Locally Consistent Concept Factorization for Document Clustering

Deng Cai, Xiaofei He, *Member, IEEE* and Jiawei Han, *Fellow, IEEE*

Abstract—Previous studies have demonstrated that document clustering performance can be improved significantly in lower dimensional linear subspaces. Recently, matrix factorization based techniques, such as Non-negative Matrix Factorization (NMF) and Concept Factorization (CF), have yielded impressive results. However, both of them effectively see only the global Euclidean geometry, whereas the local manifold geometry is not fully considered. In this paper, we propose a new approach to extract the document concepts which are consistent with the manifold geometry such that each concept corresponds to a connected component. Central to our approach is a graph model which captures the local geometry of the document submanifold. Thus we call it Locally Consistent Concept Factoriaztion (LCCF). By using the graph Laplacian to smooth the document-to-concept mapping, LCCF can extract concepts with respect to the intrinsic manifold structure and thus documents associated with the same concept can be well clustered. The experimental results on TDT2 and Reuters-21578 have shown that the proposed approach provides a better representation and achieves better clustering results in terms of accuracy and mutual information.

Index Terms—Non-negative Matrix Factorization, Concept Factorization, Graph Laplacian, Manifold Regularization, Clustering.

1 Introduction

In the last decade, matrix factorization based approaches have attracted considerable attention in document clustering [25], [24]. When using matrix factorization based methods, a text document is usually represented as a point in a high dimensional linear space, each dimension corresponding to a term. Central to all of the goals of cluster analysis is the notion of the degree of similarity (or dissimilarity) between the individual objects being clustered. Recent studies have shown that similarity can be measured more accurately in lower dimensional spaces, and thus the clustering performance can be enhanced [5], [18]. In particular, Non-negative Matrix Factorization (NMF) [25] and Concept Factorization (CF) [24] have been applied to document clustering with impressive results.

In general, the NMF problem is the following: given a nonnegative data matrix $\mathbf{X}$, find reduced rank nonnegative matrices $\mathbf{U}$ and $\mathbf{V}$ so that $\mathbf{UV}^T$ provides a *good* approximation to $\mathbf{X}$. The column vectors of $\mathbf{U}$ can be thought of as basis vectors and $\mathbf{V}$ contains the coordinates. Previous studies have shown there is psychological and physiological evidence for parts-based representation in human brain [16]. The non-negative constraints in NMF lead to a parts-based representation because it allows only additive, not subtractive, combinations. The major limitation of NMF is that it is unclear how to effectively perform NMF in the transformed data space, e.g. reproducing kernel Hilbert space (RKHS).

To address the limitations of NMF while inheriting all its strengths, Xu and Gong proposed Concept Factorization (CF) for data clustering [24]. CF models each cluster as a linear combination of the data points, and each data point as a linear combination of the cluster centers. The data clustering is then accomplished by computing the two sets of linear coefficients, which is carried out by finding the non-negative solution that minimizes the reconstruction error of the data points. The major advantage of CF over NMF is that it can be performed on any data representations, either in the original space or RKHS. Besides NMF and CF, another popular matrix factorization method is Latent Semantic Indexing (LSI) [9].

All the above mentioned methods essentially aim to find new basis vectors based on which the data points are represented. The similarity between data points are measured based on the new representations. However, many previous studies have shown human generated text data is probably sampled from a submanifold of the ambient Euclidean space [1], [19], [23]. In fact, the human generated text documents cannot possibly "fill up" the high dimensional Euclidean space uniformly. Therefore, the intrinsic manifold structure needs to be considered while learning new data representations.

The goal of this paper is to extract the underlying concepts which are consistent with the low dimensional

- *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, China 310058. 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.*

Manuscript received 19 May. 2009; revised 03 Dec. 2009; accepted 14 Jan. 2010

manifold structure with the hope that this will facilitate further processing such as clustering. Central to our approach is a graph model which captures the local geometry of the document submanifold. Thus we call it Locally Consistent Concept Factoriaztion (LCCF). The graph Laplacian, analogous to the Laplace-Beltrami operator on manifolds, can be used to smooth the document-to-concept mapping. Thus, the obtained concepts can well capture the intrinsic geometrical structure and the documents associated with similar concepts can be well clustered. The Euclidean and manifold geometry is unified through a regularization framework where a regularization parameter controls their balance. Although the new approach is no longer optimal in the sense of reconstruction error in Euclidean space, it has a better interpretation from manifold perspective. Moreover, like CF, our method also can be performed in RKHS which gives rise to nonlinear mappings.

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

1) The standard CF fits the data in Euclidean space, LCCF exploits the intrinsic geometry of the data and incorporates it as an additional regularization term. Hence, LCCF is particularly applicable when the data is sampled from a submanifold which is embedded in high dimensional ambient space.
2) The proposed framework is a general one that can leverage the power of both CF 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 CF).
3) Our algorithm uses a nearest neighbor graph to model the manifold structure. Since the weight matrix of the graph is highly sparse, the multiplicative updating rules for LCCF are still very efficient.

The rest of the paper is organized as follows. Section 2 describes NMF and CF. Our Locally Consistent Concept Factorization approach is introduced in Section 3. Section 4 introduces some related work. A variety of experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2 NMF AND CF

Non-negative Matrix Factorization (NMF) [13] is a matrix factorization algorithm that focuses on the analysis of data matrices whose elements are nonnegative. Given a nonnegative data matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, each column of $\mathbf{X}$ is a sample vector. NMF aims to find two non-negative matrices $\mathbf{U} = [u_{ik}] \in \mathbb{R}^{M \times K}$ and $\mathbf{V} = [v_{jk}] \in \mathbb{R}^{N \times K}$ which minimize the following objective function:

$$O = \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|^2 \tag{1}$$

where $\|\cdot\|$ denotes the matrix *Frobenius norm*.

