# Low-Rank Matrix Approximation with Manifold Regularization

Zhenyue Zhang and Keke Zhao

**Abstract**—This paper proposes a new model of low-rank matrix factorization that incorporates manifold regularization to the matrix factorization. Superior to the graph-regularized nonnegative matrix factorization, this new regularization model has globally optimal and closed-form solutions. A direct algorithm (for data with small number of points) and an alternate iterative algorithm with inexact inner iteration (for large scale data) are proposed to solve the new model. A convergence analysis establishes the global convergence of the iterative algorithm. The efficiency and precision of the algorithm are demonstrated numerically through applications to six real-world datasets on clustering and classification. Performance comparison with existing algorithms shows the effectiveness of the proposed method for low-rank factorization in general.

**Index Terms**—Matrix factorization, graph regularization, classification, clustering, manifold learning

## 1 Introduction

Low-rank matrix factorization (MF) plays an important role in data analysis such as dimension reduction, data compression, feature extraction, and information retrieval. The low-dimensional representations of high-dimensional data facilitates the retrieval of latent data structure or the selection of data features, and have many applications in engineering, for example, low-dimensional representation of symbolic systems [31]. Recently, low-rank MF also finds applications in collaborative filtering (CF) newly developed for product recommendation [15], [29] and advertisement targeting [5], [6]. In the CF applications, the observed data are generally incomplete.

Standard approaches for low-rank approximation include the rank-revealing QR that involves a reordering of the Gram-Schmidt orthogonalization, the triangular-matrix decomposition such as the LU or LDU for its easy implementation, and the truncated singular value decomposition (TSVD), which achieves the best approximation in Frobenius norm (or 2-norm, equivalently),

$$\min_{U^TU=I_r,Y} \|A-UY\|_F^2, \tag{1}$$

for a given matrix $A$ and a preindicated rank $r$ of the approximation. From the numerical point of view, the well-known principal component analysis (PCA) is nothing but a TSVD applied on recentered data. Different from the QR or the LU decomposition, where the input data matrix must be modified in order to extract the orthogonal or triangular factors, the SVD technique can be implemented iteratively via matrix-vector products, and hence it is more effective for large-scale sparse matrices. While the SVD formulation was originally meant to minimize the total sum of all approximation errors, it also naturally minimizes the entry-wise errors in a partial sum. The model is therefore also suitable for recommendation systems by focusing the approximation on observable data only. The modified minimization problem is generally solved via alternating iterative or partial-gradient descent methods.

Nonnegative matrix factorization (NMF) is a special low-rank factorization technique for nonnegative data. Given a data matrix $A \in \mathcal{R}^{m\times n}$, the NMF looks for a low-rank factorization with nonnegative entries in the factors in the sense of [16], [13] to optimize

$$\min_{U\geq 0,Y\geq 0} \|A-UY\|_F^2, \tag{2}$$

where the factors $U \in \mathcal{R}^{m\times r}$ and $Y^T \in \mathcal{R}^{n\times r}$ have $r$ columns. Different from SVD-like methods, the NMF pursues a part-based representation of the data by insisting that both the "basis" vectors and the combination coefficients are nonnegative for the sake of physical interpretation. For instance, in image articulation, the columns of the matrix $U$ represent the intrinsic parts that make up the object being imaged, whereas the columns of $Y$ represent the biometric identification of the individuals. In many cases, the resulting basis vectors and the combination coefficients are sparse. It is obvious that the approximate error of NMF in Frobenius norm is generally larger than that of the SVD if the same approximate rank is taken because of the nonnegative constraint. Practically, it is possible that the NMF may fail to give a true factorization to a nonnegative low-rank matrix with its rank. Here, is a small example that illustrates this phenomenon. The nonnegative matrix

- *Z. Zhang is with the Department of Mathematics and the State Key Laboratory of CAD&CG, Zhejiang University, Yuquan Campus, Hangzhou 310027, China. E-mail: zyzhang@zju.edu.cn.*
- *K. Zhao is with the Department of Mathematics, Zhejiang University, Yuquan Campus, Hangzhou 310027, China. E-mail: kkzhao@zju.edu.cn.*

*Manuscript received 8 Aug. 2011; revised 15 Mar. 2012; accepted 30 Nov. 2012; published online 20 Dec. 2012.*
*Recommended for acceptance by H. Park.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-08-0536.*
*Digital Object Identifier no. 10.1109/TPAMI.2012.274.*

$$A = \begin{bmatrix} 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 0 & 2 & 1 & 1 \end{bmatrix}$$

is of rank 3, and hence the SVD produces the optimal rank-3 factorization of $A$ without errors. However, the optimal nonnegative approximation $UY$ of rank 3 generated by the NMF has the error $\|A - UY\|_F \approx 0.4823$. We remark that the nonnegative rank of a nonnegative matrix, defined by the minimal number in a sum of several nonnegative rank-one matrices that yields a representation of $A$, may be different from the matrix rank [7]. Practically, it is an NP-hard problem to compute the nonnegative rank. One can refer to [8] for a heuristic approach.

The traditional SVD or the NMF ignore the possible nonlinearity inherent in data. For instance, data vectors may be attributed from a low-dimensional manifold [3], [27]. The low-dimensional projection $Y$ obtained by solving (1) or (2) may lose the nonlinear structure. The data nonlinearity could be preserved by nonlinear dimensionality reduction (NDR) methods such as Isomap [23], LLE [21], Laplacian eigenmaps (LE) [2], or LTSA [25], [26]. Generally, these NDR problems can be uniformly rewritten as

$$\min_{YY^T = I_r} \operatorname{tr}\big(Y \Phi Y^T\big),$$

with a symmetric matrix $\Phi$ that characterizes the data manifold. However, these NDR algorithms are not suitable for updating the nonlinear low-dimensional projections when additional data are involved. The locality preserving projection (LPP) method [12] gives a linear approach to determine a low-dimensional projection that partially preserves the locality of the data because it is a linear version of the LE. The LPP requires solving a generalized eigenvalue problem of a dense matrix pair, which results in high overhead in both computation and memory for large scale problems.

Recently, a new model was proposed [4] to incorporate the intrinsic nonlinear DS into a linear low-rank factorization. The idea is to add a regularization term to the NMF model in order to find a part-based representation of the data in which the neighborhood connections are preserved. The graph-regularized NMF, denoted by GNMF henceforth, is defined as the optimization problem

$$\min_{U \geq 0, Y \geq 0} \|A - UY\|_F^2 + \lambda \operatorname{tr}\big(YLY^T\big), \tag{3}$$

where $\lambda$ is a prescribed positive parameter, $L = D - W$ is the Laplacian matrix of the neighborhood graph with the connection weight matrix $W$, and $D$ is the diagonal matrix with column sums of $W$ as its diagonal entries. In the same spirit as that in [16], this optimization problem is solved iteratively by multiplicative rules which update the nonnegative factors alternately. The alternate process guarantees that the objective function is monotonic decreasing at the iterative points. Rigorously speaking, however, the problem (3) as it is has no optimal solutions different from the NMF solutions. The following argument shows that (3) has no critical points if the regularization works.

Given a pair of nonnegative factors $(U, Y)$, the new pair $(\frac{1}{t}U, tY)$ with any positive $t < 1$ produces a smaller objective value because

$$\begin{aligned} &\|A - (t^{-1}U)(tY)\|_F^2 + \lambda \operatorname{tr}\big((tY)L(tY)^T\big) \\ &\quad = \|A - UY\|_F^2 + t^2 \lambda \operatorname{tr}(YLY^T). \end{aligned} \tag{4}$$

One can conclude that if the GNMF problem (3) has an optimal solution $(U_*, Y_*)$, $Y_*$ must satisfy $\operatorname{tr}(Y_* L Y_*^T) = 0$, or equivalently, $LY_*^T = 0$, because $L$ is positive semidefinite. Furthermore, $(U_*, Y_*)$ should be also an optimal solution to the NMF problem (2) without the graph constraint.[1] For a nonzero optimal $Y$-factor of the NMF, however, there is no guarantee for $LY_*^T = 0$. Therefore, the GNMF has no optimal solutions if $LY^T \neq 0$ for any optimal $Y$-factor of the NMF. In that case, the iteration sequence $\{Y_k\}$ generated by the alternate algorithm given in [4] may tend to zero, while the corresponding $\{U_k\}$ tends to infinity. It is hence necessary to scale $Y_k$ before it vanishes. However, the scaling procedure like $Y \leftarrow D^{-1}Y, U \leftarrow UD$ with a diagonal matrix $D$ of positive diagonals will increase the value of the objective function, which will destroy the nonincreasing property of the alternating iterations and result in divergence. It is not clear whether such a diagonal scaling can generate a local minimum better than the global one (the NMF solution) or the factorization could benefit from the scaling in applications such as clustering.

There are other ways to ascertain the factors $U$ or $Y$ without scaling. In [11], a new regularization term in the form $\alpha \operatorname{tr}(U(E - I)U^T)$, with $E$ the matrix of all ones, is added to the objective function. This modification makes the model more complicated because one has to deal with the balance between the two parameters $\lambda$ and $\alpha$, yet the improvement is very limited. In [18], the following model is considered:

$$\min_{U,Y} \|A - UY\|_F^2 + \alpha\big(\|U\|_F^2 + \|Y\|_F^2\big) + \lambda \operatorname{tr}(YLY^T), \tag{5}$$

where the norms of the factors $U$ and $Y$ are used as penalty, but $U$ and $V$ are not subject to the nonnegative constraint. The problem (5) can be solved by an alternate approach. This approach has better performance than the GNMF for SVM classification in our experiments. However, the alternating method needs to solve the complex linear system

$$U^T UY + Y(\alpha I + \lambda L) = U^T A$$

for updating $Y$ at each iteration. As a substitute, a steepest descending approach is suggested in [18] to inexactly minimize the objective function with respect to $Y$.[2] Our numerical experiments in Section 5 show that the alternate method with the steepest decent still converges slowly. The graph regularization technique can also be used to weaken unnecessary connections, while necessary connections are strengthened as suggested in [20] for recommendation systems. We emphasize that for the low-rank approximation problem with missing data such as in the CF, some implicit information should be taken into account in both

1. The inequality $\|A - U_* Y_*\|_F > \min_{U \geq 0, Y \geq 0} \|A - UY\|_F$ implies $\|A - UY\|_F < \|A - U_* Y_*\|_F$ for a nonnegative pair $(U, Y)$. It follows that for a sufficiently small $t > 0$, $(\frac{1}{t}U, tY)$ achieves a smaller value of the objective function of (3) than $(U_*, Y_*)$.

