Graph Regularized Nonnegative Matrix Factorization for Data Representation

Deng Cai, *Member*, *IEEE*, Xiaofei He, *Senior Member*, *IEEE*, Jiawei Han, *Fellow*, *IEEE*, and Thomas S. Huang, *Fellow*, *IEEE*

Abstract—Matrix factorization techniques have been frequently applied in information retrieval, computer vision, and pattern recognition. Among them, Nonnegative Matrix Factorization (NMF) has received considerable attention due to its psychological and physiological interpretation of naturally occurring data whose representation may be parts based in the human brain. On the other hand, from the geometric perspective, the data is usually sampled from a low-dimensional manifold embedded in a high-dimensional ambient space. One then hopes to find a compact representation,which uncovers the hidden semantics and simultaneously respects the intrinsic geometric structure. In this paper, we propose a novel algorithm, called *Graph Regularized Nonnegative Matrix Factorization* (GNMF), for this purpose. In GNMF, an affinity graph is constructed to encode the geometrical information and we seek a matrix factorization, which respects the graph structure. Our empirical study shows encouraging results of the proposed algorithm in comparison to the state-of-the-art algorithms on real-world problems.

Index Terms—Nonnegative matrix factorization, graph Laplacian, manifold regularization, clustering.

1 Introduction

The techniques for matrix factorization have become popular in recent years for data representation. In many problems in information retrieval, computer vision, and pattern recognition, the input data matrix is of very high dimension. This makes *learning from example* infeasible [15]. One then hopes to find two or more lower dimensional matrices whose product provides a good approximation to the original one. The canonical matrix factorization techniques include LU decomposition, QR decomposition, vector quantization, and Singular Value Decomposition (SVD).

SVD is one of the most frequently used matrix factorization techniques. A singular value decomposition of an $M \times N$ matrix $\mathbf{X}$ has the following form:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^T,$$

where $\mathbf{U}$ is an $M \times M$ orthogonal matrix, $\mathbf{V}$ is an $N \times N$ orthogonal matrix, and Σ is an $M \times N$ diagonal matrix with $\Sigma_{ij} = 0$ if $i \neq j$ and $\Sigma_{ii} \geq 0$. The quantities Σ_{ii} are called the *singular values* of $\mathbf{X}$, and the columns of $\mathbf{U}$ and $\mathbf{V}$ are called left and right *singular vectors*, respectively. By removing those singular vectors corresponding to sufficiently small singular values, we get a low-rank approximation to the original matrix. This approximation is optimal in terms of the reconstruction error, and thus optimal for data representation when euclidean structure is concerned. For this reason, SVD has been applied to various real-world applications such as face recognition (*eigenface*, [40]) and document representation (*latent semantic indexing*, [11]).

Previous studies have shown that there is psychological and physiological evidence for parts-based representation in the human brain [34], [41], [31]. The Nonnegative Matrix Factorization (NMF) algorithm is proposed to learn the parts of objects like human faces and text documents [33], [26]. NMF aims to find two nonnegative matrices whose product provides a good approximation to the original matrix. The nonnegative constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. NMF has been shown to be superior to SVD in face recognition [29] and document clustering [42]. It is optimal for learning the parts of objects.

Recently, various researchers (see [39], [35], [1], [36], [2]) have considered the case when the data is drawn from sampling a probability distribution that has support on or near to a *submanifold* of the ambient space. Here, a d-dimensional submanifold of a euclidean space $\mathbb{R}^M$ is a subset $\mathcal{M}^d \subset \mathbb{R}^M$, which locally looks like a flat d-dimensional euclidean space [28]. In order to detect the underlying manifold structure, many *manifold learning* algorithms have been proposed, such as Locally Linear Embedding (LLE) [35], ISOMAP [39], and Laplacian Eigenmap [1]. All of these algorithms use the so-called locally invariant idea [18], i.e., the nearby points are likely to have similar embeddings. It has been shown that learning performance can be significantly enhanced if the geometrical structure is exploited and the local invariance is considered.

- *D. Cai and X. He are with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yu Hang Tang Rd., Hangzhou, Zhejiang 310058, China. E-mail: {dengcai, xiaofeihe}@cad.zju.edu.cn.*
- *J. Han is with the Department of Computer Science, University of Illinois at Urbana Champaign, Siebel Center, 201 N. Goodwin Ave., Urbana, IL 61801. E-mail: hanj@cs.uiuc.edu.*
- *T.S. Huang is with the Beckman Institute for Advanced Sciences and Technology, University of Illinois at Urbana Champaign, Beckman Institute Center, 405 North Mathews Ave., Urbana, IL 61801. E-mail: huang@ifp.uiuc.edu.*

Manuscript received 28 Apr. 2009; revised 21 Dec. 2009; accepted 22 Oct. 2010; published online 13 Dec. 2010.
Recommended for acceptance by D.D. Lee.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-04-0266.
Digital Object Identifier no. 10.1109/TPAMI.2010.231.

Motivated by recent progress in matrix factorization and manifold learning [2], [5], [6], [7], in this paper we propose a novel algorithm, called Graph regularized Nonnegative Matrix Factorization (GNMF), which explicitly considers the local invariance. We encode the geometrical information of the data space by constructing a nearest neighbor graph. Our goal is to find a parts-based representation space in which two data points are sufficiently close to each other, if they are connected in the graph. To achieve this, we design a new matrix factorization objective function and incorporate the graph structure into it. We also develop an optimization scheme to solve the objective function based on iterative updates of the two factor matrices. This leads to a new parts-based data representation which respects the geometrical structure of the data space. The convergence proof of our optimization scheme is provided.

It is worthwhile to highlight several aspects of the proposed approach here:

1. While the standard NMF fits the data in a euclidean space, our algorithm exploits the intrinsic geometry of the data distribution and incorporates it as an additional regularization term. Hence, our algorithm is particularly applicable when the data are sampled from a submanifold which is embedded in high-dimensional ambient space.
2. Our algorithm constructs a nearest neighbor graph to model the manifold structure. The weight matrix of the graph is highly sparse. Therefore, the multiplicative update rules for GNMF are very efficient. By preserving the graph structure, our algorithm can have more discriminating power than the standard NMF algorithm.
3. Recent studies [17], [13] show that NMF is closely related to Probabilistic Latent Semantic Analysis (PLSA) [21]. The latter is one of the most popular topic modeling algorithms. Specifically, NMF with KL-divergence formulation is equivalent to PLSA [13]. From this viewpoint, the proposed GNMF approach also provides a principled way for incorporating the geometrical structure into topic modeling.
4. The proposed framework is a general one that can leverage the power of both NMF and graph Laplacian regularization. Besides the nearest neighbor information, other knowledge (e.g., label information, social network structure) about the data can also be used to construct the graph. This naturally leads to other extensions (e.g., semi-supervised NMF).

The rest of the paper is organized as follows: In Section 2, we give a brief review of NMF. Section 3 introduces our algorithm and provides a convergence proof of our optimization scheme. Extensive experimental results on clustering are presented in Section 4. Finally, we provide some concluding remarks and suggestions for future work in Section 5.

2 A Brief Review of NMF

NMF [26] is a matrix factorization algorithm that focuses on the analysis of data matrices whose elements are nonnegative.

Given a data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M\times N}$, each column of $\mathbf{X}$ is a sample vector. NMF aims to find two nonnegative matrices $\mathbf{U} = [u_{ik}] \in \mathbb{R}^{M\times K}$ and $\mathbf{V} = [v_{jk}] \in \mathbb{R}^{N\times K}$ whose product can well approximate the original matrix $\mathbf{X}$:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T.$$

There are two commonly used cost functions that quantify the quality of the approximation. The first one is the square of the euclidean distance between two matrices (the square of the *Frobenius norm* of two matrices difference) [33]:

$$O_1 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 = \sum_{i,j}\left(x_{ij} - \sum_{k=1}^{K} u_{ik}v_{jk}\right)^2. \qquad (1)$$

The second one is the "divergence" between two matrices [27]:

$$O_2 = D(\mathbf{X}\|\mathbf{U}\mathbf{V}^T) = \sum_{i,j}\left(x_{ij}\log\frac{x_{ij}}{y_{ij}} - x_{ij} + y_{ij}\right), \qquad (2)$$

where $\mathbf{Y} = [y_{ij}] = \mathbf{U}\mathbf{V}^T$. This cost function is referred to as "divergence" of $\mathbf{X}$ from $\mathbf{Y}$ instead of "distance" between $\mathbf{X}$ and $\mathbf{Y}$ because it is not symmetric. In other words, $D(\mathbf{X}\|\mathbf{Y}) \neq D(\mathbf{Y}\|\mathbf{X})$. It reduces to the Kullback-Leibler divergence or relative entropy, when $\sum_{ij} x_{ij} = \sum_{ij} y_{ij} = 1$, so that $\mathbf{X}$ and $\mathbf{Y}$ can be regarded as normalized probability distributions. We will refer O_1 as F-norm formulation and O_2 as divergence formulation in the rest of the paper.

Although the objective functions O_1 in (1) and O_2 in (2) are convex in $\mathbf{U}$ only or $\mathbf{V}$ only, they are not convex in both variables together. Therefore, it is unrealistic to expect an algorithm to find the global minimum of O_1 (or O_2). Lee and Seung [27] presented two iterative update algorithms. The algorithm minimizing the objective function O_1 in (1) is as follows:

$$u_{ik} \leftarrow u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}, \quad v_{jk} \leftarrow v_{jk}\frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk}}.$$

The algorithm minimizing the objective function O_2 in (2) is

$$u_{ik} \leftarrow u_{ik}\frac{\sum_j \left(x_{ij}v_{jk}/\sum_k u_{ik}v_{jk}\right)}{\sum_j v_{jk}},$$
$$v_{jk} \leftarrow v_{jk}\frac{\sum_i \left(x_{ij}u_{ik}/\sum_k u_{ik}v_{jk}\right)}{\sum_i u_{ik}}.$$

It is proven that the above two algorithms will find local minima of the objective functions O_1 and O_2 [27].