Although the objective function O in Eq. (1) is convex in $\mathbf{U}$ only or $\mathbf{V}$ only, it is not convex in both variables together. Therefore, it is unrealistic to expect an algorithm to find the global minimum of O. Lee & Seung [14] presented an iterative update algorithm as follows:

$$u_{ik}^{t+1} = u_{ik}^t \frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}, \qquad v_{jk}^{t+1} = v_{jk}^t \frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk}} \tag{2}$$

It is proved that the above update steps will find a local mimimum of the objective function O in Eq. (1) [14].

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, $\mathbf{X} \approx \mathbf{U}\mathbf{V}^T$. We can view this approximation column by column as

$$\mathbf{x}_j \approx \sum_{k=1}^{K} \mathbf{u}_k v_{jk} \tag{3}$$

where $\mathbf{u}_k$ is the k-th 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 j-th row of $\mathbf{V}$, $\mathbf{z}_j = [v_{j1}, \cdots, v_{jK}]^T$. $\mathbf{z}_j$ can be regarded as the new representation of each data point in the new basis $\mathbf{U}$. Since relatively fewer basis vectors are used to represent many data vectors, good approximation can only be achieved if the basis vectors discover structure that is latent in the data [14]. The non-negative constraints on $\mathbf{U}$ and $\mathbf{V}$ require the combination coefficients among different basis can only be positive. This is the most significant difference between NMF and 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 [13]. The advantages of this parts-based representation have been observed in many real world problems such as face analysis [6], document clustering [25] and DNA gene expression analysis [4]. Please see [15], [26] for other NMF extensions.

NMF can only be performed in the original feature space of the data points. In the case that the data are highly non-linear distributed, it is desirable that we can kernelize NMF and apply the powerful idea of the kernel method. To achieve this goal, Xu and Gong [24] proposed an extension of NMF which is called Concept Factorization (CF). In CF, each basis $\mathbf{u}_k$ is required to be a non-negative linear combination of the sample vectors $\mathbf{x}_j$

$$\mathbf{u}_k = \sum_{j=1}^{N} \mathbf{x}_j w_{jk} \tag{4}$$

where $w_{jk} \geq 0$. Let $\mathbf{W} = [w_{jk}] \in \mathbb{R}^{N \times K}$, CF essentially tries to find the following approximation:

$$\mathbf{X} \approx \mathbf{X}\mathbf{W}\mathbf{V}^T \tag{5}$$

through minimization of

$$O = \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|^2. \tag{6}$$

The multiplicative updating rules minimizing the above objective function are given as [24]

$$w_{jk}^{t+1} = w_{jk}^t \frac{(\mathbf{K}\mathbf{V})_{jk}}{(\mathbf{K}\mathbf{W}\mathbf{V}^T\mathbf{V})_{jk}} \quad v_{jk}^{t+1} \leftarrow v_{jk}^t \frac{(\mathbf{K}\mathbf{W})_{jk}}{(\mathbf{V}\mathbf{W}^T\mathbf{K}\mathbf{W})_{jk}} \tag{7}$$

where $\mathbf{K} = \mathbf{X}^T\mathbf{X}$. These multiplicative updating rules only involve the inner product of $\mathbf{x}$ and thus CF can be easily kernelized. Please see [24] for details. With extensive experimental results, Xu and Gong [24] show the superiority of CF over NMF for document clustering.

3 Locally Consistent Concept Factorization

Both NMF and CF perform the factorization in the Euclidean space. They fail to discover the local geometrical and discriminating structure of the data space, which is essential to the clustering problem. In this Section, we introduce our *Locally Consistent Concept Factorization* (LCCF) algorithm which avoids this limitation by incorporating a geometrically based regularizer.

3.1 The Objective Function

Recall that CF tries to find a basis that is optimized for the linear approximation of the data. The j-th row of matrix $\mathbf{V}$, $\mathbf{z}_j^T = [v_{j1}, \cdots, v_{jk}]$, can be regarded as the new representation of each data point in the new basis. One might hope that knowledge of the geometric structure of the data can be exploited for better discovery of this basis. A natural assumption here could be that if two data points $\mathbf{x}_j, \mathbf{x}_s$ are *close* in the *intrinsic* geometry of the data distribution, then $\mathbf{z}_j$ and $\mathbf{z}_s$, the representations of this two points in the new basis, are also close to each other. This assumption is usually referred to as *local consistency assumption* [28], which plays an essential role in developing various kinds of algorithms including dimensionality reduction algorithms [1] and semi-supervised learning algorithms [2].

Recent studies on spectral graph theory [7] 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 document in the corpus. Define the edge weight matrix $\mathbf{S}$ as follows:

$$\mathbf{S}_{js} = \begin{cases} \frac{\mathbf{x}_j^T\mathbf{x}_s}{\|\mathbf{x}_j\|\|\mathbf{x}_s\|}, & \text{if } \mathbf{x}_j \in N_p(\mathbf{x}_s) \text{ or } \mathbf{x}_s \in N_p(\mathbf{x}_j) \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

where $N_p(\mathbf{x}_s)$ denotes the set of p nearest neighbors of $\mathbf{x}_s$. Then, the following term can be used to measure the smoothness of the low dimensional representations on this p-nearest neighbor graph.

