by the corresponding values in the current model estimate in the E-step and the standard NMF multiplicative updates are applied on the filled-in matrix in the M-step. These existing methods for WNMF are easy to implement but suffer from slow convergence. Moreover, their accuracy of predicting missing values are often slightly worse than WLRA even though nonnegativity is considered. In this paper, we develop relatively fast and scalable two algorithms for WNMF, exploiting well-studied optimization techniques:

1. alternating nonnegative least squares (**ANLS-WNMF**)
2. generalized expectation-maximization (**GEM-WNMF**).

We consider collaborative prediction as an application of WNMF, the task of which is to estimate missing values in a user-rating matrix to predict a user's preference on an item (movie in this case). We demonstrate the useful behavior of our methods in a task of collaborative prediction using MovieLens and Netflix prize datasets.

2. WEIGHTED NMF

Given nonnegative data matrix $\boldsymbol{X} = [X_{ij}] \in \mathbb{R}_+^{m \times n}$, WNMF seeks two nonnegative *factor matrices* $\boldsymbol{U} \in \mathbb{R}_+^{m \times r}$ and $\boldsymbol{V} \in \mathbb{R}_+^{n \times r}$ which minimize the following objective function

$$\mathcal{J}_{WNMF}(\boldsymbol{U}, \boldsymbol{V}) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij} \left(X_{ij} - [\boldsymbol{U}\boldsymbol{V}^\top]_{ij} \right)^2, \qquad (1)$$

where W_{ij} are nonnegative weights. For example, missing values are taken care of by binary weights W_{ij} given by

$$W_{ij} = \begin{cases} 1 & \text{if } X_{ij} \text{ is observed} \\ 0 & \text{if } X_{ij} \text{ is unobserved.} \end{cases}$$

When all weights are equal to 1, i.e., $W_{ij} = 1$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, (1) is identical to the standard NMF

Throughout this paper, we consider a *user-rating* matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, where the (i,j)-entry X_{ij} corresponds to the rating given by user i on item (movie) j. For example, Netflix prize dataset contains 100,480,507 observed entries corresponding to ratings given by 480,189 users on 17,770 movies. Approximately only 1 % is filled with observed ratings. Matrix completion is a popular method to predict unobserved predictions, which motivated us to pay attention to the problem of WNMF, since user-rating matrices are incomplete nonnegative matrices. In the case of a user-rating matrix, two factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ determined by the minimization of (1) are interpreted as follows:

- Row i in $\boldsymbol{X}$ is user i's rating profile.
- Columns in $\boldsymbol{V}$ are associated with rating profiles from r user communities. In other words, V_{ij} corresponds to the rating given by user community j on item i.
- Row i in $\boldsymbol{U}$ is user i's affinities for r user communities.

We briefly summarize two existing methods for WNMF. A direct extension of NMF which incorporates binary weights into NMF multiplicative updates [11] is referred to as **Mult-WNMF** in this paper. Multiplicative updates for Mult-WNMF are as follows.

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{(\boldsymbol{W} \odot \boldsymbol{X})\,\boldsymbol{V}}{(\boldsymbol{W} \odot [\boldsymbol{U}\boldsymbol{V}^\top])\,\boldsymbol{V}}, \tag{2}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{(\boldsymbol{W}^\top \odot \boldsymbol{X}^\top)\,\boldsymbol{U}}{(\boldsymbol{W}^\top \odot [\boldsymbol{V}\boldsymbol{U}^\top])\,\boldsymbol{U}}, \tag{3}$$

where $\odot$ is Hadamard product (element-wise product) and the division is performed in an element-wise manner as well.

An EM algorithm for WNMF was proposed by Zhang *et al.* [15]. E-step corresponds to imputation where a filled-in matrix $\boldsymbol{Y}$ is computed using the current model estimate and the standard NMF multiplicative updates are applied on the filled-in matrix in the M-step.