In reality, we have $K \ll M$ and $K \ll N$. Thus, NMF essentially tries to find a compressed approximation of the original data matrix. We can view this approximation column by column as

$$\mathbf{x}_j \approx \sum_{k=1}^{K} \mathbf{u}_k v_{jk}, \qquad (3)$$

where $\mathbf{u}_k$ is the kth column vector of $\mathbf{U}$. Thus, each data vector $\mathbf{x}_j$ is approximated by a linear combination of the columns of $\mathbf{U}$, weighted by the components of $\mathbf{V}$. Therefore, $\mathbf{U}$ can be regarded as containing a basis, that is, optimized for the linear approximation of the data in $\mathbf{X}$. Let $\mathbf{z}_j^T$ denote the jth row of $\mathbf{V}, \mathbf{z}_j = [v_{j1}, \dots, v_{jk}]^T$. $\mathbf{z}_j$ can be regarded as the new representation of the jth data point with respect to the new basis $\mathbf{U}$. Since relatively few basis vectors are used to represent many data vectors, a good approximation can only be achieved if the basis vectors discover structure that is latent in the data [27].

The nonnegative constraints on $\mathbf{U}$ and $\mathbf{V}$ only allow additive combinations among different bases. This is the most significant difference between NMF and the other matrix factorization methods, e.g., SVD. Unlike SVD, no subtractions can occur in NMF. For this reason, it is believed that NMF can learn a *parts-based* representation [26]. The advantages of this parts-based representation have been observed in many real-world problems such as face analysis [29], document clustering [42], and DNA gene expression analysis [3].

3 Graph Regularized Nonnegative Matrix Factorization

By using the nonnegative constraints, NMF can learn a parts-based representation. However, NMF performs this learning in the euclidean space. It fails to discover the intrinsic geometrical and discriminating structure of the data space, which is essential to the real-world applications. In this section, we introduce our GNMF algorithm, which avoids this limitation by incorporating a geometrically based regularizer.

3.1 NMF with Manifold Regularization

Recall that NMF tries to find a set of basis vectors that can be used to best approximate the data. One might further hope that the basis vectors can respect the intrinsic Riemannian structure, rather than ambient euclidean structure. A natural assumption here could be that if two data points $\mathbf{x}_j, \mathbf{x}_l$ are *close* in the *intrinsic* geometry of the data distribution, then $\mathbf{z}_j$ and $\mathbf{z}_l$, the representations of these two points with respect to the new basis, are also close to each other. This assumption is usually referred to as *local invariance assumption* [1], [19], [7], which plays an essential role in the development of various kinds of algorithms, including dimensionality reduction algorithms [1] and semi-supervised learning algorithms [2], [46], [45].

Recent studies in spectral graph theory [9] and manifold learning theory [1] have demonstrated that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter of data points. Consider a graph with N vertices, where each vertex corresponds to a data point. For each data point $\mathbf{x}_j$, we find its p nearest neighbors and put edges between $\mathbf{x}_j$ and its neighbors. There are many choices to define the weight matrix $\mathbf{W}$ on the graph. Three of the most commonly used are as follows:

1. **0-1 Weighting**. $\mathbf{W}_{jl} = 1$, if and only if nodes j and l are connected by an edge. This is the simplest weighting method and is very easy to compute.
2. **Heat Kernel Weighting**. If nodes j and l are connected, put
$$\mathbf{W}_{jl} = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_l\|^2}{\sigma}}.$$
Heat kernel has an intrinsic connection to the Laplace-Beltrami operator on differentiable functions on a manifold [1].
3. **Dot-Product Weighting**. If nodes j and l are connected, put
$$\mathbf{W}_{jl} = \mathbf{x}_j^T \mathbf{x}_l.$$
Note that if $\mathbf{x}$ is normalized to 1, the dot product of two vectors is equivalent to the cosine similarity of the two vectors.

The W_{jl} is used to measure the closeness of two points $\mathbf{x}_j$ and $\mathbf{x}_l$. The different similarity measures are suitable for different situations. For example, the cosine similarity (dot-product weighting) is very popular in the IR community (for processing documents), while for image data, the heat kernel weight may be a better choice. Since W_{jl} in our paper is only for measuring the closeness, we do not treat the different weighting schemes separately.

The low-dimensional representation of $\mathbf{x}_j$ with respect to the new basis is $\mathbf{z}_j = [v_{j1}, \dots, v_{jk}]^T$. Again, we can use either euclidean distance

$$d(\mathbf{z}_j, \mathbf{z}_l) = \|\mathbf{z}_j - \mathbf{z}_l\|^2,$$

or divergence

$$D(\mathbf{z}_j \| \mathbf{z}_l) = \sum_{k=1}^{K} \left(v_{jk} \log \frac{v_{jk}}{v_{lk}} - v_{jk} + v_{lk} \right),$$

to measure the "dissimilarity" between the low-dimensional representations of two data points with respect to the new basis.

With the above defined weight matrix $\mathbf{W}$, we can use the following two terms to measure the smoothness of the low-dimensional representation

$$\begin{aligned} \mathcal{R}_2 &= \frac{1}{2} \sum_{j,l=1}^{N} (D(\mathbf{z}_j \| \mathbf{z}_l) + D(\mathbf{z}_l \| \mathbf{z}_j)) \mathbf{W}_{jl} \\ &= \frac{1}{2} \sum_{j,l=1}^{N} \sum_{k=1}^{K} \left(v_{jk} \log \frac{v_{jk}}{v_{lk}} + v_{lk} \log \frac{v_{lk}}{v_{jk}} \right) \mathbf{W}_{jl}, \end{aligned} \tag{4}$$

and

$$\begin{aligned} \mathcal{R}_1 &= \frac{1}{2} \sum_{j,l=1}^{N} \|\mathbf{z}_j - \mathbf{z}_l\|^2 \mathbf{W}_{jl} \\ &= \sum_{j=1}^{N} \mathbf{z}_j^T \mathbf{z}_j \mathbf{D}_{jj} - \sum_{j,l=1}^{N} \mathbf{z}_j^T \mathbf{z}_l \mathbf{W}_{jl} \\ &= \mathrm{Tr}(\mathbf{V}^T \mathbf{D} \mathbf{V}) - \mathrm{Tr}(\mathbf{V}^T \mathbf{W} \mathbf{V}) = \mathrm{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}), \end{aligned} \tag{5}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix and $\mathbf{D}$ is a diagonal matrix whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}, \mathbf{D}_{jj} = \sum_l \mathbf{W}_{jl}$. $\mathbf{L} = \mathbf{D} - \mathbf{W}$, which is called graph Laplacian [9].

By minimizing $\mathcal{R}_1$ (or $\mathcal{R}_2$), we expect that if two data points $\mathbf{x}_j$ and $\mathbf{x}_l$ are close (i.e., $\mathbf{W}_{jl}$ is big), $\mathbf{z}_j$ and $\mathbf{z}_l$ are also close to each other. Combining this geometrically-based regularizer with the original NMF objective function leads to our GNMF.

Given a data matrix $\mathbf{X} = [x_{ij}] \in \mathbb{R}^{M \times N}$, our GNMF aims to find two nonnegative matrices $\mathbf{U} = [u_{ik}] \in \mathbb{R}^{M \times K}$ and

$\mathbf{V} = [v_{jk}] \in \mathbb{R}^{N\times K}$. Similarly to NMF, we can also use two "distance" measures here. If the euclidean distance is used, GNMF minimizes the objective function as follows:

$$\mathcal{O}_1 = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \lambda \mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}). \tag{6}$$

If the divergence is used, GNMF minimizes

$$\begin{aligned}\mathcal{O}_2 = &\sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij}\log\frac{x_{ij}}{\sum_{k=1}^{K}u_{ik}v_{jk}} - x_{ij} + \sum_{k=1}^{K}u_{ik}v_{jk}\right)\\ &+\frac{\lambda}{2}\sum_{j=1}^{N}\sum_{l=1}^{N}\sum_{k=1}^{K}\left(v_{jk}\log\frac{v_{jk}}{v_{lk}} + v_{lk}\log\frac{v_{lk}}{v_{jk}}\right)\mathbf{W}_{jl},\end{aligned} \tag{7}$$

where the regularization parameter $\lambda \geq 0$ controls the smoothness of the new representation.

3.2 Updating Rules Minimizing (6)

The objective functions $\mathcal{O}_1$ and $\mathcal{O}_2$ of GNMF in (6) and (7) are not convex in both $\mathbf{U}$ and $\mathbf{V}$ together. Therefore, it is unrealistic to expect an algorithm to find the global minima. In the following, we introduce two iterative algorithms which can achieve local minima.

We first discuss how to minimize the objective function $\mathcal{O}_1$, which can be rewritten as

$$\begin{aligned}\mathcal{O}_1 &= \mathrm{Tr}\big((\mathbf{X}-\mathbf{U}\mathbf{V}^T)(\mathbf{X}-\mathbf{U}\mathbf{V}^T)^T\big) + \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V})\\ &= \mathrm{Tr}\big(\mathbf{X}\mathbf{X}^T\big) - 2\mathrm{Tr}\big(\mathbf{X}\mathbf{V}\mathbf{U}^T\big) + \mathrm{Tr}\big(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T\big)\\ &\quad + \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}),\end{aligned} \tag{8}$$

where the second equality applies the matrix properties $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$ and $\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$. Let ψ_{ik} and ϕ_{jk} be the lagrange multiplier for constraint $u_{ik} \geq 0$ and $v_{jk} \geq 0$, respectively, and $\Psi = [\psi_{ik}]$, $\Phi = [\phi_{jk}]$, the Lagrange $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} = &\mathrm{Tr}\big(\mathbf{X}\mathbf{X}^T\big) - 2\mathrm{Tr}\big(\mathbf{X}\mathbf{V}\mathbf{U}^T\big) + \mathrm{Tr}\big(\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T\big)\\ &+ \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) + \mathrm{Tr}(\Psi\mathbf{U}^T) + \mathrm{Tr}(\Phi\mathbf{V}^T).\end{aligned} \tag{9}$$

The partial derivatives of $\mathcal{L}$ with respect to $\mathbf{U}$ and $\mathbf{V}$ are

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = -2\mathbf{X}\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} + \Psi, \tag{10}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = -2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda\mathbf{L}\mathbf{V} + \Phi. \tag{11}$$

Using the KKT conditions $\psi_{ik}u_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$, we get the following equations for u_{ik} and v_{jk}:

$$-(\mathbf{X}\mathbf{V})_{ik}u_{ik} + (\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik} = 0, \tag{12}$$

$$-(\mathbf{X}^T\mathbf{U})_{jk}v_{jk} + (\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk}v_{jk} + \lambda(\mathbf{L}\mathbf{V})_{jk}v_{jk} = 0. \tag{13}$$

These equations lead to the following updating rules:

$$u_{ik} \leftarrow u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}, \tag{14}$$

$$v_{jk} \leftarrow v_{jk}\frac{(\mathbf{X}^T\mathbf{U} + \lambda\mathbf{W}\mathbf{V})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}}. \tag{15}$$

Regarding these two updating rules, we have the following theorem:

Theorem 1. *The objective function $\mathcal{O}_1$ in (6) is nonincreasing under the updating rules in (14) and (15).*

Please see the Appendix for a detailed proof for the above theorem. Our proof essentially follows the idea in the proof of Lee and Seung's [27] paper for the original NMF. Recent studies [8], [30] show that Lee and Seung's [27] multiplicative algorithm cannot guarantee the convergence to a stationary point. Particularly, Lin [30] suggests minor modifications on Lee and Seung's algorithm, which can converge. Our updating rules in (14) and (15) are essentially similar to the updating rules for NMF, and therefore, Lin's modifications can also be applied.