2. The computation cost of each steepest descent step for all rows of $Y$ is $O(r\|L\|_0 + r\|A\|_0 + r^2 n)$, where $\|\cdot\|_0$ is the number of nonzero entries of a matrix. The cost given in [18] is incorrect.

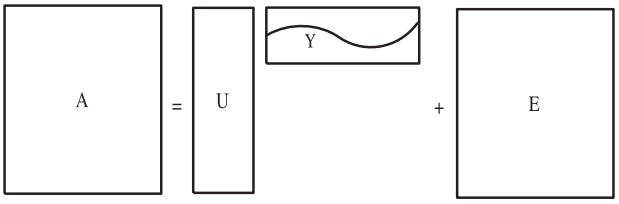


Fig. 1. Low-rank factorization preserving the nonlinearity in the projection $Y$.

the factor structure and the regularization as discussed in [14] and [28].

Different from the GNMF, the idea of the symmetric NMF (SymNMF) given in [17] for clustering is to directly apply the nonnegative factorization on a symmetric and nonnegative graph matrix $W$. The SymNMF produces a symmetric and nonnegative low-rank approximation to the graph matrix by solving

$$\min_{H\geq 0} \|W - HH^T\|_F^2, \tag{6}$$

a symmetric model of (2). It was reported in [17] that the SymNMF performs better than GNMF on some examples of real-world data.

The goal of this paper is to present a simple low-rank factorization model that also takes into account the intrinsic nonlinear structure of the data without data missing. We look for a low-rank factorization that preserves the nonlinearities of the data as much as possible while the optimal approximation to the data matrix is also attained. This motivation is illustrated in Fig. 1. To this end, we release the restriction of nonnegativity but impose the orthonormal constraint on the factor $U$. We also adopt the manifold constraint as in (3). Our regularization problem is stated as follows:

$$\min_{U^TU=I_r,Y} \|A - UY\|_F^2 + \lambda \mathrm{tr}\big(Y\Phi Y^T\big), \tag{7}$$

where $\Phi$ is a specified symmetric and positive definite matrix that extracts the nonlinearity of the manifold or the graph structure inherent in the data. For example, $\Phi$ could be the Laplacian matrix in the LE, the centered weight matrix $I - W$ in the LLE, or the alignment matrix in the LTSA. It can also be the Laplacian of the link matrix in text analysis [19], [30]. The new model combines the traditional optimal low-rank approximation that minimizes the first term and the modern nonlinear dimensional reduction methods that minimizes the second term into one. Fig. 2 gives a diagram of relations of our new model with the optimal low-rank approximation and the modern nonlinear dimensional reduction. Compared with other graph-regularization models, the model (7) has the following advantages:

- Similarly to the SVD, being relaxed from nonnegativity helps to produce a better approximation. Good approximation strengthens the ability of extracting the nonlinear structure via the regularization term.
- Our model has globally optimal solutions, each of which differs only by a rotation. More importantly, the optimal solutions have closed form. This property is distinctive from other graph-regularized methods and nonnegative factorization methods.
- The global optimal solution can be computed directly or iteratively. If the data have a small number of points, the closed form can be employed directly; otherwise, we can use an iterative algorithm which has global convergence.
- Our iterative scheme uses matrix-vector multiplications, making it effective for large and sparse matrices. Additionally, our method works in a low-dimensional space and the coefficient matrix is well conditioned, making the linear system involved in each iteration easy to solve. This fast convergence makes the algorithm particularly suitable for large problems.

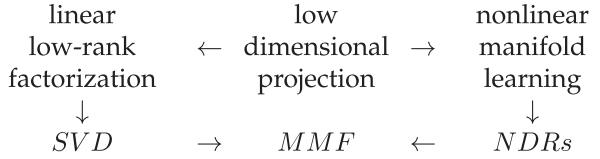


Fig. 2. Diagram of the relations of MMF with the optimal low-rank approximation and nonlinear dimension reduction.

These advantages will be discussed in detail and illustrated via several real-world examples in the applications of data clustering and classification. The proposed algorithm will also be compared with the existing low-rank factorization methods mentioned above.

The rest of the paper is organized as follows: In Section 2, we describe our model in detail and give the closed-form solution to the regular model (7). An economic implementation scheme of the closed form is given for data with a small number of points. Our fast iteration algorithm for solving the problem with large number of data is given in Section 3. Section 4 contains the convergence analysis of the iterative algorithm. Numerical experimental results and comparisons of our algorithm with six other algorithms are reported in Section 5. Some conclusions will be given in the last section.

## 2 Manifold Regularized Low-Rank Approximation

Let $A \in \mathcal{R}^{m\times n}$ be a given matrix of data whose columns are vectors in an $m$-dimensional space. We assume that we also have, a priori, a symmetric positive semidefinite matrix $\Phi$ that characterize the low-dimensional structure of the manifold behind the data points in the sense that $\mathrm{tr}(Y\Phi Y^T)$ is small for a low-dimensional representation $Y$ of the data points if $Y$ preserves the intrinsic structure of the manifold approximately. The constraint matrix $\Phi$ can be the Laplacian matrix of the neighborhood graph or others. In our model, this manifold constraint is embodied in the optimal low-rank approximation model to extract the intrinsic structure in the low-rank factorization, as in

$$\min_{U^TU=I_r,Y} \|A - UY\|_F^2 + \lambda \mathrm{tr}\big(Y\Phi Y^T\big). \tag{8}$$

The left factor $U$ is an orthonormal matrix $U \in \mathcal{R}^{m\times r}$ whose columns consist of a principle orthogonal basis of

the data space. The right factor $Y \in \mathcal{R}^{r\times n}$ has $n$ columns of $r$-dimensional vectors, which can be viewed as a projection or representation of the data in the low-dimension space. Because of the normality of $U$, it is not necessary to renormalize the factor $U$ or $Y$ that is a big disadvantage in nonnegative factorization. We will see later, as in the truncated SVD, this kind of factorization yields better approximation than a nonnegative model. The prominent character of the new model is that the global solutions exist and have a closed form, which we now demonstrate.

Obviously, the optimal solutions of (8) are not unique due to the orthogonal invariance of the trace function $\mathrm{tr}(\cdot)$, that is, $(U, Y)$ is an optimal solution if and only if $(UQ, Q^TY)$ is also an optimal solution for any orthogonal matrix $Q$ of order $r$. However, as we will show later (Theorem 1), the product matrix $X = UY$ for any optimal solution is generally unique. Practically, if $X = UY$, then $\mathrm{tr}(Y\Phi Y^T) = \mathrm{tr}(X\Phi X^T)$. Conversely, given a matrix $X$ with rank not larger than $r$, one can also rewrite it as $X = UY$ with an orthonormal $U$.[3] Hence, $(U, Y)$ is an optimal solution of (8) if and only if $X = UY$ is an optimal solution to

$$\min_{\mathrm{rank}(X)\le r} \|A - X\|_F^2 + \lambda \mathrm{tr}\big(X\Phi X^T\big). \tag{9}$$

That is, (8) and (9) are equivalent to each other in the sense $X = UY$.

The objective function

$$f(X) = \|A - X\|_F^2 + \lambda \mathrm{tr}\big(X\Phi X^T\big)$$

is quadratic in $X$. We rewrite it in a standard form of matrix approximation. To this end, let

$$\Psi = I + \lambda\Phi,$$

which is a symmetric and positive definite matrix, and write $f$ in the form

$$f(X) = \|A\|_F^2 - 2\mathrm{tr}\big(XA^T\big) + \mathrm{tr}\big(X\Psi X^T\big).$$

To further represent it in a form of the classical low-rank approximation, we need the square root $B$ of $\Psi$, which is also a symmetric and positive definite matrix.[4] Thus, $\mathrm{tr}(X\Psi X^T) = \|XB\|_F^2$. It follows that

$$\begin{aligned} f(X) &= \|A\|_F^2 - 2\mathrm{tr}\big(XBB^{-1}A^T\big) + \|XB\|_F^2 \\ &= \|A\|_F^2 + \|XB - AB^{-1}\|_F^2 - \|AB^{-1}\|_F^2. \end{aligned}$$

The problem (9) is hence equivalent to

$$\min_{\mathrm{rank}(X)\le r} \|XB - AB^{-1}\|_F^2.$$

Under the one-to-one transformation,

$$X \to Z = XB,$$

which preserves the rank of $X$, the original problem (8) or (9) appears in the standard form of low-rank approximation:

$$\min_{\mathrm{rank}(Z)\le r} \|AB^{-1} - Z\|_F^2.$$

Therefore, the optimal solution $Z^*$ to the above problem exists and it is given by the truncated SVD of $AB^{-1}$, which shows that (9) has the globally optimal solution $X^* = Z^*B^{-1}$. We remark that $Z^*$ is unique if $AB^{-1}$ has different singular values $\sigma_r$ and $\sigma_{r+1}$ as shown in [10]. The following theorem is hence proven.