$$\begin{aligned} \mathcal{R} &= \frac{1}{2}\sum_{j,s=1}^{N} \|\mathbf{z}_j - \mathbf{z}_s\|^2 \mathbf{S}_{js} \\ &= \sum_{j=1}^{N} \mathbf{z}_j^T\mathbf{z}_j\mathbf{D}_{jj} - \sum_{j,s=1}^{N} \mathbf{z}_j^T\mathbf{z}_s\mathbf{S}_{js} \\ &= \mathrm{Tr}(\mathbf{V}^T\mathbf{D}\mathbf{V}) - \mathrm{Tr}(\mathbf{V}^T\mathbf{S}\mathbf{V}) = \mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}), \end{aligned} \tag{9}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix and $\mathbf{D}$ is a diagonal matrix whose entries are column (or row, since $\mathbf{S}$ is symmetric) sums of $\mathbf{S}$, $\mathbf{D}_{jj} = \sum_s \mathbf{S}_{js}$. $\mathbf{L} = \mathbf{D} - \mathbf{S}$, which is called graph Laplacian [7]. By minimizing $\mathcal{R}$, we essentially try to formalize our intuition that if two data points $\mathbf{x}_j$ and $\mathbf{x}_s$ are close (*i.e.* $\mathbf{S}_{js}$ is big), $\mathbf{z}_j$ and $\mathbf{z}_s$ are also close to each other.

Our LCCF incorporates the $\mathcal{R}$ term and minimizes the objective function as follows:

$$\mathcal{O} = \|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|^2 + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \tag{10}$$

with the constraint that $\mathbf{W}$ and $\mathbf{V}$ are non-negative matrices. The $\lambda \geq 0$ is the regularization parameter.

3.2 A Multiplicative Algorithm

The objective function $\mathcal{O}$ of LCCF in Eq. (10) is not convex in both $\mathbf{W}$ and $\mathbf{V}$ together. Therefore, it is unrealistic to expect an algorithm to find the global minimum of $\mathcal{O}$. In the following, we introduce an iterative algorithm which can achieve a local minimum.

Define $\mathbf{K} = \mathbf{X}^T\mathbf{X}$ and use the properties $\|\mathbf{A}\|^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{A})$, $\mathrm{Tr}(\mathbf{A}\mathbf{B}) = \mathrm{Tr}(\mathbf{B}\mathbf{A})$ and $\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$, we can rewrite the objective function $\mathcal{O}$:

$$\begin{aligned} \mathcal{O} &= \mathrm{Tr}\left((\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T)\right) + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}\left((\mathbf{I} - \mathbf{W}\mathbf{V}^T)^T\mathbf{K}(\mathbf{I} - \mathbf{W}\mathbf{V}^T)\right) + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &= \mathrm{Tr}(\mathbf{K}) - 2\,\mathrm{Tr}(\mathbf{V}\mathbf{W}^T\mathbf{K}) + \mathrm{Tr}(\mathbf{V}\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V}^T) \\ &\quad + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \end{aligned} \tag{11}$$

Let ψ_{jk} and ϕ_{jk} be the Lagrange multiplier for constraints $w_{jk} \geq 0$ and $v_{jk} \geq 0$, respectively. We define matrix $\Psi = [\psi_{jk}]$ and $\Phi = [\phi_{jk}]$, then the Lagrange $\mathcal{L}$ is

$$\begin{aligned} \mathcal{L} = &\mathrm{Tr}(\mathbf{K}) - 2\,\mathrm{Tr}(\mathbf{V}\mathbf{W}^T\mathbf{K}) + \mathrm{Tr}(\mathbf{V}\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V}^T) \\ &+ \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) + \mathrm{Tr}(\Psi\mathbf{W}^T) + \mathrm{Tr}(\Phi\mathbf{V}^T) \end{aligned} \tag{12}$$

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

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}} = -2\mathbf{K}\mathbf{V} + 2\mathbf{K}\mathbf{W}\mathbf{V}^T\mathbf{V} + \Psi \tag{13}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = -2\mathbf{K}\mathbf{W} + 2\mathbf{V}\mathbf{W}^T\mathbf{K}\mathbf{W} + 2\lambda\mathbf{L}\mathbf{V} + \Phi \tag{14}$$

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

$$-(\mathbf{K}\mathbf{V})_{jk}w_{jk} + (\mathbf{K}\mathbf{W}\mathbf{V}^T\mathbf{V})_{jk}w_{jk} = 0 \tag{15}$$

$$-(\mathbf{KW})_{jk}v_{jk} + (\mathbf{VW}^T\mathbf{KW})_{jk}v_{jk} + \lambda(\mathbf{LV})_{jk}v_{jk} = 0 \quad (16)$$

The above equations lead to the following updating rules:

$$w_{jk} \leftarrow w_{jk}\frac{(\mathbf{KV})_{jk}}{(\mathbf{KWV}^T\mathbf{V})_{jk}} \quad (17)$$

$$v_{jk} \leftarrow v_{jk}\frac{(\mathbf{KW}+\lambda\mathbf{SV})_{jk}}{(\mathbf{VW}^T\mathbf{KW}+\lambda\mathbf{DV})_{jk}} \quad (18)$$

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

Theorem 1: The objective function $\mathcal{O}$ in Eq. (10) is non-increasing under the updating rules in Eq. (17) and (18). The objective function is invariant under these updates if and only if $\mathbf{W}$ and $\mathbf{V}$ are at a stationary point.

Theorem 1 grantees that the updating rules of $\mathbf{W}$ and $\mathbf{V}$ in Eq. (17) and (18) converge and the final solution will be a local optimum. Please see the Appendix for a detailed proof.