When $\lambda = 0$, it is easy to check that the updating rules in (14) and (15) reduce to the updating rules of the original NMF.

For the objective function of NMF, it is easy to check that if $\mathbf{U}$ and $\mathbf{V}$ are the solution, then $\mathbf{UD}, \mathbf{VD}^{-1}$ will also form a solution for any positive diagonal matrix $\mathbf{D}$. To eliminate this uncertainty, in practice, people will further require that the euclidean length of each column vector in matrix $\mathbf{U}$ (or $\mathbf{V}$) is 1 [42]. The matrix $\mathbf{V}$ (or $\mathbf{U}$) will be adjusted accordingly so that $\mathbf{U}\mathbf{V}^T$ does not change. This can be achieved by

$$u_{ik} \leftarrow \frac{u_{ik}}{\sqrt{\sum_i u_{ik}^2}}, \qquad v_{jk} \leftarrow v_{jk}\sqrt{\sum_i u_{ik}^2}. \tag{16}$$

Our GNMF also adopts this strategy. After the multiplicative updating procedure converges, we set the euclidean length of each column vector in matrix $\mathbf{U}$ to 1 and adjust the matrix $\mathbf{V}$ so that $\mathbf{U}\mathbf{V}^T$ does not change.

3.3 Connection to Gradient Descent Method

Another general algorithm for minimizing the objective function of GNMF in (6) is gradient descent [25]. For our problem, gradient descent leads to the following additive update rules:

$$u_{ik} \leftarrow u_{ik} + \eta_{ik}\frac{\partial\mathcal{O}_1}{\partial u_{ik}}, \quad v_{jk} \leftarrow v_{jk} + \delta_{jk}\frac{\partial\mathcal{O}_1}{\partial v_{jk}}. \tag{17}$$

The η_{ik} and δ_{jk} are usually referred as step size parameters. As long as η_{ik} and δ_{jk} are sufficiently small, the above updates should reduce $\mathcal{O}_1$ unless $\mathbf{U}$ and $\mathbf{V}$ are at a stationary point.

Generally speaking, it is relatively difficult to set these step size parameters while still maintaining the nonnegativity of u_{ik} and v_{jk}. However, with the special form of the partial derivatives, we can use some tricks to set the step size parameters automatically. Let $\eta_{ik} = -u_{ik}/2(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}$, we have

$$\begin{aligned}u_{ik} + \eta_{ik}\frac{\partial\mathcal{O}_1}{\partial u_{ik}} &= u_{ik} - \frac{u_{ik}}{2(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}\frac{\partial\mathcal{O}_1}{\partial u_{ik}}\\ &= u_{ik} - \frac{u_{ik}}{2(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}(-2(\mathbf{X}\mathbf{V})_{ik} + 2(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik})\\ &= u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}.\end{aligned} \tag{18}$$

TABLE 1
Computational Operation Counts for Each Iteration in NMF and GNMF

	F-norm formulation			
	fladd	flmlt	fldiv	overall
NMF	$2MNK + 2(M+N)K^2$	$2MNK + 2(M+N)K^2 + (M+N)K$	$(M+N)K$	$O(MNK)$
GNMF	$2MNK + 2(M+N)K^2$ $+N(p+3)K$	$2MNK + 2(M+N)K^2 + (M+N)K$ $+N(p+1)K$	$(M+N)K$	$O(MNK)$

	Divergence formulation			
	fladd	flmlt	fldiv	overall
NMF	$4MNK + (M+N)K$	$4MNK + (M+N)K$	$2MN + (M+N)K$	$O(MNK)$
GNMF	$4MNK + (M+2N)K$ $+q(p+4)NK$	$4MNK + (M+N)K$ $+Np + q(p+4)NK$	$2MN + MK$	$\mathrm{O}\Big((\mathrm{M+q(p+4)})\mathrm{NK}\Big)$

fladd: a floating-point addition, flmlt: a floating-point multiplication, fldiv: a floating-point division.
N: the number of sample points, M: the number of features, K: the number of factors.
p: the number of nearest neighbors, q: the number of iterations in Conjugate Gradient (CG).

Similarly, letting $\delta_{jk} = -v_{jk}/2(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}$, we have

$$\begin{aligned} v_{jk} + \delta_{jk}\frac{\partial\mathcal{O}_1}{\partial v_{jk}} &= v_{jk} - \frac{v_{jk}}{2(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}}\frac{\partial\mathcal{O}_1}{\partial v_{jk}} \\ &= v_{jk} - \frac{v_{jk}}{2(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}}(-2(\mathbf{X}^T\mathbf{U})_{jk} \\ &\quad + 2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk} + 2\lambda(\mathbf{L}\mathbf{V})_{jk}) \\ &= v_{jk}\frac{(\mathbf{X}^T\mathbf{U} + \lambda\mathbf{W}\mathbf{V})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}}. \end{aligned} \tag{19}$$

Now, it is clear that the multiplicative updating rules in (14) and (15) are special cases of gradient descent with an automatic step parameter selection. The advantage of multiplicative updating rules is the guarantee of nonnegativity of $\mathbf{U}$ and $\mathbf{V}$. Theorem 1 also guarantees that the multiplicative updating rules in (14) and (15) converge to a local optimum.

3.4 Updating Rules Minimizing (7)

For the divergence formulation of GNMF, we also have two updating rules, which can achieve a local minimum of (7):

$$u_{ik} \leftarrow u_{ik}\frac{\sum_j \big(x_{ij}v_{jk}/\sum_k u_{ik}v_{jk}\big)}{\sum_j v_{jk}}, \tag{20}$$

$$\mathbf{v}_k \leftarrow \left(\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L}\right)^{-1} \begin{bmatrix} v_{1k}\sum_i \big(x_{i1}u_{ik}/\sum_k u_{ik}v_{1k}\big) \\ v_{2k}\sum_i \big(x_{i2}u_{ik}/\sum_k u_{ik}v_{2k}\big) \\ \vdots \\ v_{Nk}\sum_i \big(x_{iN}u_{ik}/\sum_k u_{ik}v_{Nk}\big) \end{bmatrix}, \tag{21}$$

where $\mathbf{v}_k$ is the kth column of $\mathbf{V}$ and $\mathbf{I}$ is an $N \times N$ identity matrix.

Similarly, we have the following theorem:

Theorem 2. *The objective function $\mathcal{O}_2$ in (7) is nonincreasing with the updating rules in (20) and (21). The objective function is invariant under these updates if and only if $\mathbf{U}$ and $\mathbf{V}$ are at a stationary point.*

Please see the Appendix for a detailed proof. The updating rules in this section (minimizing the divergence formulation of (7)) are different from the updating rules in Section 3.2 (minimizing the F-norm formulation). For the divergence formulation of NMF, previous studies [16] successfully analyzed the convergence property of the multiplicative algorithm [27] from EM algorithm's maximum likelihood point of view. Such an analysis is also valid in the GNMF case.

When $\lambda = 0$, it is easy to check that the updating rules in (20) and (21) reduce to the updating rules of the original NMF.

3.5 Computational Complexity Analysis

In this section, we discuss the extra computational cost of our proposed algorithm in comparison to standard NMF. Specifically, we provide the computational complexity analysis of GNMF for both the F-Norm and KL-Divergence formulations.

The common way to express the complexity of one algorithm is using big O notation [10]. However, this is not precise enough to differentiate between the complexities of GNMF and NMF. Thus, we count the arithmetic operations for each algorithm.

Based on the updating rules, it is not hard to count the arithmetic operations of each iteration in NMF. We summarize the result in Table 1. For GNMF, it is important to note that $\mathbf{W}$ is a sparse matrix. If we use a p-nearest neighbor graph, the average nonzero elements on each row of $\mathbf{W}$ is p. Thus, we only need NpK flam (a floating-point addition and multiplication) to compute $\mathbf{W}\mathbf{V}$. We also summarize the arithmetic operations for GNMF in Table 1.

The updating rule (21) in GNMF with the divergence formulation involves inverting a large matrix $\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L}$. In reality, there is no need to actually compute the inversion. We only need to solve the linear equations system as follows:

$$\left(\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L}\right)\mathbf{v}_k = \begin{bmatrix} v_{1k}\sum_i \big(x_{i1}u_{ik}/\sum_k u_{ik}v_{1k}\big) \\ v_{2k}\sum_i \big(x_{i2}u_{ik}/\sum_k u_{ik}v_{2k}\big) \\ \vdots \\ v_{Nk}\sum_i \big(x_{iN}u_{ik}/\sum_k u_{ik}v_{Nk}\big) \end{bmatrix}.$$

Since matrix $\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L}$ is symmetric, positive definite, and sparse, we can use the iterative algorithm CG [20] to solve this linear system of equations very efficiently. In each iteration, CG needs to compute the matrix-vector products in the form of $(\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L})\mathbf{p}$. The remaining work load of CG in each iteration is $4N$ flam. Thus, the time cost of CG in

TABLE 2
Statistics of the Three Data Sets

dataset	size (N)	dimensionality (M)	# of classes (K)
COIL20	1440	1024	20
PIE	2856	1024	68
TDT2	9394	36771	30

each iteration is $pN + 4N$. If CG stops after q iterations, the total time cost is $q(p + 4)N$. CG converges very fast, usually within 20 iterations. Since we need to solve K linear equations systems, the total time cost is $q(p + 4)NK$.