**Theorem 1.** *Let $B$ be the squared root of $\Psi = I + \Phi$. Then, $X^*$ is an optimal solution of (9) if and only if $X^* = Z^*B^{-1}$ with $Z^*$ an optimal rank $r$ approximation $Z^*$ of $AB^{-1}$. Furthermore, if the $r$th and $(r+1)$th singular values of $AB^{-1}$ are different, the optimal solution of (9) is unique.*

The closed form $X^* = Z^*B^{-1}$ employs the inverse of the square root $B$ of $\Psi$. From the numerical point of view, this formula is very complicated and computationally expensive for datasets with large numbers of points. Another difficulty is that the matrix $AB^{-1}$ could be dense even if $\Psi$ is sparse. Fortunately, a simple representation of the globally optimal solution $X^*$ exists in which the square root $B$ can be avoided, as we next show.

An optimal rank-$r$ approximation to $AB^{-1}$ can be written in the form

$$Z^* = UU^TAB^{-1},$$

where $U$ is any orthogonal basis matrix of the column space of $Z^*$.[5] Hence, the optimal solution of (9) is given by

$$X^* = Z^*B^{-1} = UU^TA\Psi^{-1}, \tag{10}$$

which also produces an optimal solution $(U, Y)$ to the original problem (8) with $Y = U^TA\Psi^{-1}$.

It is known that $UU^T$ is the orthogonal projector onto the $r$-dimensional space spanned by the $r$ left singular vectors of $AB^{-1}$, or equivalently, onto the invariant subspace of $A\Psi^{-1}A^T$ spanned by its $r$ largest eigenvectors. This simple representation (10) is important to understanding our iteration algorithm given in the next section. We summarize above analysis as the following theorem.

**Theorem 2.** *Let $U$ be the matrix of $r$ eigenvectors corresponding to the $r$ largest eigenvalues of $A\Psi^{-1}A^T$, where $\Psi = I + \lambda\Phi$. Then, for any orthogonal matrix $Q$ of order $r$, $(UQ, Q^TY)$ with $Y = U^TA\Psi^{-1}$ is a global optimal solution of (8).*

The quandary of computing the optimal solution via the closed form is the involvement of the inverse matrix of the sparse matrix $\Psi$. If the number $n$ of the data points is small, computing $\Psi^{-1}$ is not a big problem because $\Psi$ is small in size. In this case, one can also compute the eigenvector matrix $U$ of the resulting matrix $A\Psi^{-1}A^T$ via an eigenvalue solver directly because $A\Psi^{-1}A^T$ has at most $n$ nonzero eigenvalues, whether the dimension of the data points is large or not. Below is an economical way of computing $U$ and $Y$ when $n$ is not large.

First, compute the Cholesky factorization $\Psi = CC^T$ with $C$ a low-triangular matrix, giving the decomposition $A\Psi^{-1}A^T = (AC^{-T})(AC^{-T})^T$. Then, compute the QR decomposition $AC^{-T} = U_0R$ of the factor $AC^{-T}$. The computational

3. There are many such factorizations, one of which is the well-known QR decomposition.

4. Mathematically, $S = U_\Psi D_\Psi U_\Psi^T$ from the EVD of $\Psi = U_\Psi D_\Psi^2 U_\Psi^T$ with positive diagonal matrix $D_\Psi$.

5. An efficient way of generating such a matrix $U$ is to compute the $r$ left singular vectors of $AB^{-1}$ corresponding to its $r$ largest singular values.

cost is low. Because the $r \times r$ upper triangular matrix $R$ is small, it is easy to compute its SVD. Let $U_1$ be the matrix of the left singular vectors of $R$ corresponding to the $r$ largest singular values. We obtain $U = U_0U_1$ immediately. Furthermore, substituting $U = U_0U_1$ and $\Psi^{-1} = C^{-T}C^{-1}$ into $Y = U^TA\Psi^{-1}$ and using the equality $R = U_0^TAC^{-T}$, we have $Y$ in the form

$$Y = U_1^TU_0^TAC^{-T}C^{-1} = U_1^TRC^{-1}.$$

The optimal $Y$ can be obtained at a low cost. We list the direct approach of solving the low-rank factorization problem for datasets with a small number of data points as follows:

**MMF: Direct algorithm for manifold regularized MF.**

1. Compute the Cholesky factorization: $\Psi = CC^T$.
2. Compute $C^{-1}$ and the QR factorization of $AC^{-T}$: $AC^{-T} = U_0R$.
3. Compute the $r$ left singular vectors $U_1$ of the small matrix $R$ corresponding to the largest singular values. Then, set

$$U = U_0U_1, \quad Y = U_1^TRC^{-1}.$$

## 3 Alternate Iteration Method

It is absolutely necessary to avoid the inverse matrix $\Psi^{-1}$ in an effective scheme of computing optimal solution of (8) for data with large number of points. To this end, iteratively solving this optimization problem is a good choice. However, because of nonconvexity, the classical optimization algorithms such as Newton-like methods could not work well. In this section, we propose an effective iterative method via the technique of alternating iteration to address this implementation problem.

Write the objective function $f(X)$ in terms of the two factors $U$ and $Y$:

$$f(U,Y) = \|A - UY\|_F^2 + \lambda \mathrm{tr}(Y\Phi Y^T).$$

The idea of the alternate method can be simply stated as follows: Fixing an estimate of one factor, say $U$, one can determine the other factor $Y$ by minimizing $f(\cdot, Y)$ and vice versa. For the problem $\min_Y f(U,Y)$ with a fixed $U$, it is easy to derive the optimal solution by setting the derivative of $f(\cdot, Y)$ with respect to $Y$ to be zero. Since

$$\frac{\partial f(U,Y)}{\partial Y} = 2(Y\Psi - U^TA),$$

this derivative is zero if and only if $Y = U^TA\Psi^{-1}$. This optimal property is also indicated in the last section.

For a fixed $Y$, the problem of minimizing $f(U,\cdot)$ with the orthonormal constraint is equivalent to the Procrustes problem:

$$\min_{U^TU=I_r} \|A - UY\|_F^2, \tag{11}$$

which can be solved easily via the SVD of $AY^T = GDV^T$. It is shown in [10] that the optimal solution is $U = GV^T$. We give the details of the proof. Writing

$$\|A - UY\|_F^2 = \|A\|_F^2 + \|Y\|_F^2 - 2\mathrm{tr}\big(U^TAY^T\big),$$

we see that the Procrustes problem (11) is equivalent to maximizing the trace function $\mathrm{tr}(U^TAY^T)$ that can be represented in terms of the singular values of $AY^T$. Let $AY^T = GDV^T$ be the SVD of $AY^T$ with $D$ a diagonal matrix of positive diagonal entries. We have

$$\begin{aligned}\mathrm{tr}\big(U^TAY^T\big) &= \mathrm{tr}(U^TGDV^T) = \mathrm{tr}\big(V^TU^TGD\big)\\ &= \sum_{i=1}^{r} \big(V^TU^TG\big)_{ii}D_{ii}.\end{aligned}$$

Because $V^TU^TG$ is an orthogonal matrix, its diagonal entries are not larger than one in absolute value. The maximum of the trace function is then achieved when all the diagonals of $V^TU^TG$ are one, which implies that $V^TU^TG$ is an identity matrix of order $r$. Hence, the optimal solution to the Procrustes problem (11) is given by $U = GV^T$.

The SVD of the thin matrix $AY^T$ can be obtained in an economical way similar to the direct approach shown in the last section. First, compute its QR-decomposition $AY^T = G_0R$, where $G_0$ is an orthogonal matrix of $r$ columns and $R$ is an upper triangular matrix of order $r$. Then, compute the SVD of the small matrix $R = G_1DV^T$. The required SVD of $AY^T$ follows immediately: $AY^T = GDV^T$ with $G = G_0G_1$.

The standard steps for updating $U$ and $Y$ can be summarized as.

$$\begin{cases} \text{Update } U: & (a)\ AY^T = GDV^T; \quad (\text{SVD of } AY^T) \\ & (b)\ U = GV^T; \\ \text{Update } Y: & Y = U^TA\Psi^{-1}. \end{cases} \tag{12}$$

To get $Y$, it is not necessary to compute the inverse matrix $\Psi^{-1}$. One can solve the matrix equation

$$Y\Psi = U^TA \tag{13}$$

for updating $Y$, which is equivalent to the following independent linear systems:

$$y_i\Psi = u_i^TA, \quad i = 1, 2, \ldots, r, \tag{14}$$

for the $r$ rows of $Y$, where $u_1, \ldots, u_r$ are the $r$ columns of $U$. Iteratively solving the linear systems is a good idea because the coefficient matrix $\Psi$ is sparse and symmetric positive definite. There are many effective iterative algorithms for solving this kind of system. We suggest the preconditioned conjugate gradient (PCG) method with a preconditioner matrix $M$, see [22]. We can simply take the matrix $M$ to be the diagonal matrix of $\Psi$. We will refer to the iterations of PCG for solving the linear systems with fixed $U$ as inner iterations relative to the (outer) iterations for updating $U$ and $Y$.

We emphasize that it is not necessary to solve the linear systems (14) with high accuracy. One can fix the number of inner iterations for each outer iteration. Practically, the matrix $Y$ obtained in the previous outer iteration is a good initial guess for the current inner iterations of the PCG. As the outer iterations go on, the inner iterations converge quickly. The strategy of fixing the number of inner iterations helps to reduce the computational cost to a large extent. In our numerical experiments, we used at most 25 iterations of the PCG for each outer iteration.

Now, we are ready to give the details of our alternating iteration algorithm for datasets with large number of points.