For the objective function of CF, It is easy to check that if $\mathbf{W}$ and $\mathbf{V}$ are the solution, then, $\mathbf{WD}$, $\mathbf{VD}^{-1}$ will also form a solution for any positive diagonal matrix $\mathbf{D}$. To eliminate this uncertainty, in practice people will further require that $\mathbf{w}^T\mathbf{Kw} = 1$, where $\mathbf{w}$ is the column vector of $\mathbf{W}$ [24]. The matrix $\mathbf{V}$ will be adjusted accordingly so that $\mathbf{WV}^T$ does not change. This can be achieved by:

$$\mathbf{V} \leftarrow \mathbf{V}[\mathrm{diag}(\mathbf{W}^T\mathbf{KW})]^{1/2},$$
$$\mathbf{W} \leftarrow \mathbf{W}[\mathrm{diag}(\mathbf{W}^T\mathbf{KW})]^{-1/2}$$

Our LCCF also adopts this strategy.

3.3 Connection with Gradient Descent Method

Another general algorithm for minimizing the objective function of LCCF in Eq. (10) could be gradient descent [12]. For our problem, gradient descent leads to the following additive updating rules:

$$w_{jk} \leftarrow w_{jk} + \eta_{jk}\frac{\partial\mathcal{O}}{\partial w_{jk}}, \qquad v_{jk} \leftarrow v_{jk} + \delta_{jk}\frac{\partial\mathcal{O}}{\partial v_{jk}}$$

The η_{jk} and δ_{jk} are usually referred as step size parameters. As long as η_{jk} and δ_{jk} are sufficiently small, the above updates should reduce $\mathcal{O}$.

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

$$w_{jk} + \eta_{jk}\frac{\partial\mathcal{O}}{\partial w_{jk}} = w_{jk} - \frac{w_{jk}}{2(\mathbf{KWV}^T\mathbf{V})_{jk}}\frac{\partial\mathcal{O}}{\partial w_{jk}}$$
$$= w_{jk} - \frac{w_{jk}}{2(\mathbf{KWV}^T\mathbf{V})_{jk}}\Big(-2(\mathbf{KV})_{jk} + 2(\mathbf{KWV}^T\mathbf{V})_{jk}\Big)$$
$$= w_{jk}\frac{(\mathbf{KV})_{jk}}{(\mathbf{KWV}^T\mathbf{V})_{jk}}$$

TABLE 1
Abbreviations for reporting operation counts

Abbreviation	Description
fladd	a floating-point addition
flmlt	a floating-point multiplication
fldiv	a floating-point division
flam	a compound operation consisting of one addition and one multiplication

Similarly, let $\delta_{jk} = -v_{jk}/2(\mathbf{VW}^T\mathbf{KW}+\lambda\mathbf{DV})_{jk}$, we have

$$v_{jk} + \delta_{jk}\frac{\partial\mathcal{O}}{\partial v_{jk}} = v_{jk} - \frac{v_{jk}}{2(\mathbf{VW}^T\mathbf{KW}+\lambda\mathbf{DV})_{jk}}\frac{\partial\mathcal{O}}{\partial v_{jk}}$$
$$= v_{jk} - \frac{v_{jk}}{2(\mathbf{VW}^T\mathbf{KW}+\lambda\mathbf{DV})_{jk}}\Big(-2(\mathbf{KW})_{jk}$$
$$+ 2(\mathbf{VW}^T\mathbf{KW})_{jk} + 2\lambda(\mathbf{DV})_{jk} - 2\lambda(\mathbf{SV})_{jk}\Big)$$
$$= v_{jk}\frac{(\mathbf{KW}+\lambda\mathbf{SV})_{jk}}{(\mathbf{VW}^T\mathbf{KW}+\lambda\mathbf{DV})_{jk}}$$

Now it is clear that the multiplicative updating rules in Eq. (17) and Eq. (18) are special cases of gradient descent with automatically step size parameter selection. The advantage of multiplicative updating rules is the guarantee the non-negativity of $\mathbf{W}$ and $\mathbf{V}$. Theorem 1 also guarantees that the multiplicative updating rules in Eq. (17) and (18) converge to a local optimum.

3.4 Computational Complexity Analysis

In this subsection, we discuss the extra computational cost of our proposed algorithm comparing to standard CF and NMF.

The common way to express the complexity of one algorithm is using big O notation [8]. However, it is not precise enough to differentiate the complexities of CF and LCCF. Thus, we count the arithmetic operations for each algorithm. Three operation abbreviations used in this paper are summarized in Table 1. Please see [22] for more details about these operation abbreviations.

Based on the updating rules in Eq. (2), it is not hard to count the arithmetic operations of each iteration in NMF and CF. We summarize the results in Table 2. For LCCF, it is important to note that $\mathbf{S}$ is a sparse matrix. If we use a p-nearest neighbor graph, the average nonzero elements on each row of $\mathbf{S}$ is p. Thus, we only need NpK flam to compute $\mathbf{SV}$. We also summarize the arithmetic operations for each iteration of LCCF in Table 2. Comparing to CF, LCCF only needs $N(p+3)K$ more flam in each iteration. Since we have $p \ll N$, this additional cost is dominated by the remaining cost of LCCF. Thus, the overall cost of both CF and LCCF in each iteration are $O(N^2K)$.