Algorithm Outline: **EM-WNMF** [15]

- **E-step**

$$\boldsymbol{Y} \leftarrow \boldsymbol{W} \odot \boldsymbol{X} + (\boldsymbol{1}_{m \times n} - \boldsymbol{W}) \odot \boldsymbol{U}\boldsymbol{V}^\top \tag{4}$$

- **M-step**

$$\boldsymbol{U} \leftarrow \boldsymbol{U} \odot \frac{\boldsymbol{Y}\boldsymbol{V}}{\boldsymbol{U}\boldsymbol{V}^\top\boldsymbol{V}}, \tag{5}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \frac{\boldsymbol{Y}^\top\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^\top\boldsymbol{U}}, \tag{6}$$

In EM-WNMF, the weight matrix $\boldsymbol{W}$ is normalized such that all entries are in the range between zero and one (i.e. $\boldsymbol{W} \leftarrow \boldsymbol{W} / \max_{i,j}(W_{ij})$) and $\boldsymbol{1}_{m \times n} \in \mathbb{R}^{m \times n}$ is the matrix with all elements filled with ones. It was shown in [15] that EM-WNMF outperforms Mult-WNMF in a task of collaborative prediction. However, EM-WNMF and Mult-WNMF suffer from slow convergence, since they are based on multiplicative updates. Our empirical study indicates that the accuracy of predicting missing values by these methods is not satisfactory.

3. ALGORITHMS

We present our algorithms for WNMF, exploiting alternating nonnegative least squares and generalized EM optimization.

3.1. ANLS-WNMF

Several fast algorithms for un-weighted NMF have been recently developed [14, 10, 7, 6], where alternating nonnegative least squares (ANLS) is used to solve the following two nonnegative least squares (NLS) problems in an alternative fashion

$$\min_{\boldsymbol{U} \geq 0} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|^2 \quad \text{and} \quad \min_{\boldsymbol{V} \geq 0} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top\|^2,$$

with $\boldsymbol{V}$ and $\boldsymbol{U}$ fixed, respectively. With $\boldsymbol{U}$ fixed, the NLS problem is tackled by solving multiple right-hand-side nonnegative least squares (MRHS-NLS) problems

$$\min_{\boldsymbol{v}^1 \geq 0} \|\boldsymbol{x}_1 - \boldsymbol{U}\boldsymbol{v}^1\|^2, \ \ldots \ , \min_{\boldsymbol{v}^n \geq 0} \|\boldsymbol{x}_n - \boldsymbol{U}\boldsymbol{v}^n\|^2,$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{m \times n}$ and $\boldsymbol{V} = [\boldsymbol{v}^1, \ldots, \boldsymbol{v}^n]^\top = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r] \in \mathbb{R}^{n \times r}$ (row i and column j in $\boldsymbol{V}$ are denoted by $(\boldsymbol{v}^i)^\top$ and $\boldsymbol{v}_j$, respectively and the same notation is applied to other matrices).

Various optimization methods can be used to solve the NLS problem, including the projected gradient method [3], projected Newton method [2], and active set method [8]. In the case of MRHS-NLS problem, each problem can be solved separately but this approach is inefficient and is very slow. Note that we only have to compute the Hessian matrix one time because all Hessian matrices are equal to $\boldsymbol{U}^\top\boldsymbol{U}$ for each NLS problem. Incorporating this property, Bro and Jong made a substantial speed improvement in active set method [4]. Benthem and Keenan further improved it by re-arranging calculations such that pseudo-inverse computations of sub-matrices of Hessian are minimized [1].