**MMF: Iteration algorithm for manifold regularized MF.**

1. Set an initial guess $(U_0, Y_0)$ of $(U, Y)$ with unitary $U_0$, set $k_{inner} = 25$ and $k = 1$.
2. For $k = 1, 2, \ldots$, until convergence:
   a. Inexactly solve the linear systems (14) with previously obtained $U = U_{k-1}$ by PCG with $M = \mathrm{diag}(\Psi)$ to obtain $Y_k$, starting with $Y_{k-1}$ and terminating within at most $k_{inner}$ iterations.
   b. Compute the SVD $AY_k^T = G_k D_k V_k^T$ of $AY_k^T$ in an economical way and set $U_k = G_k V_k^T$.
3. Set $U = U_k$ and $Y = Y_k$.

The computational cost of the algorithm MMF for updating the $Y$-factor consists of the cost of solving $r$ linear systems, each of which is inexactly solved within at most $k_{inner}$ PCG iterations. The PCG iteration needs a linear system solving with the coefficient matrix $M$ for preconditioning, and a matrix-vector product with the matrix $\Psi$, and three vector inner products. In our algorithm, $M$ is a diagonal matrix, and hence one can solve the preconditioning system $Mx = b$ with cost $n$. Because $\Psi = I + \lambda\Phi$ is generally a sparse matrix, the matrix-vector product costs $n + \mathrm{nnz}(\Phi)$. So the complexity of a PCG iteration is about $5n + \mathrm{nnz}(\Phi)$. Together with the cost $r\mathrm{nnz}(A)$ for computing $r$ right-hand-side vectors, the computational complexity of Step 2a is about $r(\mathrm{nnz}(A) + k_{inner}(5n + \mathrm{nnz}(\Phi)))$. In Step 2b, it requires $r\mathrm{nnz}(A)$ for $AY_k^T$, about $2mr^2$ for the OR-decomposition of $AY_k^T$, $21r^3$ for the SVD of the small $R$-factor in the QR, and $(m + r)r^2$ for forming $U_k$. In total, the computational cost of an outer iteration of the MMF, including $k_{inner}$ inner iterations, is about

$$r\big\{2\mathrm{nnz}(A) + r(3m + 22r) + k_{inner}\big(5n + \mathrm{nnz}(\Phi)\big)\big\}.$$

Additionally, one may need to construct $\Phi$. The number of outer iterations depends on the data. In our experiments, the total computational cost is almost the same as the GNMF, see Fig. 4. This algorithm can be applied on datasets in large scale, especially on sparse data. We tested a simulative sparse dataset in the scale $m = n = 200,000$ with 10M nonzero entries, and $\Phi$ is the Laplacian of a graph with 5-10 direct connections for each node. This algorithm MMF has 100 outer iterations. In the five real-word examples reported in Section 5, the MMF can achieve relative accuracy (22) with $\tau = 10^{-6}$ within at most 40 outer iterations.

**Remark.** This algorithm can also be used for a dataset that has a small number of points. To be more effective in the case when $n$ is small, the inner iteration in Step 2a can be replaced by a direct approach as follows: One can compute the Cholesky factorization of $\Psi = CC^T$ with $C$ a triangle matrix before starting the outer iterations, and then solve the two linear systems $ZC^T = U_{k-1}^T A$ (for $Z$) and $YC = Z$ (for Y) to get $Y_k = Y$ at each outer iteration. The direct approach given in the last section may be more effective if $n$ is small.

## 4 Convergence Analysis

In this section, we give a convergence analysis for the alternating method (12) based on the subspace iteration for computing an invariant subspace. For simplicity, we assume that the matrix (13) is exactly solved at each outer iteration. We will show that the sequence $\{U_k Y_k\}$ generally converges to the optimal solution.

Consider the typical step of the algorithm from the current pair $(U, Y)$ to the new pair $(\hat{U}, \hat{Y})$:

$$AY^T = GDV^T \quad \Rightarrow \quad \hat{U} = GV^T, \quad \hat{Y} = \hat{U}^T A\Psi^{-1}, \qquad (15)$$

where $G$ and $V$ are orthonormal matrices. We first show that the updated orthonormal matrix $\hat{U}$ can be represented in a subspace updating of $\Upsilon = A\Psi^{-1}A^T$:

$$\hat{U}H = \Upsilon U, \qquad (16)$$

with $H = VDV^T$, which basically says that $\hat{U}$ is an orthogonal basis matrix of the range space of $\Upsilon U$. The equality (16) follows directly from the iteration scheme (15). Practically, by the construction of $\hat{U}$ and $H$,

$$\hat{U}H = GDV^T = AY^T.$$

Substituting $Y = U^T A\Psi^{-1}$ into the above equality, we obtain (16) immediately.

Equation (16) shows the interesting property behind the alternating method (12): The updating scheme of the $U$-factor is exactly the subspace iteration of the symmetric matrix $\Upsilon$ which converges to the subspace spanned by the eigenvectors corresponding to the $r$ largest eigenvalues [10]. Given an approximate basis $U$, one first computes the transformation $\Upsilon U$ of the basis vectors, and then factorizes it as $\Upsilon U = \hat{U}H$ with an orthonormal $\hat{U}$. Numerically, the orthogonal factorization can be done via QR decomposition. On the classical theory of subspace iteration [10], it is not difficult to show convergence as follows.

Consider the eigenvalue decomposition of $\Upsilon$:

$$\Upsilon = PSP^T = P_1 S_1 P_1^T + P_2 S_2 P_2^T, \qquad (17)$$

where $P = [P_1, P_2]$ is the matrix of orthonormal eigenvectors of $\Upsilon$, and $S = \mathrm{diag}(S_1, S_2)$ is the diagonal matrix of the eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n$ with blocks $S_1$ of the $r$ largest eigenvalues and $S_2$ of the remaining $n - r$ smaller eigenvalues. We assume that $\lambda_r > \lambda_{r+1}$. Write the initial guess $U_0$ in items of $P_1$ and $P_2$ as

$$U_0 = P_1 C_1 + P_2 C_2. \qquad (18)$$

We assume that $C_1 = P_1^T U_0$ is nonsingular, which implies that $U_0$ has full informations of the invariant subspace spanned by $P_1$.[6] Based upon (16), the $U$-factor $U_k$ obtained after $k$ iterations has the recursion $U_k H_k = \Upsilon U_{k-1}$, where $H_k$ is the corresponding matrix $H$ in (16). It follows that with $F_k = H_k \cdots H_1$ and $F_0 = I$,

$$\begin{aligned} U_k F_k &= \Upsilon(U_{k-1}F_{k-1}) = \Upsilon^2(U_{k-2}F_{k-2}) \\ &= \cdots = \Upsilon^k(U_0 F_0) = \Upsilon^k U_0. \end{aligned}$$

6. This is a natural assumption for subspace iteration, and generally it can be easily satisfied by randomly choosing an orthonormal matrix of $r$ columns as $U_0$.

Substituting $U_0$ given by (18) into the equality $U_kF_k = \Upsilon^k U_0$ and using the eigenproperty $\Upsilon^k P_i = P_i S_i^k$ for $i = 1, 2$, which follows from the eigendecomposition (17) of $\Upsilon$, we have

$$U_kF_k = P_1S_1^kC_1 + P_2S_2^kC_2. \tag{19}$$

The first term is dominant because $0 \le \lambda_{r+1}/\lambda_r < 1$ and

$$\frac{\|P_2S_2^kC_2\|_2}{\|P_1S_1^kC_1\|_2} = \frac{\|S_2^kC_2\|_2}{\|S_1^kC_1\|_2} \le \left(\frac{\lambda_{r+1}}{\lambda_r}\right)^k \frac{\|C_2\|_2}{\|C_1\|_2} \to 0,$$

as $k \to \infty$. Here, we have used the inequalities

$$\sigma_{\min}(A)\|B\|_2 \le \|AB\|_2 \le \sigma_{\max}(A)\|B\|_2,$$

for arbitrary $A$ and $B$, where $\sigma_{\min}(\cdot)$ and $\sigma_{\max}(\cdot)$ denote the smallest and largest singular values, respectively. The dominancy helps to understand that the subspace spanned by $U_k$ tends to the eigenspace spanned by $P_1$ more and more, or equivalently, $P_2^TU_k \to 0$. Because of its importance in the convergence analysis of our algorithm MMF, we prove it below.

The equality (19) implies the two equations

$$P_1^TU_kF_k = S_1^kC_1, \quad P_2^TU_kF_k = S_2^kC_2.$$

Eliminating the factor $F_k$ yields

$$P_2^TU_k\big(P_1^TU_k\big)^{-1} = S_2^kC_2\big(S_1^kC_1\big)^{-1} = S_2^k\big(C_2C_1^{-1}\big)S_1^{-k}.$$

It gives

$$P_2^TU_k = S_2^k\big(C_2C_1^{-1}\big)S_1^{-k}\big(P_1^TU_k\big).$$

Therefore, since $\|P_1^TU_k\|_2 \le 1$, we see that as $k \to \infty$,

$$\begin{aligned}\big\|P_2^TU_k\big\|_2 &\le \big\|S_2^k\big\|_2\big\|C_2C_1^{-1}\big\|_2\big\|S_1^{-k}\big\|_2\\ &\le \left(\frac{\lambda_{r+1}}{\lambda_r}\right)^k\big\|C_2C_1^{-1}\big\|_2 \to 0.\end{aligned}$$

This property further guarantees the convergence of $\{U_kU_k^T\}$ to $P_1P_1^T$, the orthogonal projector onto the eigensubspace of $\Upsilon$ corresponding to the $r$ largest eigenvalues:

$$\begin{aligned}\big\|U_kU_k^T - P_1P_1^T\big\|_F &= \big\|\big(I - P_1P_1^T\big)U_k\big\|_F\\ &= \big\|P_2P_2^TU_k\big\|_F\\ &= \big\|P_2^TU_k\big\|_F \to 0.\end{aligned}$$

Theorem 2 states that the optimal solution of (8) is given by $X^* = P_1P_1^TA\Psi^{-1}$, the orthogonal projection of $A\Psi^{-1}$ onto the largest $r$-dimensional eigensubspace. It is interesting that the iteration matrix $U_kY_k = U_kU_k^TA\Psi^{-1}$ is also an orthogonal projection of $A\Psi^{-1}$ onto an approximate space of the eigen-subspace. The approximation implies the convergence of the sequence $\{U_kY_k\}$:

$$\begin{aligned}\|U_kY_k - X^*\|_F &= \big\|\big(U_kU_k^T - P_1P_1^T\big)A\Psi^{-1}\big\|_F\\ &\le \big\|U_kU_k^T - P_1P_1^T\big\|_2\big\|A\Psi^{-1}\big\|_F \to 0.\end{aligned}$$

We have proven the convergence of the iteration algorithm (12).