Besides the multiplicative updates, GNMF also needs $O(N^2M)$ to construct the p-nearest neighbor graph. Supposing the multiplicative updates stops after t iterations, the overall cost for NMF (both formulations) is

$$O(tMNK). \tag{22}$$

The overall cost for GNMF with F-norm formulation is

$$O(tMNK + N^2M), \tag{23}$$

and the cost for GNMF with divergence formulation is

$$O(t(M + q(p + 4))NK + N^2M). \tag{24}$$

4 Experimental Results

Previous studies show that NMF is very powerful for clustering, especially in the document clustering and image clustering tasks [42], [37]. It can achieve similar or better performance than most of the state-of-the-art clustering algorithms, including the popular spectral clustering methods [32], [42].

Assume that a document corpus is comprised of K clusters each of which corresponds to a coherent topic. To accurately cluster the given document corpus, it is ideal to project the documents into a K-dimensional semantic space in which each axis corresponds to a particular topic [42]. In this semantic space, each document can be represented as a linear combination of the K topics. Because it is more natural to consider each document as an additive rather than a subtractive mixture of the underlying topics, the combination coefficients should all take nonnegative values [42]. These values can be used to decide the cluster membership. In appearance-based visual analysis, an image may be also associated with some hidden parts. For example, a face image can be thought of as a combination of nose, mouth, eyes, etc. It is also reasonable to require the combination coefficients to be non-negative. This is the main motivation of applying NMF on document and image clustering. In this section, we also evaluate our GNMF algorithm on document and image clustering problems.

For the purpose of reproducibility, we provide the code and data sets at: http://www.zjucadcg.cn/dengcai/GNMF/.

4.1 Data Sets

Three data sets are used in the experiment. Two of them are image data sets and the third one is a document corpus. The important statistics of these data sets are summarized below (see also Table 2):

- The first data set is the COIL20 image library, which contains 32×32 gray scale images of 20 objects viewed from varying angles.
- The second data set is the CMU PIE face database, which contains 32×32 gray scale face images of 68 people. Each person has 42 facial images under different light and illumination conditions.
- The third data set is the NIST Topic Detection and Tracking (TDT2) corpus. The TDT2 corpus consists of data collected during the first half of 1,998 and taken from six sources, including two newswires (APW, NYT), two radio programs (VOA, PRI), and two television programs (CNN, ABC). It consists of 11,201 on-topic documents, which are classified into 96 semantic categories. In this experiment, those documents appearing in two or more categories were removed and only the largest 30 categories were kept, thus leaving us with 9,394 documents in total.

4.2 Compared Algorithms

To demonstrate how the clustering performance can be improved by our method, we compare the following five popular clustering algorithms:

- Canonical K-means clustering method (K-means for short).
- K-means clustering in the Principle Component subspace (PCA, in short). Principle Component Analysis (PCA) [24] is one of the most well-known unsupervised dimensionality reduction algorithms. It is expected that the cluster structure will be more explicit in the principle component subspace. Mathematically, PCA is equivalent to performing SVD on the centered data matrix. On the TDT2 data set, we simply use SVD instead of PCA because the centered data matrix is too large to be fit into memory. Actually, SVD has been very successfully used for document representation (*latent semantic indexing*, [11]). Interestingly, Zha et al. [44] have shown that K-means clustering in the SVD subspace has a close connection to average association [38], which is a popular spectral clustering algorithm. They showed that if the inner product is used to measure the similarity and construct the graph, K-means after SVD is equivalent to average association.
- Normalized Cut [38], one of the typical spectral clustering algorithms (NCut in short).
- NMF-based clustering (NMF in short). We use the F-norm formulation and implement a normalized cut weighted version of NMF, as suggested in [42]. We provide a brief description of normalized cut weighted version of NMF and GNMF in Appendix C. Please refer to [42] for more details.
- GNMF with F-norm formulation, which is the new algorithm proposed in this paper. We use the 0-1 weighting scheme for constructing the p-nearest neighbor graph for its simplicity. The number of nearest neighbors p is set to 5 and the regularization parameter λ is set to 100. The parameter selection and weighting scheme selection will be discussed in a later section.

TABLE 3
Clustering Performance on COIL20

K	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	PCA	NCut	NMF	GNMF	Kmeans	PCA	NCut	NMF	GNMF
4	83.0±15.2	83.1±15.0	89.4±11.1	81.0±14.2	**93.5±10.1**	74.6±18.3	74.4±18.2	83.4±15.1	71.8±18.4	**90.9±12.7**
6	74.5±10.3	75.5±12.2	83.6±11.3	74.3±10.1	**92.4±6.1**	73.2±11.4	73.1±12.1	80.9±11.6	71.9±11.6	**91.1±5.6**
8	68.6±5.7	70.4±9.3	79.1±7.7	69.3±8.6	**84.0±9.6**	71.8±6.8	72.8±8.3	79.1±6.6	71.0±7.4	**89.0±6.5**
10	69.6±8.0	70.8±7.2	79.4±7.6	69.4±7.6	**84.4±4.9**	75.0±6.2	75.1±5.2	81.3±6.0	73.9±5.7	**89.2±3.3**
12	65.0±6.8	64.3±4.6	74.9±5.5	69.0±6.3	**81.0±8.3**	73.1±5.6	72.5±4.6	78.6±5.1	73.3±5.5	**88.0±4.9**
14	64.0±4.9	67.3±6.2	71.5±5.6	67.6±5.6	**79.2±5.2**	73.3±4.2	74.9±4.9	78.1±3.8	73.8±4.6	**87.3±3.0**
16	64.0±4.9	64.1±4.9	70.7±4.1	66.0±6.0	**76.8±4.1**	74.6±3.1	74.5±2.7	78.0±2.8	73.4±4.2	**86.5±2.0**
18	62.7±4.7	62.3±4.3	67.2±4.1	62.8±3.7	**76.0±3.0**	73.7±2.6	73.9±2.5	76.3±3.0	72.4±2.4	**85.8±1.8**
20	63.7	64.3	69.6	60.5	**75.3**	73.4	74.5	77.0	72.5	**87.5**
Avg.	68.3	69.1	76.2	68.9	**82.5**	73.6	74.0	79.2	72.7	**88.4**

TABLE 4
Clustering Performance on PIE

K	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	PCA	NCut	NMF	GNMF	Kmeans	PCA	NCut	NMF	GNMF
10	29.0±3.7	29.8±3.3	**82.5±8.6**	57.8±6.3	80.3±8.7	34.8±4.1	35.8±3.9	**88.0±5.2**	66.2±4.0	86.1±5.5
20	27.9±2.2	27.7±2.4	75.9±4.4	62.0±3.5	**79.5±5.2**	44.9±2.4	44.7±2.8	84.8±2.4	77.2±1.7	**88.0±2.8**
30	26.1±1.3	26.5±1.7	74.4±3.6	63.3±3.7	**78.9±4.5**	48.4±1.8	48.8±1.5	84.3±1.2	80.4±1.1	**89.1±1.6**
40	25.4±1.4	25.6±1.6	70.4±2.9	63.7±2.4	**77.1±3.2**	50.9±1.7	50.9±1.8	82.3±1.2	82.0±1.1	**88.6±1.2**
50	25.0±0.8	24.6±1.0	68.2±2.2	65.2±2.9	**75.7±3.0**	52.6±0.8	51.9±1.3	81.6±1.0	83.4±0.9	**88.8±1.1**
60	24.2±0.8	24.6±0.7	67.7±2.1	65.1±1.4	**74.6±2.7**	53.0±1.0	53.4±0.9	80.9±0.6	84.1±0.5	**88.7±0.9**
68	23.9	25.0	65.9	66.2	**75.4**	55.1	54.7	80.3	85.8	**88.6**
Avg	25.9	26.3	73.6	63.3	**77.4**	48.5	48.6	83.6	79.9	**88.3**

TABLE 5
Clustering Performance on TDT2

K	Accuracy (%)					Normalized Mutual Information (%)				
	Kmeans	SVD	NCut	NMF	GNMF	Kmeans	SVD	NCut	NMF	GNMF
5	80.8±17.5	82.7±16.0	96.4±0.7	95.5±10.2	**98.5±2.8**	78.1±19.0	76.8±20.3	93.1±3.9	92.7±14.0	**94.2±8.9**
10	68.5±15.3	68.2±13.6	88.2±10.8	83.6±12.2	**91.4±7.6**	73.1±13.5	69.2±14.0	83.4±11.1	82.4±11.9	**85.6±9.2**
15	64.9±8.7	65.3±7.2	82.1±11.2	79.9±11.7	**93.4±2.7**	74.0±7.9	71.8±8.9	81.1±9.8	82.0±9.2	**88.0±5.7**
20	63.9±4.2	63.4±5.5	79.0±8.1	76.3±5.6	**91.2±2.6**	75.7±4.5	71.5±5.6	78.9±6.3	80.6±4.5	**85.9±4.1**
25	61.5±4.3	60.8±4.0	74.3±4.8	75.0±4.5	**88.6±2.1**	74.6±2.4	70.9±2.3	77.1±2.7	79.0±2.5	**83.9±2.6**
30	61.2	65.9	71.2	71.9	**88.6**	74.7	74.7	76.5	77.4	**83.7**
Avg.	66.8	67.7	81.9	80.4	**92.0**	75.0	72.5	81.7	82.4	**86.9**

Among these five algorithms, NMF and GNMF can learn a parts-based representation because they allow only additive, not subtractive, combinations. NCut and GNMF are the two approaches, which consider the intrinsic geometrical structure of the data.

The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. Two metrics, the accuracy (AC) and the normalized mutual information metric (NMI) are used to measure the clustering performance. Please see [4] for the detailed definitions of these two metrics.

4.3 Clustering Results

Tables 3, 4, and 5 show the clustering results on the COIL20, PIE, and TDT2 data sets, respectively. In order to randomize the experiments, we conduct the evaluations with different cluster numbers. For each given cluster number K, 20 test runs were conducted on different randomly chosen clusters (except the case when the entire data set is used). The mean and standard error of the performance are reported in the tables.