Besides the multiplicative updating, both CF and LCCF need to compute the kernel matrix $\mathbf{K}$ which requires $O(N^2M)$ operations. Moreover, LCCF needs $O(N^2M + N^2p)$ to construct the p-nearest neighbor

TABLE 2
Computational operation counts for each iteration in NMF, CF and LCCF

	fladd	flmlt	fldiv	overall
NMF	$2MNK + 2(M+N)K^2$	$2MNK + 2(M+N)K^2 + (M+N)K$	$(M+N)K$	$O(MNK)$
CF	$4N^2K + 4NK^2$	$4N^2K + 4NK^2 + 2NK$	$2NK$	$O(N^2K)$
LCCF	$4N^2K + 4NK^2 + N(p+3)K$	$4N^2K + 4NK^2 + N(p+3)K$	$2NK$	$O(N^2K)$

N: the number of sample points M: the number of features K: the number of factors
p: the number of nearest neighbors, $p \ll N$

graph. Suppose the multiplicative updates stops after t iterations, the overall cost for NMF is

$$O(tMNK). \tag{19}$$

The overall cost for CF is

$$O(tN^2K + N^2M) \tag{20}$$

and the overall cost for LCCF is

$$O(tN^2K + N^2M + N^2p) \tag{21}$$

Since p is usually very small (around 5), LCCF and CF will have the same computational complexity by using the big O notation when dealing with the high dimensional data ($M \gg p$).

3.5 An Algorithm for Negative Data Matrices

The algorithm we introduced in Section 3.2 only works when the $\mathbf{K}$ is nonnegative. In the case that the data matrix has negative values, it is possible that the $\mathbf{K}$ has negative entries. In this Section, we will introduce a general algorithm which can be applied for any case. Our approach follows [24], which is essentially based on the following theorem proposed by Sha *et al.* [20].

Theorem 2: Define the non-negative general quadratic form as

$$f(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} + \mathbf{b}^T\mathbf{v}$$

where $\mathbf{v}$ is an m dimensional non-negative vector, $\mathbf{A}$ is a symmetric positive definite matrix and $\mathbf{b}$ is an arbitrary m dimensional vector. Let $\mathbf{A}^+$ and $\mathbf{A}^-$ denote the nonnegative matrices with elements:

$$\mathbf{A}^+_{ij} = \begin{cases} \mathbf{A}_{ij}, & \text{if } \mathbf{A}_{ij} > 0, \\ 0, & \text{otherwise.} \end{cases} \quad \mathbf{A}^-_{ij} = \begin{cases} |\mathbf{A}_{ij}|, & \text{if } \mathbf{A}_{ij} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

It is easily to see that $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$. Then the solution $\mathbf{v}$ that minimizes $f(\mathbf{v})$ can be obtained through the following iterative update

$$v_i \leftarrow v_i \left[\frac{-b_i + \sqrt{b_i^2 + 4(\mathbf{A}^+\mathbf{v})_i(\mathbf{A}^-\mathbf{v})_i}}{2(\mathbf{A}^+\mathbf{v})_i} \right] \tag{22}$$

From the Eq. (11), we can easily see that the objective function $\mathcal{O}$ of LCCF is a quadratic form of $\mathbf{W}$ (or, $\mathbf{V}$) only and the Theorem 2 can naturally be applied. We only need to identify the corresponding $\mathbf{A}$ and $\mathbf{b}$ in the objective function.

Fixing $\mathbf{V}$, the part $\mathbf{b}$ for the quadratic form $\mathcal{O}(\mathbf{W})$ can be obtained by taking the first order derivative with respect to $\mathbf{W}$ at $\mathbf{W} = 0$:

$$\left.\frac{\partial \mathcal{O}}{\partial w_{jk}}\right|_{\mathbf{W}=0} = -2(\mathbf{KV})_{jk} \tag{23}$$

The part $\mathbf{A}$ for the quadratic form $\mathcal{O}(\mathbf{W})$ can be obtained by taking the second order derivative with respect to $\mathbf{W}$.

$$\frac{\partial^2 \mathcal{O}}{\partial w_{jk}\partial w_{il}} = 2(\mathbf{K})_{ji}(\mathbf{V}^T\mathbf{V})_{lk} \tag{24}$$

Let $\mathbf{K}^+$ and $\mathbf{K}^-$ denote the nonnegative matrices with elements:

$$\mathbf{K}^+_{ij} = \begin{cases} \mathbf{K}_{ij}, & \text{if } \mathbf{K}_{ij} > 0, \\ 0, & \text{otherwise.} \end{cases} \quad \mathbf{K}^-_{ij} = \begin{cases} |\mathbf{K}_{ij}|, & \text{if } \mathbf{K}_{ij} < 0, \\ 0, & \text{otherwise.} \end{cases}$$

We have $\mathbf{K} = \mathbf{K}^+ - \mathbf{K}^-$. Substituting $\mathbf{A}$ and b_i in Eq. (22) using Eq. (24) and (23), respectively, we obtain the multiplicative updating equation for each element w_{jk} of $\mathbf{W}$:

$$w_{jk} \leftarrow w_{jk} \frac{(\mathbf{KV})_{jk} + \sqrt{(\mathbf{KV})^2_{jk} + 4\mathbf{P}^+_{jk}\mathbf{P}^-_{jk}}}{2\mathbf{P}^+_{jk}} \tag{25}$$

where $\mathbf{P}^+ = \mathbf{K}^+\mathbf{W}\mathbf{V}^T\mathbf{V}$ and $\mathbf{P}^- = \mathbf{K}^-\mathbf{W}\mathbf{V}^T\mathbf{V}$.