**Theorem 3.** *The sequence $\{X_k = U_kY_k\}$ generated by the iteration algorithm of (12) converges to the optimal solution $X^*$ of the regularized problem (8) if $\lambda_r > \lambda_{r+1}$ and the initial $U_0$ has invertible $C_1$ in the splitting (18).*

## 5 Numerical Implementations

We performed our MMF algorithm on two typical problems: clustering and classification in data processing. Six examples of real-world datasets are considered. The following five real-world datasets are used for the clustering problem[7]:

- `coil20`. It contains $32 \times 32$ gray scale images of 20 objects viewed from varying angles. Each object has 72 images.
- `pie`. It is a set of $32 \times 32$ gray scale face images of 68 people. Each person has 42 facial images under different light and illumination conditions.
- `tdt2`. This is a text database, containing 9,394 documents from 30 categories. Each document is represented in a vector of dimension 36,771. Unlike the first two datasets, the numbers of documents in different categories are different in this set.
- `Cora-I`. The database consists of 2,708 machine learning (ML) papers and can be classified as seven sets: case based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. Each paper is a word document of length 1,433. Beside the document matrix of size $1{,}433 \times 2{,}708$, we also have a citation matrix of size $2{,}708 \times 2{,}708$.
- `CiteSeer`. It consists of six classes of papers: Agents, AI, DB, IR, ML, and HCI. This dataset also contains a word-document matrix of size $3{,}703 \times 3{,}312$ and a citation matrix of size $3{,}312 \times 3{,}312$.

Another kind of dataset Cora,[8] called `Cora-II`, is used to demonstrate our approach on classification problems. `Cora-II` consist of abstracts and references of research papers in the five fields of computer science: data structure (DS), hardware and architecture (HA), ML, operation system (OS), and programming language (PL); each of the fields is of multiple subjects. Two matrices, the content matrix and the citation matrix, corresponding to each field are given in the dataset. See Table 1 for the collection details (dimensions, numbers of papers, and subfields) of the data.

We will report the numerical results and compare our algorithm MMF with six algorithms: the direct matrix factorization (MF) obtained by the truncated SVD, NMF [16], the symmetric nonnegative matrix factorization (SymNMF) [17], the graph-constrained NMF (GNMF), the joint link-content matrix factorization (LCMF) [30], and the relation regularized matrix factorization (RRMF) [18], by applying the $K$-means (for clustering) and the

7. `coil20` is available at http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php. `pie` and `tdt2` can be found at http://www.zjucadcg.cn/dengcai/GNMF/, and the last two sets are downloaded from http://www.cs.umd.edu/~sen/lbc-proj/LBC.html.

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

TABLE 1
Data Information of Cora-II

| Research field | # of papers | Data dimension | # of subjects |
|---|---|---|---|
| Data structure (DS) | 751 | 6234 | 9 |
| Hardware and architecture (HA) | 400 | 3989 | 7 |
| Machine learning (ML) | 1617 | 8329 | 7 |
| Operation system (OS) | 1246 | 6737 | 4 |
| Programming language (PL) | 1575 | 7949 | 9 |

SVM (for classification) on the low-dimensional projections of these algorithms.

### 5.1 Effectiveness Measurements for Clusterings

The effectiveness of a low-rank factorization for clustering can be measured by the clustering accuracy and the normalized mutual information (NMI) [24]. The clustering accuracy is defined as the average correction (AC) with

$$AC = \frac{1}{n}\sum_{j=1}^{n}\delta\big(l_j, l_j^*\big), \tag{20}$$

where $l_j^*$ is the true class label of data point $j$ and $l_j$ is the assigned label by the clustering algorithm.[9] The function $\delta(\cdot,\cdot)$ is the Kronecker delta whose value is 1 if the two labels are equal and zero otherwise.

The mutual information (MI) metric of two sets of clusters $\mathcal{C}$ and $\mathcal{C}'$ is defined as

$$MI(\mathcal{C},\mathcal{C}') = \sum_{c_i\in\mathcal{C},c_j'\in\mathcal{C}'} p\big(c_i,c_j'\big)\cdot\log_2\frac{p\big(c_i,c_j'\big)}{p(c_i)p\big(c_j'\big)},$$

where $p(c_i)$ and $p(c_j')$ are the probabilities of a document belonging to the clusters $c_i$ and $c_j'$, respectively, with

$$p(c_i) := \frac{1}{n}|\{j : l_j = c_i\}|, \quad p\big(c_j'\big) := \frac{1}{n}\big|\big\{i : l_i^* = c_j'\big\}\big|,$$

and $p(c_i,c_j')$ is the joint probability. The NMI metric is

$$NMI(\mathcal{C},\mathcal{C}') = \frac{MI(\mathcal{C},\mathcal{C}')}{\max(H(\mathcal{C}),H(\mathcal{C}'))}, \tag{21}$$

with $H(\mathcal{C}) = -\sum_{c_i\in\mathcal{C}} p(c_i)\log_2(p(c_i))$.

### 5.2 Clusterings

We first verify the performance of the MMF for clustering. The rank $r$ is set to be equal to the number of classes of the dataset. The Laplacian $L = D - W$ of a similarity matrix $W$ is used as the constraint matrix $\Phi$, where $D$ is the diagonal matrix of the row-sums of $W$. The matrix $W$ could be a matrix of neighborhood graph of data points or a link matrix such as the citation matrix given in a paper collection dataset. When a neighborhood graph is used, we use the binary weight for $W$, i.e., $w_{ij} = 1$ if $x_i$ and $x_j$ are neighbors, or $w_{ij} = 0$ otherwise. The neighborhood graph is generated using the $k$-nearest-neighbor (k-NN) strategy in variant distance metrics, depending on the dataset.

We use the classical k-NN of the original data points in euclidean distance with $k = 5$ for the dataset coil20. For the image dataset pie, we construct the neighborhoods by local binary patterns [1] with $k$ nearest neighbors ($k = 15$) in euclidean distance. The cosine metric is used for tdt2 to determining the neighborhood graph, according to the $k$ largest values among $\cos\langle x_1,x_i\rangle,\cdots,\cos\langle x_n,x_i\rangle$ for each point $x_i$. We set $k = \lfloor\log_2(n)\rfloor + 1$. For CiteSeer and Cora, both the k-NN neighborhood graph of the word-document matrix and the natural citation graph are used. The parameter $\lambda$ is set to be 50, 50, 5, 50, and 50 for the five datasets, respectively.

The clustering algorithm K-means is applied on the $r$-dimensional representation $Y$ of data obtained by the MMF. We use two kinds of distances in the K-means: cosine distance $1-\cos\langle y_i,y_j\rangle$ for text data and euclidean distance $\|y_i - y_j\|_2$ for image data.[10] The resulting classifications of K-means depend on the initial class centers because it may converge to a local minima. We repeat times of K-means with randomly chosen initial sets of class centers for each dataset and report the best results in the highest accuracy.

We also compare MMF with the factorization algorithms SVD, NMF, GNMF, LPP, and SymNMF with the same value of rank $r$ as in the MMF. Below, we briefly describe some numerical implementation issues of the algorithms in our experiments for clustering.

The optimal MF problem (1) is solved by the sparse SVD algorithm svds in MATLAB. The K-means is applied on the factor $Y$, repeated 50 times, as in the MMF.

It is known that NMF cannot guarantee a global optimal solution. One has to run NMF several times with randomly selected initial nonnegative factors and choose the best one as the output of NMF. In our experiments, the NMF algorithm based on multiplicative rules for alternatively updating the two nonnegative factors [16] is used because of its simple implementation. We repeated NMF 10 times, each has at most 2,000 iterations or achieves the difference $10^{-4}$ of objective values at two iterations. We repeat the K-means on the best result for 50 times.

It is necessary for the GNMF to scale the factorization factors suitably. In our experiments, we use the diagonal scaling suggested in [4]. The GNMF also greatly depends on the initial values. It is also suggested in [4] to use a rough solution of NMF as the initial solution for GNMF. In our experiments, the rough solution of NMF is given by the best one of the 10 runnings of NMF each has 200 iterations. The GNMF terminates with 1,000 iterations. So, in total, 3,000 iterations are required generally.

The SymNMF solves $\min_{H\geq 0}\|W - HH^T\|_F^2$ with a given symmetric graph matrix $W$. We use the k-NN neighborhood graph of the datasets in euclidean distance with $k = \lfloor\log_2(n)\rfloor + 1$ and the weights

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{\sigma_i\sigma_j}\right),$$

where $\sigma_i$ is the distance between $x_i$ and its $\hat{k}$th neighbor ($\hat{k} = 7$), see [17]. The graph matrix is diagonally scaled to be $D^{-1/2}WD^{-1/2}$, with $D$ as in the Laplacian $L = D - W$ before factorizing. As suggested in [17], the data points are

9. The labeling values should be reordered to match the true labels as much as possible.

10. Cosine distance is much better than the euclidean distance for the GNMF projection of text datasets in our experiments. There are no big differences when the two distances are used for MMF projection.