These experiments reveal a number of interesting points:

- The NMF-based methods, both NMF and GNMF, outperform the PCA (SVD) method, which suggests the superiority of the parts-based representation idea in discovering the hidden factors.
- Both NCut and GNMF consider the geometrical structure of the data and achieve better performance than the other three algorithms. This suggests the importance of the geometrical structure in learning the hidden factors.
- Regardless of the data sets, our GNMF always results in the best performance. This shows that by leveraging the power of both the parts-based representation and graph Laplacian regularization, GNMF can learn a better compact representation.

4.4 Parameters Selection

Our GNMF model has two essential parameters: the number of nearest neighbors p and the regularization parameter λ. Figs. 1 and 2 show how the average performance of GNMF varies with the parameters λ and p, respectively.

As we can see, the performance of GNMF is very stable with respect to the parameter λ. GNMF achieves consistently good performance when λ varies from 10 to 1,000 on all three data sets.

As we have described, GNMF uses a p-nearest graph to capture the local geometric structure of the data distribution.

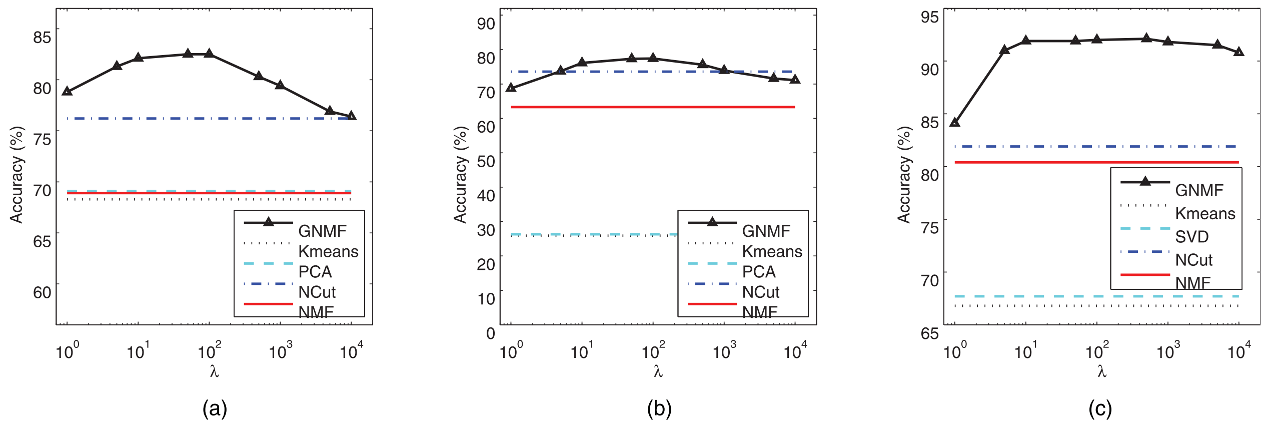


Fig. 1. The performance of GNMF versus parameter λ. The GNMF is stable with respect to the parameter λ. It achieves consistently good performance when λ varies from 10 to 1,000. (a) COIL20, (b) PIE, and (c) TDT2.

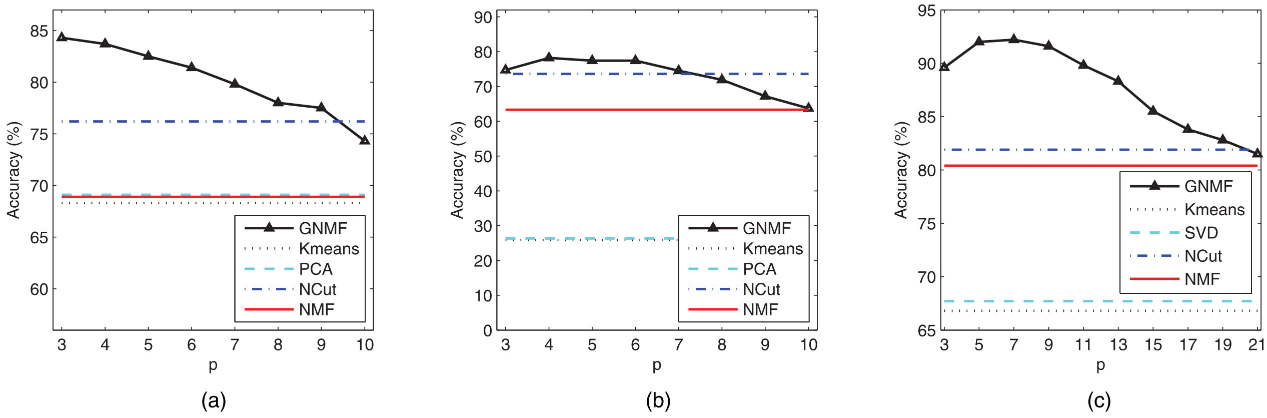


Fig. 2. The performance of GNMF decreases as p increases. (a) COIL20, (b) PIE, and (c) TDT2.

The success of GNMF relies on the assumption that two neighboring data points share the same label. Obviously, this assumption is more likely to fail as p increases. This is the reason why the performance of GNMF decreases as p increases, as shown in Fig. 2.

4.5 Weighting Scheme Selection

There are many choices on how to define the weight matrix $\mathbf{W}$ on the p-nearest neighbor graph. The three most popular ones are 0-1 weighting, heat kernel weighting, and dot-product weighting. In our previous experiment, we use 0-1 weighting for its simplicity. Given a point $\mathbf{x}$, 0-1 weighting treats its p nearest neighbors equally important. However, in many cases, it is necessary to differentiate these p neighbors, especially when p is large. In this case, one can use heat kernel weighting or dot-product weighting.

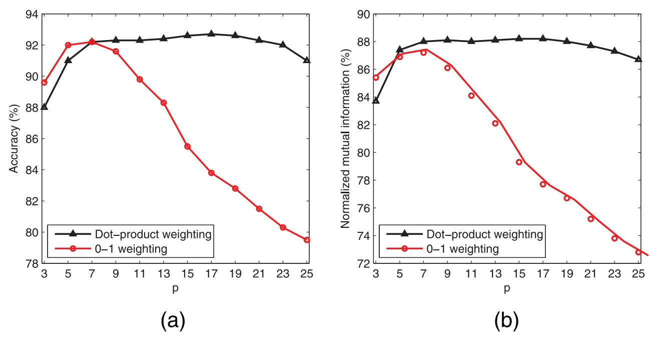


Fig. 3. The performance of GNMF versus parameter λ with different weighting schemes (dot-product versus 0-1 weighting) on the TDT2 data set. (a) AC, (b) NMI.

For text analysis tasks, the document vectors usually have been normalized to unit. In this case, the dot-product of two document vectors becomes their cosine similarity, which is a widely used similarity measure for document in information retrieval community. Thus, it is very natural to use dot-product weighting for text data. Similar to 0-1 weighting, there is also no parameter for dot-product weighting. Fig. 3 shows the performance of GNMF as a function of the number of nearest neighbors p for both dot-product and 0-1 weighting schemes on TDT2 data set. It is clear that dot-product weighting performs better than 0-1 weighting, especially when p is large. For dot-product weighting, the performance of GNMF remains reasonably good as p increases to 23, whereas, the performance of GNMF decreases dramatically for 0-1 weighting as p increases (when larger than 9).

For image data, a reasonable weighting scheme is heat kernel weighting. Fig. 4 shows the performance of GNMF as a function of the number of nearest neighbors p for the heat kernel and 0-1 weighting schemes on the COIL20 data set. We can see that heat kernel weighting is also superior to 0-1 weighting, especially when p is large. However, there is a parameter σ in heat kernel weighting which is very crucial to the performance. Automatically selecting σ in heat kernel weighting is a challenging problem and has received a lot of interest in recent studies. A more detailed analysis of this

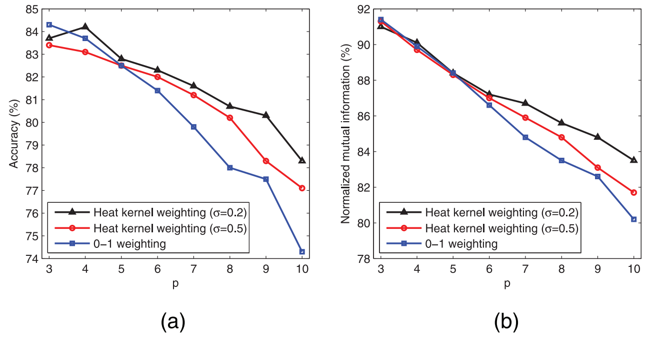


Fig. 4. The performance of GNMF versus the parameter λ with different weighting schemes (heat kernel versus 0-1 weighting) on the COIL20 data set. (a) AC, (b) NMI.

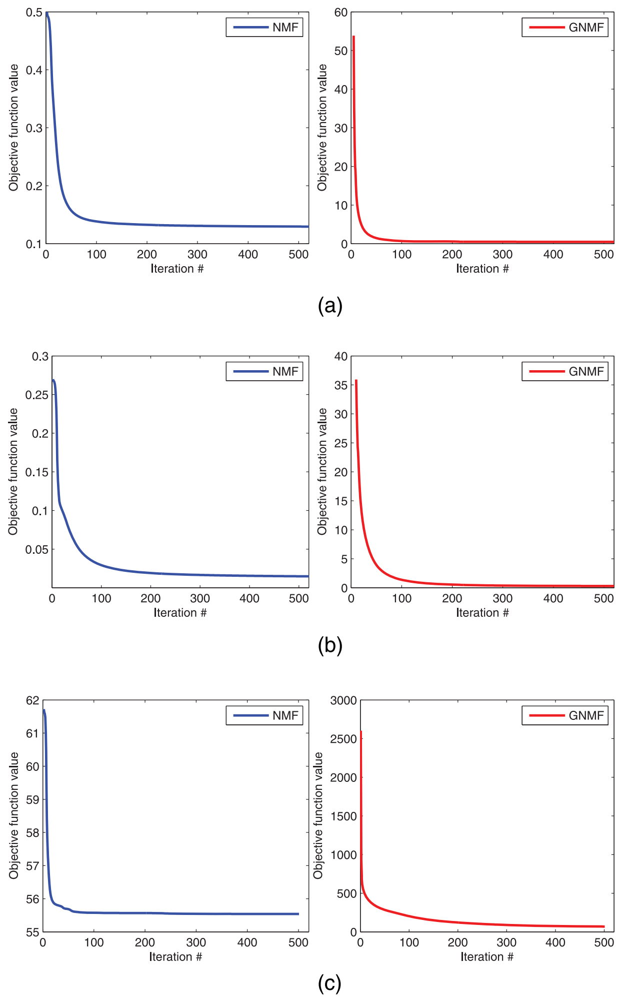


Fig. 5. Convergence curve of NMF and GNMF. (a) COIL20, (b) PIE, and (c) TDT2.

subject is beyond the scope of this paper. Interested readers can refer to [43] for more details.