Similarly, we can get the updating equation for each element v_{jk} in $\mathbf{V}$ by applying the Theorem 2 to the quadratic form $\mathcal{O}(\mathbf{V})$. Fixing $\mathbf{W}$, we get

$$\left.\frac{\partial \mathcal{O}}{\partial v_{jk}}\right|_{\mathbf{V}=0} = -2(\mathbf{KW})_{jk} \tag{26}$$

$$\frac{\partial^2 \mathcal{O}}{\partial v_{jk}\partial v_{il}} = 2\delta_{ij}(\mathbf{W}^T\mathbf{K}\mathbf{W})_{lk} + 2\lambda\delta_{lk}\mathbf{L}_{ji} \tag{27}$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \quad \delta_{lk} = \begin{cases} 1, & \text{if } l = k, \\ 0, & \text{otherwise.} \end{cases}$$

The updating equation for $\mathbf{V}$ is:

$$v_{jk} \leftarrow v_{jk} \frac{(\mathbf{KW})_{jk} + \sqrt{(\mathbf{KW})^2_{jk} + 4\mathbf{Q}^+_{jk}\mathbf{Q}^-_{jk}}}{2\mathbf{Q}^+_{jk}} \tag{28}$$

where $\mathbf{Q}^+ = \mathbf{V}\mathbf{W}^T\mathbf{K}^+\mathbf{W} + \lambda\mathbf{D}\mathbf{V}$ and $\mathbf{Q}^- = \mathbf{V}\mathbf{W}^T\mathbf{K}^-\mathbf{W} + \lambda\mathbf{S}\mathbf{V}$.

3.6 Out-of-Sample Extension

The NMF based methods (including CF and LCCF discussed in this paper) provide the factorization results only for given training points. When a new data point comes, there is no straightforward way to figure out the factorization result of the new data point. This is commonly referred as the out-of-sample extension issue [3]. In this subsection, we discuss how to handle the out-of-sample extension in LCCF.

Given a $M \times N$ data matrix $\mathbf{X}$, recall that LCCF tries to find two non-negative $N \times K$ matrices $\mathbf{W}$ and $\mathbf{V}$ through minimization of

$$\|\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T\|^2 + \lambda \operatorname{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}),$$

where W contains the coefficients for linear combinations of the data points to form clusters, and $\mathbf{V}$ contains the coefficients modeling each data point as a linear combinations of the cluster centers. For each data point $\mathbf{x}_i$ (i-th column vector of $\mathbf{X}$), the factorization result contains two parts: the i-th row of $\mathbf{W}$ and the i-th row of $\mathbf{V}$.

Given a new data point $\mathbf{x}'$, let $\widehat{\mathbf{X}} = [\mathbf{X}, \mathbf{x}'] \in \mathbb{R}^{M\times(N+1)}$, our goal is to find the factorization result of this new point. A naive way is applying the multiplicative updating algorithm discussed in Section 3.2 on $\widehat{\mathbf{X}}$ to find the new factorization matrices $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$. The factorization result of $\mathbf{x}'$ will be the last row of matrices $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$. However, this naive approach is too time consuming. Since we only have one new data point, we can assume that the matrices containing the first N rows of $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$ are the same as the matrices $\mathbf{W}$ and $\mathbf{V}$. And we only need to use the multiplicative updating to find the last row of $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$. Define

$$\widehat{\mathbf{S}} = \begin{bmatrix} \mathbf{S} & \tilde{\mathbf{s}} \\ \tilde{\mathbf{s}}^T & 0 \end{bmatrix} \in \mathbb{R}^{(N+1)\times(N+1)}$$

where $\mathbf{S} \in \mathbb{R}^{N\times N}$ is the weight matrix of the p-nearest neighbor graph constructed on the previous N samples. $\tilde{\mathbf{s}} \in \mathbb{R}^N$ is defined as follows:

$$\tilde{\mathbf{s}}_i = \begin{cases} \frac{\mathbf{x}_i^T\mathbf{x}'}{\|\mathbf{x}_i\|\|\mathbf{x}'\|}, & \text{if } \mathbf{x}_i \in N_p(\mathbf{x}') \\ 0, & \text{otherwise.} \end{cases}$$

$\widehat{\mathbf{S}}$ is the weight matrix of the p-nearest neighbor graph constructed on all the $N+1$ samples. Let $\mathbf{k}'$ be the last row vector of the kernel matrix $\widehat{\mathbf{K}}$ on the data matrix $\widehat{\mathbf{X}}$ and let $\mathbf{s}'$, $\mathbf{w}'$ and $\mathbf{v}'$ be the last row vectors of matrices $\widehat{\mathbf{S}}$, $\widehat{\mathbf{W}}$ and $\widehat{\mathbf{V}}$, respectively, it is not hard to get the multiplicative updating rules as follows:

$$\mathbf{w}'_i \leftarrow \mathbf{w}'_i \frac{\left(\mathbf{k}'\widehat{\mathbf{V}}\right)_i}{\left(\mathbf{k}'\widehat{\mathbf{W}}\widehat{\mathbf{V}}^T\widehat{\mathbf{V}}\right)_i} \tag{29}$$

$$\mathbf{v}'_i \leftarrow \mathbf{v}'_i \frac{\left(\mathbf{k}'\widehat{\mathbf{W}} + \lambda\mathbf{s}'\widehat{\mathbf{V}}\right)_i}{\left(\mathbf{v}'\widehat{\mathbf{W}}^T\widehat{\mathbf{K}}\widehat{\mathbf{W}} + \lambda\mathbf{d}'\widehat{\mathbf{V}}\right)_i} \tag{30}$$

The construction of the weight matrix $\widehat{\mathbf{S}}$ of the p-nearest neighbor graph needs $O(NM)$ operations and calculating the new kernel matrix also needs $O(NM)$ operations. The multiplicative updating in Eq. (29) and (30) needs $O(NK^2)$ operations. If the updating procedure stops after t iterations, the overall cost to find the factorization result of a new data point is

$$O(tNK^2 + NM).$$

TABLE 3
Statistics of TDT2 and Reuters corpora.