TABLE 2
Clustering Effectiveness of the Algorithms on tdt2, coil20, and pie

| Data | $K$ | Average correction (%) | | | | | | Normalized mutual information (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SVD | NMF | GNMF | LPP | SymNMF | MMF | SVD | NMF | GNMF | LPP | SymNMF | MMF |
| coil20 | 4 | 62.5 | 63.9 | 93.4 | 61.8 | 93.4 | **93.8** | 45.0 | 47.1 | 85.8 | 70.3 | 83.8 | **86.3** |
| | 8 | 50.9 | 49.3 | 80.2 | 74.3 | 73.6 | **86.1** | 52.3 | 48.7 | 80.0 | 80.3 | 75.7 | **81.9** |
| | 12 | 57.8 | 59.8 | 84.6 | 86.1 | 77.7 | **88.4** | 63.9 | 63.9 | 86.1 | **88.4** | 82.8 | 87.3 |
| | 16 | 69.4 | 67.2 | 84.9 | 78.6 | 75.7 | **88.5** | 75.2 | 71.4 | **90.6** | 88.1 | 84.2 | 90.3 |
| | 20 | 68.8 | 66.2 | 80.4 | 83.1 | 78.7 | **86.6** | 76.9 | 74.0 | 89.5 | 90.7 | 86.2 | **91.6** |
| tdt2 | 5 | 97.7 | 97.2 | 97.6 | 96.2 | **98.2** | 97.9 | 91.4 | 90.0 | 90.8 | 90.6 | **92.9** | 91.5 |
| | 10 | 83.8 | 79.5 | 85.0 | 74.7 | **93.2** | 92.4 | 75.3 | 73.9 | 76.1 | 75.1 | **87.4** | 84.9 |
| | 15 | 76.4 | 74.1 | 77.7 | 61.2 | **93.2** | 92.3 | 74.0 | 73.4 | 72.6 | 68.6 | **87.9** | 86.2 |
| | 20 | 65.9 | 62.7 | 73.5 | 56.1 | **90.9** | 90.0 | 70.2 | 68.9 | 72.1 | 65.8 | **88.6** | 87.1 |
| | 25 | 62.6 | 58.8 | 64.5 | 54.7 | **92.6** | 84.6 | 67.9 | 66.9 | 68.0 | 63.2 | **89.7** | 83.8 |
| pie27 | 10 | 38.1 | 58.8 | 98.8 | **100** | 99.1 | **100** | 45.3 | 71.6 | 97.9 | **100** | 98.2 | **100** |
| | 20 | 31.7 | 73.9 | 84.1 | 79.2 | 90.5 | **92.4** | 47.3 | 81.7 | 90.0 | 82.9 | 94.5 | **95.7** |
| | 30 | 28.8 | 72.6 | 80.2 | **89.6** | 87.2 | 89.1 | 51.2 | 84.6 | 89.6 | 94.0 | 93.3 | **95.6** |
| | 40 | 27.1 | 74.2 | 76.6 | **88.6** | 85.8 | 87.2 | 52.3 | 86.7 | 87.9 | 93.4 | 93.3 | **94.0** |
| | 50 | 25.9 | 73.8 | 77.8 | **90.1** | 77.1 | 85.2 | 53.7 | 86.5 | 86.6 | **94.9** | 87.8 | 93.3 |

TABLE 3
Clustering Effectiveness of the Algorithms on CiteSeer and Cora-I ($\lambda = 5$)

| Data | Graph | Average Accuracy (%) | | | | | | Normalized Mutual Information (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MF | NMF | GNMF | LPP | SymNMF | MMF | MF | NMF | GNMF | LPP | SymNMF | MMF |
| Cora-I | neighborhood | 38.4 | 36.2 | 35.9 | 37.8 | 53.8 | **54.0** | 17.0 | 13.7 | 13.2 | 15.2 | **31.2** | 29.7 |
| | citation | 38.3 | 36.3 | 38.6 | 41.6 | 49.7 | **63.8** | 17.0 | 13.7 | 19.2 | 17.7 | 33.2 | **47.6** |
| CiteSeer | neighborhood | 45.3 | 45.4 | 46.9 | 44.3 | 56.9 | **64.5** | 20.2 | 19.7 | 20.6 | 19.3 | 29.5 | **37.4** |
| | citation | 45.4 | 45.4 | 49.7 | 44.5 | 31.5 | **69.0** | 20.2 | 19.7 | 22.2 | 19.6 | 11.1 | **43.3** |

classified according to the indices of the largest components in each row of the optimal solution $H$. We used the Newton algorithm of SymNMF on the relatively small sets coil20 and pie, and applied the gradient descent version on the large set tdt2 because of "out of memory" when the Newton algorithm is applied on tdt2. The initial solution $H$ is randomly selected, with entries normally distributed in the interval $(0, \alpha)$ with

$$\alpha = 2\sqrt{\sum w_{ij}/(n^2 r)}.$$

Both of the two iterative versions may converge to a local minima. We execute the SymNMF 10 times within at most 1,000 iterations for each dataset to reduce the risk of local minima.[11]

The LPP solution is given by eigenvectors of a generalized eigenvalue problem. If the data dimension is larger than the number of the data points, the generalized eigenvalue problem will be singular. It is necessary to preproject the data into a space of smaller dimension before applying the LPP in that case. As a preparation, we preproject data points into a space whose dimensionality is equal to the number of data points by the truncated SVD of rank $K$ before applying the LPP. Here, $K$ is equal to the number of classes.

Table 2 shows the performances of the six algorithms on tdt2, coil20, and pie, measured by the AC (20) and the NMI (21). The first $K$ classes of each dataset are used to test the algorithm's performance for the $K$-clustering problem with various values of $K$.[12] All the data points (column vectors) are scaled to be unitary in euclidean norm. The best results are bold faced. We also note the second best results with underline. It is a bit surprising that the SymNMF performs much better on tdt2 than that reported in [17]

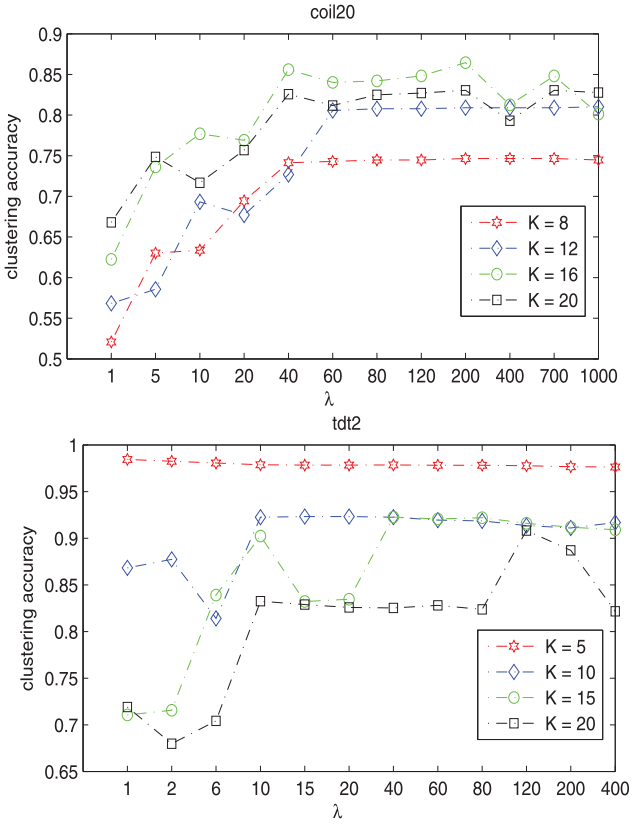


Fig. 3. The clustering accuracy of MMF versus $\lambda$.

11. Local minima still occurs in our experiments.
12. This setting makes the experiments checkable. Similar results are also obtained when we randomly select $K$ classes.

TABLE 4
Accuracy of SVM in Percent on the Projections or Original Data of Cora-II ($\lambda = 3$, Rank $r = 8C$)

| Field | C | Content data | | | Link data | | | Content data with link graph | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MF | NMF | Original data | MF | NMF | Original data | GNMF | LCMF | RRMF | MMF |
| DS | 9 | 53.6 | 55.6 | 57.9 | 52.7 | 54.8 | 55.9 | 52.6 | 63.2 | 78.1 | **81.0** |
| HA | 7 | 66.4 | 67.1 | 72.1 | 63.7 | 66.2 | 67.1 | 70.0 | 73.3 | 85.9 | **87.9** |
| ML | 7 | 68.9 | 64.3 | 70.4 | 60.2 | 55.7 | 61.6 | 65.2 | 75.5 | 86.2 | **86.7** |
| OS | 4 | 72.8 | 69.8 | 73.0 | 69.5 | 65.8 | 72.1 | 68.6 | 79.2 | 87.4 | **88.3** |
| PL | 9 | 50.7 | 47.4 | 55.6 | 55.2 | 53.1 | 56.7 | 47.6 | 60.4 | 75.7 | **76.1** |

due to the data normalization. Practically, without the normalization, the SymNMF gives much lower ACs: The AC values on the five testing subsets are

$$82.3,\ 73.0,\ 82.0,\ 78.4,\ \text{and}\ 81.0,$$

percent, similar to that reported by the authors in [17]. The MMF also performs as good as the SymNMF, with accuracy slightly lower than that of SymNMF. In the experiments of coil20, the MMF always gives the best results measured by AC. The factor-scaled GNMF performs better than the SymNMF. On pie, none of the six algorithms can always give the best results on the five subsets with various $K$, though the MMF looks better than the LPP at the NMI measurement, and vice versa at the AV measurement. The GNMF performances are quite different on this dataset when an "accurate" or a "rough" solution of NMF is used as an initial guess for GNMF, respectively. For example, if the optimal solutions of NMF reported in Table 2 are used for GNMF, the AC values of the GNMF on the five subsets are

$$23.1,\ 21.9,\ 19.8,\ 24.0,\ \text{and}\ 18.6,$$

percent, respectively, though the factor scaling technique is also used. The NMI accuracies are lower than or approximately equal to 50 percent. It supports to some extent our analysis on the convergence of GNMF. We point out that the MMF always gives the best or second best results among all the tested subsets of the three datasets. This phenomenon partially shows the stability of the MMF.