ANLS is a block coordinate descent method where its convergence to a stationary point is guaranteed when the optimal solution is determined for each block. The NMF problem is non-convex but each sub-problem is convex, leading ANLS to work properly. ANLS can be also applied to WNMF because each sub-problem is still convex. The sub-problem of WNMF is a collection of several NLS problems. For example,

$$\begin{aligned}
&\min_{\boldsymbol{V} \geq 0} \ \frac{1}{2} \sum_{i,j} W_{ij} \left(X_{ij} - [\boldsymbol{U}\boldsymbol{V}^\top]_{ij}\right)^2 \\
\Rightarrow \quad &\min_{\boldsymbol{v}^1 \geq 0} \ \frac{1}{2} \sum_{i=1}^{m} w_{i1} \left(x_{i1} - [\boldsymbol{U}\boldsymbol{v}^1]_i\right)^2, \\
&\ldots, \min_{\boldsymbol{v}^n \geq 0} \ \frac{1}{2} \sum_{i=1}^{m} w_{in} \left(x_{in} - [\boldsymbol{U}\boldsymbol{v}^n]_i\right)^2, \\
\Rightarrow \quad &\min_{\boldsymbol{v}^1 \geq 0} \ \frac{1}{2} \|(\boldsymbol{D}_1\boldsymbol{x}_1) - (\boldsymbol{D}_1\boldsymbol{U})\boldsymbol{v}_1\|^2, \\
&\ldots, \min_{\boldsymbol{v}^n \geq 0} \ \frac{1}{2} \|(\boldsymbol{D}_n\boldsymbol{x}_n) - (\boldsymbol{D}_n\boldsymbol{U})\boldsymbol{v}^n\|^2,
\end{aligned}$$

where $\boldsymbol{D}_i = \mathrm{diag}(\boldsymbol{w}_i)^{1/2}$ which is a diagonal matrix with diagonal entries corresponding to the square root of entries in $\boldsymbol{w}_i$. We also define $\boldsymbol{D}^i = \mathrm{diag}(\boldsymbol{w}^i)^{1/2}$.

Unfortunately we can not use Benthem and Keenan's algorithm [1] because the sub-problem (7) is not the MRHS-NLS problem (i.e. the Hessian matrix $\boldsymbol{U}^\top\boldsymbol{D}_{\boldsymbol{w}_i}\boldsymbol{U}$ is different for each NLS problem).

Thus, we have to solve each NLS problem separately. In our implementation, we use the projected Newton method [2] for single NLS problem.

Algorithm Outline: **ANLS-WNMF**

Initialize $\boldsymbol{U}^{(1)} \geq 0$ and $\boldsymbol{V}^{(1)} \geq 0$.
For $t = 1, 2, \ldots$

- Update $\boldsymbol{U}$ row-by-row. For $i = 1, 2, \ldots, m$

$$\boldsymbol{u}^{i(t+1)} \leftarrow \arg\min_{\boldsymbol{u} \geq 0} \frac{1}{2} \|\boldsymbol{D}^i \boldsymbol{x}^i - \boldsymbol{D}^i \boldsymbol{V}^{(t)} \boldsymbol{u}\|^2,$$

  where $\boldsymbol{X} = [\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m]^\top$, and $\boldsymbol{W} = [\boldsymbol{w}^1, \ldots, \boldsymbol{w}^m]^\top$.

- Update $\boldsymbol{V}$ row-by-row. For $j = 1, 2, \ldots, n$

$$\boldsymbol{v}^{j(t+1)} \leftarrow \arg\min_{\boldsymbol{v} \geq 0} \frac{1}{2} \|\boldsymbol{D}_j \boldsymbol{x}_j - \boldsymbol{D}_j \boldsymbol{U}^{(t+1)} \boldsymbol{v}\|^2,$$

  where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, and $\boldsymbol{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]$.

3.2. GEM-WNMF

Expectation-maximization (EM) is one of powerful methods in handling missing values in maximum likelihood estimation [5]. An EM algorithm for WNMF, proposed in [15], is described in Sec. 2, where E-step (4) is performing imputation to determine $\boldsymbol{Y}$ and M-step (5) and (6) re-estimate factor matrices by applying unweighted NMF to the filled-in matrix $\boldsymbol{Y}$.