4.6 Convergence Study

The updating rules for minimizing the objective function of GNMF are essentially iterative. We have proven that these rules are convergent. Here, we investigate how fast these rules can converge.

Fig. 5 shows the convergence curves of both NMF and GNMF on all the three data sets. For each figure, the y-axis is the value of objective function and the x-axis denotes the iteration number. We can see that the multiplicative update rules for both GNMF and NMF converge very fast, usually within 100 iterations.

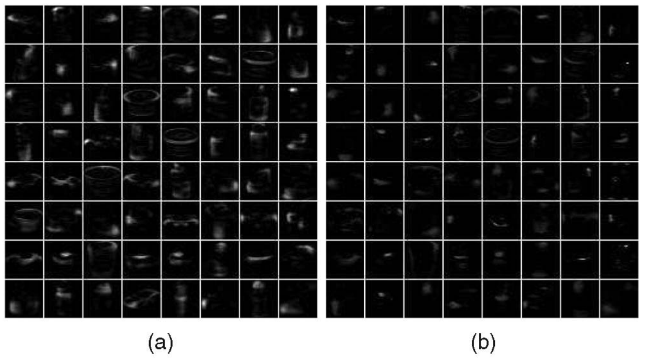


Fig. 6. Basis vectors learned from the COIL20 data set. (a) Basis vectors (column vectors of $\mathbf{U}$) learned by NMF. (b) Basis vectors (column vectors of $\mathbf{U}$) learned by GNMF.

4.7 Sparseness Study

NMF only allows additive combinations between the basis vectors and it is believed that this property enables NMF to learn a *parts-based* representation [26]. Recent studies show that NMF does not always result in parts-based representations [22], [23]. Several researchers addressed this problem by incorporating the sparseness constraints on $\mathbf{U}$ and/or $\mathbf{V}$ [23]. In this section, we investigate the sparseness of the basis vectors learned in GNMF.

Figs. 6 and 7 show the basis vectors learned by NMF and GNMF in the COIL20 and PIE data sets, respectively. Each basis vector has dimensionality 1,024 and has unit norm. We plot these basis vectors as 32×32 gray scale images. It is clear to see that the basis vectors learned by GNMF are sparser than those learned by NMF. This result suggests that GNMF can learn a better *parts-based* representation than NMF.

5 Conclusions and Future Work

We have presented a novel method for matrix factorization, called GNMF. GNMF models the data space as a submanifold embedded in the ambient space and performs the NMF

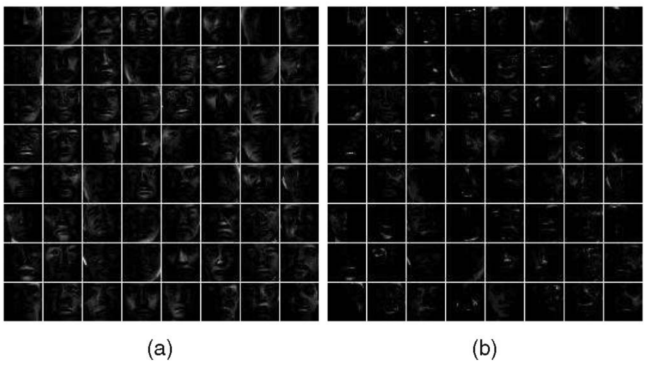


Fig. 7. Basis vectors learned from the PIE data set. (a) Basis vectors (column vectors of $\mathbf{U}$) learned by NMF. (b) Basis vectors (column vectors of $\mathbf{U}$) learned by GNMF.

on this manifold. As a result, GNMF can have more discriminating power than the ordinary NMF approach, which only considers the euclidean structure of the data. Experimental results on document and image clustering show that GNMF provides a better representation in the sense of semantic structure.

Several questions remain to be investigated in our future work:

1. There is a parameter λ which controls the smoothness of our GNMF model. GNMF boils down to original NMF when $\lambda = 0$. Thus, a suitable value of λ is critical to our algorithm. It remains unclear how to do model selection theoretically and efficiently.
2. NMF is an optimization of convex cone structure [14]. Instead of preserving the locality of close points in a euclidean manner, preserving the locality of angle similarity might fit more to the NMF framework. This suggests another way to extend NMF.
3. Our convergence proofs essentially follows the idea in the proofs of Lee and Seung's [27] paper for the original NMF. For the F-norm formulation, Lin [30] shows that Lee and Seung's multiplicative algorithm cannot guarantee the convergence to a stationary point and suggests minor modifications on Lee and Seung's algorithm, which can converge. Our updating rules in (14) and (15) are essentially similar to the updating rules for NMF. It is interesting to apply Lin's idea to GNMF approach.

Appendix A
Proofs of Theorem 1

The objective function $\mathcal{O}_1$ of GNMF in (6) is certainly bounded from below by zero. To prove Theorem 1, we need to show that $\mathcal{O}_1$ is nonincreasing under the updating steps in (14) and (15). Since the second term of $\mathcal{O}_1$ is only related to $\mathbf{V}$, we have exactly the same update formula for $\mathbf{U}$ in GNMF as in the original NMF. Thus, we can use the convergence proof of NMF to show that $\mathcal{O}_1$ is nonincreasing under the update step in (14). Please see [27] for details.

Now, we only need to prove that $\mathcal{O}_1$ is nonincreasing under the updating step in (15). We will follow a procedure similar to that described in [27]. Our proof will make use of an auxiliary function similar to that used in the expectation-maximization algorithm [12]. We begin with the definition of the *auxiliary function*.

Definition 1. *$G(v, v')$ is an* auxiliary function *for $F(v)$ if the conditions*

$$G(v, v') \geq F(v), \qquad G(v, v) = F(v),$$

are satisfied.

The auxiliary function is very useful because of the following lemma.

Lemma 3. *If G is an auxiliary function of F, then F is nonincreasing under the update*

$$v^{(t+1)} = \arg\min_v G(v, v^{(t)}). \tag{25}$$

Proof.

$$F(v^{(t+1)}) \leq G(v^{(t+1)}, v^{(t)}) \leq G(v^{(t)}, v^{(t)}) = F(v^{(t)}).$$

□

Now, we will show that the update step for $\mathbf{V}$ in (15) is exactly the update in (25) with a proper auxiliary function.

We rewrite the objective function $\mathcal{O}_1$ of GNMF in (6) as follows:

$$\begin{aligned}\mathcal{O}_1 &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 + \lambda \mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\left(x_{ij} - \sum_{k=1}^{K} u_{ik}v_{jk}\right)^2 + \lambda \sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{N} v_{jk}L_{jl}v_{lk}.\end{aligned} \tag{26}$$

Considering any element v_{ab} in $\mathbf{V}$, we use F_{ab} to denote the part of $\mathcal{O}_1$, which is only relevant to v_{ab}. It is easy to check that

$$F'_{ab} = \left(\frac{\partial \mathcal{O}_1}{\partial \mathbf{V}}\right)_{ab} = (-2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\lambda\mathbf{L}\mathbf{V})_{ab}, \tag{27}$$

$$F''_{ab} = 2\left(\mathbf{U}^T\mathbf{U}\right)_{bb} + 2\lambda\mathbf{L}_{aa}. \tag{28}$$

Since our update is essentially element wise, it is sufficient to show that each F_{ab} is nonincreasing under the update step of (15).

Lemma 4. *Function*

$$\begin{aligned}G\left(v, v_{ab}^{(t)}\right) &= F_{ab}\left(v_{ab}^{(t)}\right) + F'_{ab}\left(v_{ab}^{(t)}\right)\left(v - v_{ab}^{(t)}\right) \\ &\quad + \frac{\left(\mathbf{V}\mathbf{U}^T\mathbf{U}\right)_{ab} + \lambda\left(\mathbf{D}\mathbf{V}\right)_{ab}}{v_{ab}^{(t)}}\left(v - v_{ab}^{(t)}\right)^2\end{aligned} \tag{29}$$

is an auxiliary function for F_{ab}, the part of $\mathcal{O}_1$, which is only relevant to v_{ab}.

Proof. Since $G(v, v) = F_{ab}(v)$ is obvious, we need only show that $G(v, v_{ab}^{(t)}) \geq F_{ab}(v)$. To do this, we compare the Taylor series expansion of $F_{ab}(v)$,

$$\begin{aligned}F_{ab}(v) &= F_{ab}\left(v_{ab}^{(t)}\right) + F'_{ab}\left(v_{ab}^{(t)}\right)\left(v - v_{ab}^{(t)}\right) \\ &\quad + \left[\left(\mathbf{U}^T\mathbf{U}\right)_{bb} + \lambda\mathbf{L}_{aa}\right]\left(v - v_{ab}^{(t)}\right)^2,\end{aligned} \tag{30}$$

with (29) to find that $G(v, v_{ab}^{(t)}) \geq F_{ab}(v)$ is equivalent to

$$\frac{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + \lambda(\mathbf{D}\mathbf{V})_{ab}}{v_{ab}^{(t)}} \geq (\mathbf{U}^T\mathbf{U})_{bb} + \lambda\mathbf{L}_{aa}. \tag{31}$$

We have

$$(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} = \sum_{l=1}^{k} v_{al}^{(t)}(\mathbf{U}^T\mathbf{U})_{lb} \geq v_{ab}^{(t)}(\mathbf{U}^T\mathbf{U})_{bb}, \tag{32}$$

and

$$\begin{aligned}\lambda(\mathbf{D}\mathbf{V})_{ab} &= \lambda\sum_{j=1}^{M}\mathbf{D}_{aj}v_{jb}^{(t)} \geq \lambda\mathbf{D}_{aa}v_{ab}^{(t)} \\ &\geq \lambda(\mathbf{D} - \mathbf{W})_{aa}v_{ab}^{(t)} = \lambda\mathbf{L}_{aa}v_{ab}^{(t)}.\end{aligned} \tag{33}$$

Thus, (31) holds and $G(v, v_{ab}^{(t)}) \geq F_{ab}(v)$. □

We can now demonstrate the convergence of Theorem 1.