However, all six algorithms do not work well on CiteSeer and Cora-I as well as on the previous three datasets. We apply the MF, NMF, LPP, and SymNMF on both the document matrix and the citation matrix, and apply the GNMF and MMF on the document matrix with both the neighborhood graph of document data and the citation graph. Table 3 shows the results. The accuracies are not satisfactory. There are no big differences among the MF, NMF, and LPP on the document data or citation data. None of them has AC accuracy higher than 50 percent, and the NMI accuracy is even lower than 23 percent. It is slightly better for the GNMF to use the citation graph than the neighborhood graph of document data. The SymNMF performs a bit better than these four algorithms on the document data, rather than on the citation graphs. The MMF significantly outperforms the other five algorithms in this experiment. The highest AC accuracy and NMI accuracy obtained by SymNMF can be increased about 20 and 43 percent, respectively.

The algorithm MMF is stable for the parameter setting. Fig. 3 shows how the clustering accuracy of MMF depends on the parameter $\lambda$. There are no big changes if $\lambda$ is set a bit larger for both of tdt and coil20. A priori for $\lambda$ is to choose it from the interval $(\|\Phi\|_2, K\|\Phi\|_2)$. For example, the norm $\|\Phi\|_2$ for coil20 is $8.5, 7.9, 10.3, 12.4$ corresponding to $K = 8, 12, 16, 20$, respectively. Any choice of $\lambda$ in the interval $(\|\Phi\|_2, K\|\Phi\|_2)$ is always good for the MMF.

### 5.3 Classifications

We apply the linear SVM [9] on the low-dimensional projections of the database Cora-II obtained by the low-rank factorization algorithms to classify the papers. Eighty percent of the projected data are used as training points and others are testing points. The SVM implementation used in this experiment is the LIBLINEAR,[13] which is fast for datasets in large scale. The experiment is repeated 10 times for each training-testing splitting. We use the average accuracy (the percentage of the number of correctly classified documents in the entire testing set) of the 10 experiments.

SVM cannot classify the database Cora-II in high accuracy on the original data or the projected data obtained by the algorithms MF or NMF. So does SVM on the GMNF projection if the Laplacian is constructed by the dataset (the content matrix or the link matrix) itself for the GNMF. It is expected that combining the link information into the content data may improve the SVM accuracy. However, it is a bit unexpected that for the content data the GNMF using the link graph decreases the SVM accuracy on the original data in each of the five research areas. The SVM accuracy on the LCMF projection is slightly increased. Both of RRMF and MMF give much better results than GNMF and LCMF. Indeed, MMF gives the best result for the content data from each of the five fields. We list the accuracy of SVM on the original data or the low-dimensional projections obtained by GNMF, LCMF, RRMF, and MMF with $\lambda = 3$ and $r = 8C$ in Table 4, where $C$ is the number of subjects of each of the five research fields.

The performances of RRMF and MMF are similar in this example. However, MMF still gives slightly better results than RRMF. To further compare these two algorithms in detail, we vary the value of $\lambda$ in the set $\{1, 3, 5, 10\}$ with a fixed rank for both of RRMF and MMF and choose the best result of each algorithm. Table 5 lists the accuracy of SVM on the low-dimensional projections of RRMF and MMF with rank $r$ varying from $C$ to $8C$. In almost all the cases, MMF performs better than RRMF.

It should be emphasized that GNMF is fast, though it gives much lower accuracy than RRMF. However, RRMF is very slow and cannot handle data in large scale. The CPU of RRMF is 10 times more than that of MMF in this example.[14] In Fig. 4, we compare the CPU times (in

13. http://www.csie.ntu.edu.tw/~cjlin/liblinear/.
14. The RRMF results are obtained using the code proposed by the authors of [18].

TABLE 5
Accuracy of SVM on the $Y$-Factors of MMF and RRMF

| | $1C$ | | $2C$ | | $4C$ | | $8C$ | |
|---|---|---|---|---|---|---|---|---|
| | RRMF | MMF | RRMF | MMF | RRMF | MMF | RRMF | MMF |
| DS | 57.3 | 57.8 | 70.8 | 71.8 | 76.9 | 79.0 | 80.6 | 81.4 |
| HA | 77.3 | 81.3 | 82.9 | 84.0 | 86.9 | 89.1 | 87.7 | 89.0 |
| ML | 76.5 | 81.2 | 82.4 | 85.2 | 86.4 | 86.0 | 86.8 | 86.7 |
| OS | 76.0 | 80.0 | 83.6 | 84.8 | 84.5 | 86.5 | 87.6 | 88.3 |
| PL | 58.4 | 63.1 | 69.5 | 71.5 | 73.1 | 75.0 | 77.2 | 78.5 |

seconds) of the four algorithms MMF, GNMF, LCMF, and RRMF with rank $r = 8C$ and $\lambda = 3$ on the five research fields. MMF is fast as GNMF.

### 5.4 Effectiveness of Graph Regularization

The effectiveness of the graph regularization in the models (3) and (7) is affected by the approximation error, the first term in each of the two models. It is known that the nonnegative constraint on the factors may cause the approximation error much larger than the optimal error achieved by the truncated SVD. Thus, graph regularization may lose its effectiveness because of the bad approximation in the nonnegative model. Compared with the nonnegative model, our regularization model could make the regularization more active and more efficient, which helps to preserve the nonlinear structure of data in the linear factorization. To illustrate this observation, in Table 6 we list the relative values $\mathrm{tr}(Y\Phi Y^T)/\|Y\|_F^2$ of the graph regularization achieved by GNMF and MMF on the four datasets `tdt2`, `coil`, `pie`, and `cora-II`. It is obvious that the graph regularization works more effectively in MMF than it does in GNMF. This significant difference partially explains why MMF outperforms GNMF on `tdt2`, `pie`, and `cora-II`. For `coil`, there are no big differences on graph effectiveness, and the clustering performances of MMF and GNMF are similar.

### 5.5 Convergence of MMF

As shown in the convergence analysis given in Section 4, MMF converges globally if the linear system involved in each iteration for updating $Y$ is solved exactly. However, it is not necessary to have exact solutions for updating $Y$ in applications. When PCG with limited iterations is used, we only have an approximate solution for the linear system. In this section, we report the numerical performance of MMF with inexact inner iteration.

Three datasets, `CiteSeer`, `coil20`, and `pie`, are tested in this experiment. Since the scales of these datasets are small, one can get the exact solution $(U^*, Y^*)$ within machine accuracy by a direct method. We also compute the iterative solution of MMF under the accurate rule or the inaccurate rule for the PCG iteration. At most 25 inner iterations are used for PCG and the outer iteration terminates if the relative error of the objective function satisfies

$$\frac{f(U_k, Y_k) - f(U^*, Y^*)}{f(U^*, Y^*)} < \tau, \tag{22}$$

for a given tolerance $\tau > 0$.

Table 7 gives the iteration numbers of MMF under the accurate rule and the inaccurate rule with various tolerances for the three testing datasets. There are almost no differences on the required iterations of MMF under the two different rules. It clearly shows that the inexact PCG approximation is enough for updating the $Y$-factor. Hence, the MMF iteration converges fast and the computational cost is low, as shown in Fig. 4.

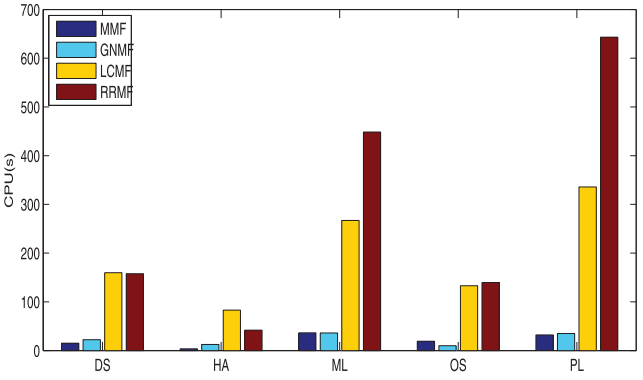


Fig. 4. Comparing CPU times between MMF, GNMF, LCMF, and RRMF on `Cora-II` with $\lambda = 3$ and $r = 8C$.

## 6 Conclusions

In this paper, we presented a novel low-rank MF model that incorporates the networked information or manifold structure of data. It balances the best approximation as SVD and the preservation of nonlinearity as algorithms for nonlinear dimensional reduction. Compared with the existing graph-regularization models for nonnegative factorization, this new model has some prominent properties: A global optimization solution exists and it has a closed form. We proposed a direct algorithm (for data with a small number of points) and an alternate iterative algorithm with inexact inner iteration (for large scale data) for solving the new model. A convergence analysis is given to show that the alternating iterative algorithm converges globally. This is an evident advantage compared with other alternating algorithms for solving similar problems.