The filled-in matrix $\boldsymbol{Y}$, in general, is a dense matrix even if $\boldsymbol{X}$ is a very sparse matrix, which prohibits the E-step in EM-WNMF for large-scale data. The memory space required to store $\boldsymbol{Y}$ and the computational complexity to calculate $\boldsymbol{U}\boldsymbol{V}^\top$ are extremely large. Moreover, the M-step requires much more computation time.

We overcome this problem by interweaving E-step and partial M-step. In contrast to EM-WNMF where multiplicative updates are used in the M-step, we employ ANLS optimization for the M-step, where we use Benthem and Keenan's algorithm [1] with projected Newton method to solve MRHS-NLS:

$$\boldsymbol{U} \leftarrow \text{MRHS-NLS}(\boldsymbol{V}^\top \boldsymbol{V}, \boldsymbol{Y}\boldsymbol{V}), \quad (7)$$

$$\boldsymbol{V} \leftarrow \text{MRHS-NLS}(\boldsymbol{U}^\top \boldsymbol{U}, \boldsymbol{Y}^\top \boldsymbol{U}), \quad (8)$$

where two input arguments in MRHS-NLS$(\cdot, \cdot)$ are the terms required to compute the gradient and Hessian. For example, $\nabla \boldsymbol{U} = \boldsymbol{V}^\top \boldsymbol{V}\boldsymbol{U} - \boldsymbol{Y}\boldsymbol{V}$ and $\nabla^2 \boldsymbol{u}^1 = \cdots = \nabla^2 \boldsymbol{u}^m = \boldsymbol{V}^\top \boldsymbol{V}$. It follows from (7) and (8) that only $\boldsymbol{Y}\boldsymbol{V}$ or $\boldsymbol{Y}^\top \boldsymbol{U}$ are required in the M-step, instead of $\boldsymbol{Y}$ itself. This simple trick dramatically alleviates the computation and space burden. For example, the calculation $\boldsymbol{Y}\boldsymbol{V}$ is given by

$$\begin{aligned} \boldsymbol{Y}\boldsymbol{V} &= \left[\boldsymbol{W} \odot \boldsymbol{X} + (\boldsymbol{1}_{m \times n} - \boldsymbol{W}) \odot (\boldsymbol{U}\boldsymbol{V}^\top)\right] \boldsymbol{V} \\ &= \left[\boldsymbol{W} \odot (\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\top)\right] \boldsymbol{V} + \boldsymbol{U}(\boldsymbol{V}^\top \boldsymbol{V}), \end{aligned}$$

which requires $[\boldsymbol{U}\boldsymbol{V}^\top]_{ij}$ for only $W_{ij} > 0$, instead of the entire $\boldsymbol{U}\boldsymbol{V}^\top$.

In addition to this simple trick, we use partial M-step. At earlier iterations, estimation for missing values is not accurate, so solving M-step exactly is not desirable. Thus, iterations in the M-step (7) and (8) proceed just until substantial improvement (instead of determining optimal solutions).

Algorithm Outline: **GEM-WNMF**

Initialize $\boldsymbol{U}^{(1)} \geq 0$ and $\boldsymbol{V}^{(1)} \geq 0$.
For $t = 1, 2, \ldots$