Proof of Theorem 1. Replacing $G(v, v_{ab}^{(t)})$ in (25) by (29) results in the update rule

$$\begin{aligned} v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)} \frac{F'_{ab}\big(v_{ab}^{(t)}\big)}{2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + 2\lambda(\mathbf{D}\mathbf{V})_{ab}} \\ &= v_{ab}^{(t)} \frac{(\mathbf{X}^T\mathbf{U} + \lambda\mathbf{W}\mathbf{V})_{ab}}{(\mathbf{V}\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{ab}}. \end{aligned} \tag{34}$$

Since (29) is an auxiliary function, F_{ab} is nonincreasing under this update rule. □

Appendix B

Proofs of Theorem 2

Similarly, the second term of $\mathcal{O}_2$ in (7) is only related to $\mathbf{V}$; we have exactly the same update formula for $\mathbf{U}$ in GNMF as the original NMF. Thus, we can use the convergence proof of NMF to show that $\mathcal{O}_2$ is nonincreasing under the update step in (20). Please see [27] for details.

Now, we will show that the update step for $\mathbf{V}$ in (21) is exactly the update in (25) with a proper auxiliary function.

Lemma 5. *Function*

$$\begin{aligned} &G(\mathbf{V}, \mathbf{V}^{(t)}) \\ &= \sum_{i,j}\left(x_{ij}\log x_{ij} - x_{ij} + \sum_{k=1}^{K} u_{ik}v_{jk}\right) \\ &- \sum_{i,j,k}\left(x_{ij}\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\left(\log u_{ik}v_{jk} - \log\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\right)\right) \\ &+ \frac{\lambda}{2}\sum_{j,l,k}\left(v_{jk}\log\frac{v_{jk}}{v_{lk}} + v_{lk}\log\frac{v_{lk}}{v_{jk}}\right)\mathbf{W}_{jl}, \end{aligned}$$

is an auxiliary function for the objective function of GNMF in (7)

$$\begin{aligned} F(\mathbf{V}) = &\sum_{i,j}\left(x_{ij}\log\frac{x_{ij}}{\sum_k u_{ik}v_{jk}} - x_{ij} + \sum_k u_{ik}v_{jk}\right) \\ &+ \frac{\lambda}{2}\sum_{j,l,k}\left(v_{jk}\log\frac{v_{jk}}{v_{lk}} + v_{lk}\log\frac{v_{lk}}{v_{jk}}\right)\mathbf{W}_{jl}. \end{aligned}$$

Proof. It is straightforward to verify that $G(\mathbf{V},\mathbf{V}) = F(\mathbf{V})$. To show that $G(\mathbf{V},\mathbf{V}^{(t)}) \geq F(\mathbf{V})$, we use convexity of the log function to derive the inequality

$$-\log\left(\sum_{k=1}^{K} u_{ik}v_{jk}\right) \leq -\sum_{k=1}^{K}\left(\alpha_k \log\frac{u_{ik}v_{jk}}{\alpha_k}\right),$$

which holds for all nonnegative α_k that sum to unity. Setting

$$\alpha_k = \frac{u_{ik}v_{jk}^{(t)}}{\sum_{k=1}^{K} u_{ik}v_{jk}^{(t)}},$$

we obtain

$$\begin{aligned} &-\log\left(\sum_k u_{ik}v_{jk}\right) \leq \\ &-\sum_k\left(\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\left(\log u_{ik}v_{jk} - \log\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\right)\right). \end{aligned}$$

From this inequality, it follows that $G(\mathbf{V},\mathbf{V}^{(t)}) \geq F(\mathbf{V})$. □

Theorem 2, then follows from the application of Lemma 5:

Proof of Theorem 2. The minimum of $G(\mathbf{V},\mathbf{V}^{(t)})$ with respect to $\mathbf{V}$ is determined by setting the gradient to zero:

$$\begin{aligned} &\sum_{i=1}^{M} u_{ik} - \sum_{i=1}^{M} x_{ij}\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\frac{1}{v_{jk}} \\ &\quad + \frac{\lambda}{2}\sum_{l=1}^{N}\left(\log\frac{v_{jk}}{v_{lk}} + 1 - \frac{v_{lk}}{v_{jk}}\right)\mathbf{W}_{jl} = 0 \\ &\quad 1 \leq j \leq N, 1 \leq k \leq K. \end{aligned} \tag{35}$$

Because of the log term, it is really hard to solve the above system of equations. Let us recall the motivation of the regularization term. We hope that if two data points $\mathbf{x}_j$ and $\mathbf{x}_r$ are close (i.e., $\mathbf{W}_{jr}$ is big), $\mathbf{z}_j$ will be close to $\mathbf{z}_r$ and v_{js}/v_{rs} will be approximately 1. Thus, we can use the following approximation:

$$\log(x) \approx 1 - \frac{1}{x}, \quad x \rightarrow 1.$$

The above approximation is based on the first order expansion of Taylor series of the log function. With this approximation, the equations in (35) can be written as

$$\begin{aligned} &\sum_{i=1}^{M} u_{ik} - \sum_{i=1}^{M} x_{ij}\frac{u_{ik}v_{jk}^{(t)}}{\sum_k u_{ik}v_{jk}^{(t)}}\frac{1}{v_{jk}} \\ &\quad + \frac{\lambda}{v_{jk}}\sum_{l=1}^{N}(v_{jk} - v_{lk})\mathbf{W}_{jl} = 0 \\ &\quad 1 \leq j \leq N, \quad 1 \leq k \leq K. \end{aligned} \tag{36}$$

Let $\mathbf{D}$ denote a diagonal matrix whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $\mathbf{D}_{jj} = \sum_l \mathbf{W}_{jl}$. Define $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Let $\mathbf{v}_k$ denote the kth column of $\mathbf{V}, \mathbf{v}_k = [v_{1k}, \ldots, v_{Nk}]^T$. It is easy to verify that $\sum_l (v_{jl} - v_{lk})\mathbf{W}_{jl}$ equals the jth element of vector $\mathbf{L}\mathbf{v}_k$.

The system of equations in (36) can be rewritten as

$$\begin{aligned} &\sum_i u_{ik}\mathbf{I}\mathbf{v}_k + \lambda\mathbf{L}\mathbf{v}_k = \begin{bmatrix} v_{1k}^{(t)}\sum_i\left(x_{i1}u_{ik}/\sum_k u_{ik}v_{1k}^{(t)}\right) \\ \vdots \\ v_{Nk}^{(t)}\sum_i\left(x_{iN}u_{ik}/\sum_k u_{ik}v_{Nk}^{(t)}\right) \end{bmatrix}, \\ &1 \leq k \leq K. \end{aligned}$$

Thus, the update rule of (25) takes the form

$$\mathbf{v}_k^{(t+1)} = \left(\sum_i u_{ik}\mathbf{I} + \lambda\mathbf{L}\right)^{-1} \begin{bmatrix} v_{1k}^{(t)} \sum_i \left(x_{i1}u_{ik} / \sum_k u_{ik}v_{1k}^{(t)}\right) \\ \vdots \\ v_{Nk}^{(t)} \sum_i \left(x_{iN}u_{ik} / \sum_k u_{ik}v_{Nk}^{(t)}\right) \end{bmatrix}, \quad 1 \le k \le K.$$

Since G is an auxiliary function, F is nonincreasing under this update. □

Appendix C
Weighted NMF and GNMF

In this appendix, we provide a brief description of normalized cut weighted NMF, which was first introduced by Xu et al. [42]. Letting $\mathbf{z}_j^T$ be the jth row vector of $\mathbf{V}$, the objective function of NMF can be written as

$$O = \sum_{j=1}^{N} \left(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j\right)^T \left(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j\right),$$

which is the summation of the reconstruction errors over all of the data points, and each data point is equally weighted. If each data point has weight γ_j, the objective function of weighted NMF can be written as

$$\begin{aligned} O' &= \sum_{j=1}^{N} \gamma_j(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j)^T(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j) \\ &= \mathrm{Tr}((\mathbf{X} - \mathbf{U}\mathbf{V}^T)\Gamma(\mathbf{X} - \mathbf{U}\mathbf{V}^T)^T) \\ &= \mathrm{Tr}((\mathbf{X}\Gamma^{1/2} - \mathbf{U}\mathbf{V}^T\Gamma^{1/2})(\mathbf{X}\Gamma^{1/2} - \mathbf{U}\mathbf{V}^T\Gamma^{1/2})^T) \\ &= \mathrm{Tr}((\mathbf{X}' - \mathbf{U}\mathbf{V}'^T)^T(\mathbf{X}' - \mathbf{U}\mathbf{V}'^T)), \end{aligned}$$

where Γ is the diagonal matrix consists of γ_j, $\mathbf{V}' = \Gamma^{1/2}\mathbf{V}$, and $\mathbf{X}' = \mathbf{X}\Gamma^{1/2}$. Notice that the above equation has the same form as (1) in Section 2 (the objective function of NMF), so the same algorithm for NMF can be used to find the solution of this weighted NMF problem. In [42], Xu et al. calculate $\mathbf{D} = diag(\mathbf{X}^T\mathbf{X}\mathbf{e})$, where $\mathbf{e}$ is a vector of all ones. They use $\mathbf{D}^{-1}$ as the weight and named this approach as normalized cut weighted NMF (NMF-NCW). The experimental results [42] have demonstrated the effectiveness of this weighted approach on document clustering.