Due to the implementation of iterations in a low-dimensional space and its fast convergence, this algorithm, named MMF, has computational cost as low as the

TABLE 6
Accuracy $\mathrm{tr}(Y\Phi Y^T)/\|Y\|_F^2$ of MMF and GNMF

| `tdt2` | | | `coil` | | | `pie` | | | `cora-II` | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| K | GNMF | MMF | K | GNMF | MMF | K | GNMF | MMF | Field | GNMF | MMF |
| 5 | 8.12e-03 | 9.57e-03 | 4 | 1.35e-03 | 7.92e-04 | 10 | 4.87e-03 | 1.60e-04 | DS | 2.67e+00 | 4.54e-02 |
| 10 | 1.67e-02 | 1.32e-02 | 8 | 1.47e-03 | 1.03e-03 | 20 | 4.05e-03 | 1.28e-04 | HA | 2.91e+00 | 5.30e-02 |
| 15 | 5.75e-02 | 1.49e-02 | 12 | 9.52e-04 | 7.72e-04 | 30 | 3.15e-03 | 1.26e-04 | ML | 3.19e+00 | 6.63e-02 |
| 20 | 4.77e-02 | 1.58e-02 | 16 | 1.34e-03 | 7.30e-04 | 40 | 3.20e-03 | 1.26e-04 | OS | 5.15e+00 | 4.34e-02 |
| 25 | 8.21e-02 | 1.69e-02 | 20 | 8.80e-04 | 5.84e-04 | 50 | 3.55e-03 | 1.30e-04 | PL | 4.04e+00 | 8.11e-02 |

TABLE 7
The Iteration Numbers After Convergence with $Y$ Updated in Two Methods

| Data set | $\lambda$ | Accurate rule | | | | Inaccurate rule | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1E-3 | 1E-4 | 1E-5 | 1E-6 | 1E-3 | 1E-4 | 1E-5 | 1E-6 |
| CiteSeer $r = 6$ | 1 | 7 | 25 | 42 | 58 | 7 | 25 | 42 | 58 |
| | 10 | 6 | 16 | 29 | 40 | 6 | 16 | 29 | 40 |
| | 25 | 5 | 12 | 27 | 42 | 5 | 12 | 26 | 41 |
| | 50 | 5 | 13 | 32 | 52 | 5 | 12 | 30 | 50 |
| coil20 $r = 20$ | 1 | 6 | 12 | 26 | 41 | 6 | 12 | 26 | 41 |
| | 10 | 6 | 10 | 14 | 20 | 6 | 10 | 14 | 19 |
| | 25 | 6 | 13 | 21 | 31 | 5 | 10 | 18 | 28 |
| | 50 | 4 | 8 | 12 | 17 | 4 | 6 | 8 | 12 |
| pie $r = 68$ | 1 | 6 | 13 | 24 | 38 | 6 | 13 | 24 | 38 |
| | 10 | 3 | 6 | 12 | 21 | 3 | 6 | 12 | 21 |
| | 25 | 2 | 4 | 10 | 21 | 2 | 4 | 10 | 24 |
| | 50 | 2 | 3 | 8 | 25 | 2 | 3 | 7 | 26 |

algorithms for nonnegative factorization. The effectiveness of the MMF has been illustrated numerically by applying it on the two classical problems of data clustering and classification. In the six real-world examples with which we experimented, our new model performs better than the SVD or the NMF techniques without graph constraints and other five graph-involved algorithms. The low-dimensional projection of the new regularized model can increase the accuracy of the clustering method $K$-means and the classifier SVM, especially for the problem of paper classification with citation information.

The effectiveness of a manifold/graph regularized model for low-rank factorization depends on both the abilities of the factorization on low-rank approximation and the nonlinearity preserving. When the manifold structures can be retrieved with high accuracy by minimizing the regularization term only, the constraints on the low-rank factors should be suitably imposed so that the optimal approximate error is also small. Otherwise, larger approximation errors may obstruct the efforts on extracting DS via the regularization. The experiments on `Cora-I`, `CiteSeer`, and `Cora-II` show that a good low-rank factorization may greatly benefit from a heterogenous graph or restriction matrix in the regularization as the link matrix to the document data. Because of the page limit, we did not touch the effectiveness of other kinds of constraints of manifolds in our manifold regularized model for low-rank factorization. This topic will be investigated in our further work.

## Acknowledgments

The work of Zhenyue Zhang was supported in part by NSFC projects 11071218 and 91230112, and the National Basic Research Program of China (973 Program) 2009CB320804.

## References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Description with Local Binary Patterns: Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 2037-2041, Dec. 2006.

[2] M. Belkin and P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering," *Proc. Advances in Neural Information Processing Systems,* vol. 14, pp. 585-591, 2001.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization: A Geometric Framework for Learning from Examples," *J. Machine Learning Research,* vol. 7, pp. 2399-2434, 2006.

[4] D. Cai, X. He, J. Han, and T.S. Huang, "Graph Regularized Nonnegative Matrix Factorization for Data Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 8, pp. 1548-1560, Aug. 2011.

[5] J.F. Canny, "GaP: A Factor Model for Discrete Data," *Proc. ACM Conf. Information Retrieval,* pp. 122-129, 2004.

[6] Y. Chen, D. Pavlov, M. Kapralov, and J.F. Canny, "Factor Modeling for Advertisement Targeting," *Proc. Advances in Neural Information Processing Systems,* vol. 22, pp. 324-332, 2009.

[7] J.E. Cohen and U.G. Rothblum, "Nonnegative Ranks, Decompositions, and Factorizations of Nonnegative Matrices," *Linear Algebra and Its Applications,* vol. 190, pp. 149-168, 1993.

[8] B. Dong, M.T. Lin, and M.T. Chu, "Nonnegative Rank Factorization via Rank Reduction," http://www4.ncsu.edu/mtchu/Research/Papers/NRF04d.pdf, 2013.

[9] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A Library for Large Linear Classification," *J. Machine Learning Research,* vol. 9, pp. 1871-1874, 2008.

[10] G.H. Golub and C.F. Van Loan, *Matrix Computations.* The Johns Hopkins Univ. Press, 1996.

[11] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Manifold Regularized Discriminative Nonnegative Matrix Factorization with Fast Gradient Descent," *IEEE Trans. Image Processing,* vol. 20, no. 7, pp. 2030-2048, July 2011.

[12] X. He and P. Niyogi, "Locality Preserving Projections," *Proc. Advances in Neural Information Processing Systems 16,* 2004.

[13] J. Kim and H. Park, "Fast Nonnegative Matrix Factorization: An Active-Set-Like Method and Comparisons," *SIAM J. Scientific Computing,* vol. 33, no. 6, pp. 3261-3281, 2011.

[14] Y. Koren, "Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 426-434, 2008.

[15] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *Computer,* vol. 42, no. 8, pp. 30-37, Aug. 2009.

[16] D. Lee and H. Seung, "Algorithms for Nonnegative Matrix Factorization," *Proc. Advances in Neural Information Processing Systems* vol. 13, pp. 556-562, 2001.

[17] D. Kuang, C. Ding, and H. Park, "Symmetric Nonnegative Matrix Factorization for Graph Clustering," *Proc. SIAM Int'l Conf. Data Mining,* pp. 106-117, 2012.

[18] W. Li and D. Yeung, "Relation Regularized Matrix Factorization," *Proc. 21st Int'l Joint Conf. Artificial Intelligence,* 2009.

[19] Q. Lu and L. Getoor, "Link-Based Classification," *Proc. 20th Int'l Conf. Machine Learning,* Aug. 2003.

[20] H. Ma, M.R. Lyu, and I. King, "Learning to Recommend with Trust and Distrust Relationships," *Proc. Third ACM Conf. Recommender Systems,* vol. 8, pp. 189-196, 2009.

[21] S.T. Roweis and L.K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, pp. 2323-2326, 2000.

[22] Y. Saad, *Iterative Methods for Sparse Linear Systems,* second ed. SIAM, 2003.

[23] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, pp. 2319-2323, 2000.

[24] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," *Proc. Int'l Conf. Research and Development in Information Retrieval,* pp. 267-273, Aug. 2003.

[25] H. Zha and Z. Zhang, "Spectral Properties of the Alignment Matrices in Manifold Learning," *SIAM Rev.,* vol. 51, no. 3, pp. 545-566, 2009.

[26] Z. Zhang and H. Zha, "Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment," *SIAM J. Scientific Computing,* vol. 26, no. 1, pp. 313-338, 2005.

[27] Z. Zhang, H. Zha, and M. Zhang, "Spectral Methods for Semi-Supervised Manifold Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[28] Z. Zhang, K. Zhao, and H. Zha, "Inducible Regularization for Low-Rank Matrix Factorizations for Collaborative Filtering," *Neurocomputing,* vol. 97, pp. 52-62, 2012.

[29] D. Zhou, S. Zhu, K. Yu, X. Song, B.L. Tseng, H. Zha, and C. Giles, "Learning Multiple Graphs for Document Recommendations," *Proc. 17th Int'l Conf. World Wide Web,* pp. 141-150, 2008.

[30] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining Content and Link for Classification Using Matrix Factorization," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* 2007.
[31] Y. Wen, A. Ray, and S. Phoha, "Hilbert Space Formulation of Symbolic Systems for Signal Representation and Analysis," *Signal Processing,* to appear.

**Zhenyue Zhang** received the BS degree in mathematics and the PhD degree in scientific computing from Fudan University, Shanghai, China, in 1982 and 1989, respectively. He was an assistant professor in the Department of Mathematics, Fudan University, from 1982 to 1985, and has been a full professor in the Department of Mathematics, Zhejiang University, since 1998. His current research interests include machine learning and its applications, numerical linear algebra, and recommendation systems.

**Keke Zhao** received the BS degree from the Department of Mathematics, Xiamen University, Xiamen, China, in 2007, and the PhD degree in scientific computing from Zhejiang University, Hangzhou, China, in 2012, under the supervision of Zhenyue Zhang. His research interests include machine learning, recommendation systems, and numerical optimization.