	TDT2	Reuters
No. docs. used	10021	8213
No. clusters used	56	41
Max. cluster size	1844	3713
Min. cluster size	10	10
Med. cluster size	58	38
Avg. cluster size	179	200

4 Experimental Results

Previous studies show that both CF and NMF are very powerful on document clustering [25][24]. They can achieve similar or better performance than most of the state-of-the-art clustering algorithms, including the popular spectral clustering methods [25]. In this section, we also evaluate our LCCF algorithm on document clustering problem.

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

4.1 Data Corpora

We conduct the performance evaluations using the TDT2 and the Reuters document corpora.

The TDT2[1] corpus consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this experiment, those documents appearing in two or more categories are removed, and the categories with more than 10 documents are kept, thus leaving us with 10,021 documents in total.

The Reuters[2] corpus contains 21578 documents which are grouped into 135 clusters. Compared with TDT2 corpus, the Reuters corpus is more difficult for clustering. In TDT2, the content of each cluster is narrowly defined, whereas in Reuters, documents in each cluster have a broader variety of content. Moreover, the Reuters corpus is much more unbalanced, with some large clusters more than 300 times larger than some small ones. In our test, we discard documents with multiple category labels, and only select the categories with more than 10 documents. This lefts us with 8,213 documents in total. Table 3 provides the statistics of the two document corpora.

1. Nist Topic Detection and Tracking corpus at http://www.nist.gov/speech/tests/tdt/tdt98/index.htm
2. Reuters-21578 corpus is at http://www.daviddlewis.com/resources/testcollections/reuters21578/

In both of the two corpora, the stop words are removed and each document is represented as a *tf-idf* vector. These two processed data sets can be downloaded at:
http://www.zjucadcg.cn/dengcai/Data/TextData.html

4.2 Evaluation Metric

The clustering result is evaluated by comparing the obtained label of each document using clustering algorithms with that provided by the document corpus. We use two metrics to measure the performance. The first metric is the accuracy (AC) [5], [25]. Given a document $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$\text{AC} = \frac{\sum_{i=1}^{N} \delta(s_i, map(r_i))}{N}$$

where N is the total number of documents and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [17].

The second metric is the normalized mutual information metric (NMI) [5], [25]. Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $\text{MI}(C, C')$ is defined as follows:

$$\text{MI}(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a document arbitrarily selected from the corpus belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected document belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information NMI as follows:

$$\text{NMI}(C, C') = \frac{\text{MI}(C, C')}{\max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\text{NMI}(C, C')$ ranges from 0 to 1. NMI= 1 if the two sets of clusters are identical, and NMI= 0 if the two sets are independent.

4.3 Performance Evaluations and Comparisons

To demonstrate how the document clustering performance can be improved by our method, we compared LCCF with other five popular document clustering algorithms as follows:

- Traditional kmeans clustering method (KM in short).
- Spectral clustering algorithm based on average association criterion (AA in short)[27]. As shown in [27], the spectral clustering algorithm based on kmeans criterion is equivalent to AA.
- Spectral clustering algorithm based on normalized cut criterion (NC in short)[21]. As shown in [25], NC can be considered as a variation of the AA spectral clustering algorithm that applies certain weighting scheme to the given data set.
- Nonnegative Matrix Factorization based clustering (NMF in short) [25].
- Concept Factorization based clustering (CF in short) [24].

In addition to the original form of all the above algorithms, we also implement the normalized-cut weighted form (NCW) suggested by [25][24]. When the data set is unbalanced, the NCW weighting can automatically reweight the samples which leads to better clustering performance [25][24]. The weighted form of LCCF is derived in appendix B. It is important to note that NC is exactly the NC-weighted form of AA [25]. Essentially, we have compared five approaches (KM, AA, NMF, CF, LCCF) and their NC-weighted versions in the experiment. For the algorithms to which the kernel trick can be applied (*i.e.* KM, AA, CF, LCCF and their NC-weighted versions), we also implement their kernelized versions with degree 2 polynomial kernel .

There are two parameters in our LCCF approach: the number of nearest neighbors p and the regularization parameter λ. Throughout our experiments, we empirically set the number of nearest neighbors p to 5, the value of the regularization parameter λ to 100.

Table 4, 5, 6 and 7 show the evaluation results using the TDT2 and the Reuters corpus, respectively. The evaluations were conducted with the cluster numbers ranging from two to ten. For each given cluster number k, 50 test runs were conducted on different randomly chosen clusters and the average performance is reported in the tables.

These experiments reveal a number of interesting points:

- The use of NC weighting consistently improves the clustering performance (KM-NCW vs. KM, NC vs. AA, NMF-NCW vs. NMF, CF-NCW vs. CF and LCCF-NCW vs. LCCF). This result is consistent with the finding in [25][24].
- The similar performance between KM and AA (or, KM-NCW and NC) confirms the theoretical analysis that KM and AA (or, KM-NCW and NC) are equivalent with respect of their objective functions [27][11].
- For both two concept factorization based methods (CF and LCCF), the use of polynomial kernel has positive effect (except the NMI metric on Reuters for CF). The use of polynomial kernel has negative effect for KM and AA. However, their NC weighted version (KM-NCW and NC) can gain benefit from applying the kernel trick.
- Regardless of the data corpora, LCCF-NCW always has the best performance. This shows that by con-