Similarly, we can also introduce this weighting scheme into our GNMF approach. The objective function of weighted GNMF is

$$\begin{aligned} \mathcal{O}' &= \sum_{j=1}^{N} \gamma_j(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j)^T(\mathbf{x}_j - \mathbf{U}\mathbf{z}_j) + \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}((\mathbf{X} - \mathbf{U}\mathbf{V}^T)\Gamma(\mathbf{X} - \mathbf{U}\mathbf{V}^T)^T) + \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}((\mathbf{X}\Gamma^{1/2} - \mathbf{U}\mathbf{V}^T\Gamma^{1/2})(\mathbf{X}\Gamma^{1/2} - \mathbf{U}\mathbf{V}^T\Gamma^{1/2})^T) \\ &\quad + \lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}((\mathbf{X}' - \mathbf{U}\mathbf{V}'^T)^T(\mathbf{X}' - \mathbf{U}\mathbf{V}'^T)) + \lambda\mathrm{Tr}(\mathbf{V}'^T\mathbf{L}'\mathbf{V}'), \end{aligned}$$

where $\Gamma, \mathbf{V}'$, and $\mathbf{X}'$ are defined as before and $\mathbf{L}' = \Gamma^{-1/2}\mathbf{L}\Gamma^{-1/2}$. Notice that the above equation has the same form as (8) in Section 3.2, so the same algorithm for GNMF can be used to find the solution of weighted GNMF problem.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 60905001 and 90920303, the National Key Basic Research Foundation of China under Grant 2009CB320801, US National Science Foundation IIS-09-05215, and the US Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

References

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

[2] 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.

[3] J.-P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov, "Metagenes and Molecular Pattern Discovery Using Matrix Factorization," *Nat'l Academy of Sciences,* vol. 101, no. 12, pp. 4164-4169, 2004.

[4] D. Cai, X. He, and J. Han, "Document Clustering Using Locality Preserving Indexing," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 12, pp. 1624-1637, Dec. 2005.

[5] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality Preserving Nonnegative Matrix Factorization," *Proc. Int'l Joint Conf. Artificial Intelligence,* 2009.

[6] D. Cai, X. He, X. Wu, and J. Han, "Non-Negative Matrix Factorization on Manifold," *Proc. Int'l Conf. Data Mining,* 2008.

[7] D. Cai, X. Wang, and X. He, "Probabilistic Dyadic Data Analysis with Local and Global Consistency," *Proc. 26th Ann. Int'l Conf. Machine Learning,* pp. 105-112, 2009.

[8] M. Catral, L. Han, M. Neumann, and R. Plemmons, "On Reduced Rank Nonnegative Matrix Factorization for Symmetric Nonnegative Matrices," *Linear Algebra and Its Applications,* vol. 393, pp. 107-126, 2004.

[9] F.R.K. Chung, *Spectral Graph Theory.* Am. Math. Soc., 1997.

[10] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms,* second ed. MIT Press and McGraw-Hill, 2001.

[11] S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, and R.A. harshman, "Indexing by Latent Semantic Analysis," *J. Am. Soc. of Information Science,* vol. 41, no. 6, pp. 391-407, 1990.

[12] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the Em Algorithm," *J. Royal Statistical Soc. Series B (Methodological),* vol. 39, no. 1, pp. 1-38, 1977.

[13] C. Ding, T. Li, and W. Peng, "Nonnegative Matrix Factorization and Probabilistic Latent Semantic Indexing: Equivalence, Chi-Square Statistic, and a Hybrid Method," *Proc. Nat'l Conf. American Association for Artificial Intelligence,* 2006.

[14] D. Donoho and V. Stodden, "When Does Non-Negative Matrix Factorization Give a Correct Decomposition into Parts?" *Advances in Neural Information Processing Systems 16,* MIT Press, 2003.

[15] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification,* second ed. Wiley-Interscience, 2000.

[16] L. Finesso and P. Spreij, "Nonnegative Matrix Factorization and I-Divergence Alternating Minimization," *Linear Algebra and Its Applications,* vol. 416, nos. 2/3, pp. 270-287, 2006.

[17] E. Gaussier and C. Goutte, "Relation between PLSA and NFM and Implications," *Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 601-602, 2005.

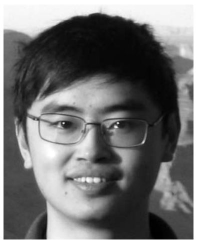
[18] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," *Proc. IEEE CS. Conf. Computer Vision and Pattern Recognition,* pp. 1735-1742, 2006.

[19] X. He and P. Niyogi, "Locality Preserving Projections," *Advances in Neural Information Processing Systems 16,* MIT Press, 2003.

[20] M.R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *J. Research of the Nat'l Bureau of Standards,* vol. 49, no. 6, pp. 409-436, 1952.

[21] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning,* vol. 42, nos. 1/2, pp. 177-196, 2001.
[22] P.O. Hoyer, "Non-Negative Sparse Coding," *Proc. IEEE Workshop Neural Networks for Signal Processing,* pp. 557-565, 2002.
[23] P.O. Hoyer, "Non-Negative Matrix Factorization with Sparseness Constraints," *J. Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.
[24] I.T. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1989.
[25] J. Kivinen and M.K. Warmuth, "Additive versus Exponentiated Gradient Updates for Linear Prediction," *Proc. Ann. ACM Symp. Theory of Computing,* pp. 209-218, 1995.
[26] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Non-Negative Matrix Factorization," *Nature,* vol. 401, pp. 788-791, 1999.
[27] D.D. Lee and H.S. Seung, "Algorithms for Non-Negative Matrix Factorization," *Advances in Neural Information Processing Systems 13,* MIT Press, 2001.
[28] J.M. Lee, *Introduction to Smooth Manifolds.* Springer-Verlag, 2002.
[29] S.Z. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning Spatially Localized, Parts-Based Representation," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 207-212, 2001.
[30] C.-J. Lin, "On the Convergence of Multiplicative Update Algorithms for Non-Negative Matrix Factorization," *IEEE Trans. Neural Networks,* vol. 18, no. 6, pp. 1589-1596, Nov. 2007.
[31] N.K. Logothetis and D.L. Sheinberg, "Visual Object Recognition," *Ann. Rev. of Neuroscience,* vol. 19, pp. 577-621, 1996.
[32] A.Y. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Advances in Neural Information Processing Systems 14,* pp. 849-856, MIT Press, 2001.
[33] P. Paatero and U. Tapper, "Positive Matrix Factorization: A Non-Negative Factor Model with Optimal Utilization of Error Estimates of Data Values," *Environmetrics,* vol. 5, no. 2, pp. 111-126, 1994.
[34] S.E. Palmer, "Hierarchical Structure in Perceptual Representation," *Cognitive Psychology,* vol. 9, pp. 441-474, 1977.
[35] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, 2000.
[36] H.S. Seung and D.D. Lee, "The Manifold Ways of Perception," *Science,* vol. 290, no. 12, pp. 2268/2269, 2000.
[37] F. Shahnaza, M.W. Berrya, V. Paucab, and R.J. Plemmonsb, "Document Clustering Using Nonnegative Matrix Factorization," *Information Processing and Management,* vol. 42, no. 2, pp. 373-386, 2006.
[38] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[39] J. Tenenbaum, V. de Silva, and J. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science,* vol. 290, no. 5500, pp. 2319-2323, 2000.
[40] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.
[41] E. Wachsmuth, M.W. Oram, and D.I. Perrett, "Recognition of Objects and Their Component Parts: Responses of Single Units in the Temporal Cortex of the Macaque," *Cerebral Cortex,* vol. 4, pp. 509-522, 1994.
[42] W. Xu, X. Liu, and Y. Gong, "Document Clustering Based on Non-Negative Matrix Factorization," *Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 267-273, Aug. 2003.
[43] L. Zelnik-Manor and P. Perona, "Self-Tuning Spectral Clustering," *Advances in Neural Information Processing Systems 17,* pp. 1601-1608, MIT Press, 2004.
[44] H. Zha, C. Ding, M. Gu, X. He, and H. Simon, "Spectral Relaxation for k-Means Clustering," *Advances in Neural Information Processing Systems 14,* pp. 1057-1064, MIT Press, 2001.
[45] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with Local and Global Consistency," *Advances in Neural Information Processing Systems 16,* MIT Press, 2003.
[46] X. Zhu and J. Lafferty, "Harmonic Mixtures: Combining Mixture Models and Graph-Based Methods for Inductive and Scalable Semi-Supervised Learning," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 1052-1059, 2005.

Deng Cai received the bachelor's and master's degrees from Tsinghua University in 2000 and 2003, respectively, both in automation. He received the PhD degree in computer science from the University of Illinois at Urbana Champaign in 2009. He is an associate professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. His research interests include machine learning, data mining, and information retrieval. He is a member of the IEEE.



Xiaofei He received the BS degree in computer science from Zhejiang University, China, in 2000, and the PhD degree in computer science from the University of Chicago in 2005. He is a professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a research scientist at Yahoo! Research Labs, Burbank, California. His research interests include machine learning, information retrieval, and computer vision. He is a senior member of the IEEE.

Jiawei Han is a professor of computer science at the University of Illinois. He has chaired or served on more than 100 program committees of the major international conferences in the fields of data mining and database systems, and also served or is serving on the editorial boards for *Data Mining and Knowledge Discovery*, the *IEEE Transactions on Knowledge and Data Engineering*, the *Journal of Computer Science and Technology*, and the *Journal of Intelligent Information Systems.* He is the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data (TKDD)*. He has received IBM Faculty Awards, HP Innovation Awards, the ACM SIGKDD Innovation Award (2004), the IEEE Computer Society Technical Achievement Award (2005), and the IEEE W. Wallace McDowell Award (2009). He is currently the director of the Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of US Army Research Lab. His book *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been used worldwide as a textbook. He is a fellow of the ACM and IEEE.

Thomas S. Huang received the ScD degree from the Massachusetts Institute of Technology in electrical engineering, and was on the faculty of MIT and Purdue University. He joined the University of Illinois at Urbana Champaign in 1980 and is currently the William L. Everitt Distinguished Professor of Electrical and Computer Engineering, a research professor in the Coordinated Science Laboratory, a professor in the Center for Advanced Study, and cochair of the Human Computer Intelligent Interaction major research theme of the Beckman Institute for Advanced Science and Technology. He is a member of the National Academy of Engineering and has received numerous honors and awards, including the IEEE Jack S. Kilby Signal Processing Medal (with Ar. Netravali) and the King-Sun Fu Prize of the International Association of Pattern Recognition. He has published 21 books and more than 600 technical papers in network theory, digital holography, image and video compression, multimodal human computer interfaces, and multimedia databases. He is a fellow of the IEEE.