$$\begin{aligned} \boldsymbol{E} &\leftarrow \boldsymbol{W} \odot (\boldsymbol{X} - \boldsymbol{U}^{(t)} \boldsymbol{V}^{(t)\top}), \\ \boldsymbol{U}^{(t+1)} &\leftarrow \text{MRHS-NLS}\left(\boldsymbol{V}^{(t)\top} \boldsymbol{V}^{(t)}, \boldsymbol{E}\boldsymbol{V}^{(t)} + \boldsymbol{U}^{(t)} \boldsymbol{V}^{(t)\top} \boldsymbol{V}^{(t)}\right), \\ \boldsymbol{E} &\leftarrow \boldsymbol{W} \odot (\boldsymbol{X} - \boldsymbol{U}^{(t+1)} \boldsymbol{V}^{(t)\top}), \\ \boldsymbol{V}^{(t+1)} &\leftarrow \text{MRHS-NLS}\left(\boldsymbol{U}^{(t+1)\top} \boldsymbol{U}^{(t+1)}, \right. \\ &\qquad \left. \boldsymbol{E}^\top \boldsymbol{U}^{(t+1)} + \boldsymbol{V}^{(t)} \boldsymbol{U}^{(t+1)\top} \boldsymbol{U}^{(t+1)}\right). \end{aligned}$$

4. NUMERICAL EXPERIMENTS

We evaluate the performance of three algorithms (Mult-WNMF, ANLS-WNMF, GEM-WNMF) in terms of the convergence speed and the prediction accuracy of missing values. All these algorithms were implemented in Matlab and all experiments were run on Intel Core2 Quad 2.4 GHz processor with 8 GB memory under Windows Vista 64bit. We use MovieLens and Netflix prize datasets for our experiments, which are summarized in Table 1.

Table 1. Dataset descriptions.

	MovieLens	Netflix prize
# of users	6,040	480,189
# of movies	3,952	17,770
# of ratings	1,000,209	100,480,507
density (%)	4.19	1.18

In order to get a fair comparison, we design our experiment following way for each test case;

1. Always use the same randomly generated starting point for every algorithms.
2. Run the Mult-WNMF algorithm for a given number of iterations, and record the CPU time used.
3. Then run other algorithms, and stop them once the CPU time used is equal to or greater than that used by the Mult-WNMF algorithm.

We note that each algorithm require different computation time for one iteration. Steps 2 and 3 in our experiment design ensure that a fair comparison is carried out so that every algorithms are run for approximately the same amount of time for each test case.

We set the iteration number of Mult-WNMF to 600. With these setting, the training time was about 500 seconds and 24 hours for MovieLens and Netflix prize data respectively. In the case of MovieLens data, we randomly split the user-ratings into 5 folds and only used 4 partitions for training. In the case of Netflix prize data, we held out 1,408,395 user-ratings which belong to prove set for test data. Note that we added regularization term $\frac{\lambda}{2}(\|\boldsymbol{U}\|^2 + \|\boldsymbol{V}\|^2)$ into objective function (1) to prevent over-fitting problem, where $\lambda = 3$ in our all experiments.

Fig. 1 shows the root mean squared error (RMSE) values against time. Our algorithms, especially ANLS-WNMF, are faster

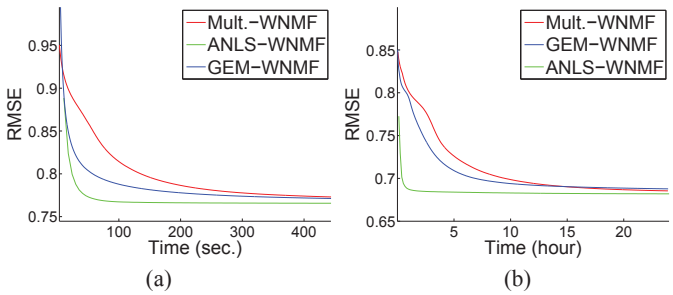

Fig. 1. Convergence speed comparison of Mult-WNMF, ANLS-WNMF, and GEM-WNMF on: (a) MovieLens; (b) Netflix prize data

than Mult-WNMF. Note that, existing EM algorithm for WNMF was about 200 times slower than Mult-WNMF in experiments with MovieLens data [15] and can not applied to Netflix prize data.

In addition to training speed, our algorithms outperformed the Mult-WNMF in the perspective of prediction accuracy of missing values. Table 2 shows the training and test error of each algorithm for MovieLens and Netflix Prize datasets. ALS-WLRA and GEM-WLRA are algorithms for WLRA which are obtained from ANLS-WNMF and GEM-WNMF by releasing the nonnegativity constraints.

Table 2. From top to bottom, training error of MovieLens, test error of MovieLens, training error of Netflix dataset, and test error of Netflix dataset with different rank.

Rank	Mult. WNMF	ANLS WNMF	GEM WNMF	ALS WLRA	GEM WLRA
6	0.8029	0.7988	0.8048	0.8004	**0.7948**
8	0.7815	0.7782	0.7825	0.7751	**0.7714**
10	0.7676	0.7617	0.7671	**0.7511**	0.7564
12	0.7556	0.7462	0.7543	**0.7325**	0.7390
6	0.8664	**0.8615**	0.8624	0.8646	0.8645
8	0.8640	**0.8602**	**0.8602**	0.8613	0.8664
10	0.8689	**0.8637**	0.8638	0.8768	0.8692
12	0.8796	0.8680	**0.8652**	0.8889	0.8776
6	0.7065	0.7049	0.7106	**0.7026**	0.7067
8	0.6944	0.6919	0.6975	**0.6881**	0.6922
10	0.6854	0.6820	0.6878	**0.6770**	0.6810
6	0.9482	**0.9479**	0.9493	0.9600	09529
8	0.9493	0.9483	**0.9439**	0.9676	0.9513
10	0.9523	0.9509	**0.9425**	0.9770	0.9520

The over-fitting problems occurs as the rank increase (i.e. training error is reduced but test error is increased). Note that even nonnegativity is considered, the test error of Mult-WNMF slightly worse than WLRA in the case of MovieLens data set. However our algorithms always showed the best performance.

5. CONCLUSIONS

We have presented relatively fast and scalable algorithms for WNMF, showing its useful behavior in a task of collaborative prediction. Compared to existing methods such as Multi-WNMF and EM-WNMF, we have shown that our algorithms ANLS-WNMF and GEM-WNMF work better with lower computational burden, which fit better for large-scale data.

Acknowledgments: This work was supported by Korea Research Foundation (Grant KRF-2008-313-D00939) and KOSEF WCU Program (Project No. R31-2008-000-10100-0).

6. REFERENCES

[1] M. H. V. Benthem and M. R. Keenan, "Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems," *Journal of Chemometrics*, vol. 18, pp. 441–450, 2004.

[2] D. P. Bertsekas, "Projected Newton methods for optimization problems with simple constraints," *SIAM Journal on Control and Optimization*, vol. 20, no. 2, pp. 221–246, 1982.

[3] D. P. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, 2003.

[4] R. Bro and S. D. Jong, "A fast non-negativity-constrained least squares algorithm," *Journal of Chemometrics*, vol. 11, no. 5, pp. 393–401, 1997.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, 1977.

[6] D. Kim, S. Sra, and I. S. Dhillon, "Fast Newton-type methods for the least squares nonnegative matrix approximation problem," in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2007.

[7] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 2, pp. 713–730, 2008.

[8] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*. Prentice-Hall, 1974.

[9] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[10] C. J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, 2007.

[11] Y. Mao and L. K. Saul, "Modeling distances in large-scale networks by matrix factorization," in *Proceedings of the ACM Internet Measurement Conference*, Taormina, Sicily, Italy, 2004.

[12] J. D. M. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the International Conference on Machine Learning (ICML)*, Bonn, Germany, 2005.

[13] N. Srebro and T. Jaakkola, "Weighted low-rank approximation," in *Proceedings of the International Conference on Machine Learning (ICML)*, Washington DC, 2003.

[14] R. Zdunek and A. Cichocki, "Non-negative matrix factorization with quasi-Newton optimization," in *Proceedings of the Eighth International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, 2006.

[15] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proceedings of the SIAM International Conference on Data Mining (SDM)*, 2006.