TABLE 4
Clustering performance on TDT2 (Accuracy)

k	2	3	4	5	6	7	8	9	10	Avg.
KM	0.938	0.882	0.834	0.778	0.762	0.708	0.653	0.651	0.613	0.758
KM-NCW	**0.985**	0.936	0.950	0.884	0.902	0.878	0.834	0.814	0.775	0.884
AA	0.929	0.846	0.803	0.753	0.733	0.705	0.655	0.641	0.609	0.742
NC	0.983	0.960	0.950	0.883	0.910	0.880	0.822	0.811	0.786	0.887
NMF	0.866	0.804	0.755	0.705	0.699	0.681	0.608	0.611	0.578	0.701
NMF-NCW	0.984	0.959	0.949	0.885	0.910	0.878	0.844	0.824	0.809	0.894
CF	0.863	0.817	0.779	0.721	0.708	0.695	0.658	0.646	0.594	0.720
CF-NCW	**0.985**	0.965	0.950	0.894	0.932	0.896	0.864	0.846	0.820	0.906
LCCF	0.952	0.896	0.877	0.821	0.807	0.794	0.729	0.738	0.689	0.811
LCCF-NCW	**0.985**	**0.979**	**0.985**	**0.973**	**0.976**	**0.981**	**0.981**	**0.971**	**0.957**	**0.976**
following are kernelized version of each algorithm with kernel function $\langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$										
KM	0.876	0.786	0.784	0.738	0.706	0.698	0.659	0.660	0.621	0.725
KM-NCW	0.970	0.958	0.951	0.940	0.947	0.925	0.926	0.910	0.889	0.935
AA	0.847	0.801	0.781	0.753	0.718	0.699	0.682	0.654	0.619	0.728
NC	**0.993**	0.933	0.948	0.934	0.942	0.935	0.941	0.931	0.925	0.943
CF	0.862	0.824	0.817	0.785	0.791	0.791	0.742	0.753	0.726	0.788
CF-NCW	**0.993**	0.945	0.929	0.922	0.941	0.940	0.926	0.928	0.932	0.940
LCCF	0.971	0.917	0.888	0.843	0.850	0.830	0.767	0.777	0.748	0.843
LCCF-NCW	**0.993**	**0.983**	**0.991**	**0.988**	**0.987**	**0.980**	**0.976**	**0.971**	**0.961**	**0.981**

TABLE 5
Clustering performance on TDT2 (Normalized Mutual Information)

k	2	3	4	5	6	7	8	9	10	Avg.
KM	0.807	0.771	0.739	0.691	0.716	0.668	0.629	0.648	0.622	0.699
KM-NCW	0.937	0.839	0.885	0.804	0.841	0.814	0.775	0.774	0.750	0.824
AA	0.745	0.696	0.690	0.650	0.673	0.652	0.611	0.614	0.594	0.658
NC	0.918	0.891	0.880	0.798	0.843	0.819	0.766	0.767	0.756	0.826
NMF	0.687	0.678	0.667	0.625	0.661	0.648	0.593	0.616	0.596	0.641
NMF-NCW	0.947	0.903	0.891	0.807	0.863	0.832	0.797	0.792	0.781	0.846
CF	0.678	0.695	0.678	0.636	0.670	0.663	0.626	0.639	0.605	0.654
CF-NCW	0.947	0.908	0.893	0.817	0.888	0.852	0.814	0.805	0.795	0.858
LCCF	0.850	0.791	0.783	0.726	0.743	0.733	0.684	0.696	0.669	0.742
LCCF-NCW	**0.954**	**0.949**	**0.949**	**0.912**	**0.941**	**0.947**	**0.946**	**0.928**	**0.913**	**0.938**
following are kernelized version of each algorithm with kernel function $\langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$										
KM	0.556	0.542	0.595	0.555	0.571	0.567	0.539	0.564	0.546	0.559
KM-NCW	0.850	0.868	0.863	0.856	0.859	0.847	0.851	0.826	0.816	0.849
AA	0.417	0.487	0.502	0.499	0.523	0.521	0.492	0.517	0.502	0.495
NC	0.906	0.834	0.863	0.855	0.880	0.872	0.870	0.858	0.847	0.865
CF	0.576	0.634	0.664	0.659	0.691	0.703	0.660	0.695	0.670	0.661
CF-NCW	0.908	0.834	0.816	0.822	0.879	0.887	0.846	0.857	0.852	0.856
LCCF	0.924	0.828	0.793	0.754	0.793	0.771	0.721	0.744	0.720	0.783
LCCF-NCW	**0.959**	**0.940**	**0.949**	**0.941**	**0.954**	**0.942**	**0.937**	**0.934**	**0.912**	**0.941**

sidering the intrinsic geometrical structure of the data, LCCF can learn a better compact representation in the sense of semantic structure.

- The improvement of LCCF over other methods is more significant on the TDT2 corpus than the Reuters corpus. One possible reason is that the document clusters in TDT2 are generally more compact and focused than the clusters in Reuters. Thus, the nearest neighbor graph constructed over TDT2 can better capture the geometrical structure of the document space.

4.4 Parameters Selection

Our LCCF algorithm has two essential parameters: the number of nearest neighbors p and the regularization parameter λ. Figure 1 and Figure 2 show how the performance of LCCF varies with the parameters λ and p, respectively.

As we can see, the performance of LCCF is very stable with respect to the parameter λ. LCCF achieves consistent good performance with the λ varying from 50 to 10000 on TDT2 data set and from 50 to 1000 on Reuters data set.

Figure 2 is more interesting. As we described, LCCF uses a p-nearest neighbor graph to capture the local geometric structure of the document space. The success of LCCF relies on how the assumption that a document shares the same label with its p-nearest neighbor holds. Obviously this assumption is more likely to fail when p increases. The performance curve drops much more dramatically in Reuters data set, which is consistent with

TABLE 6
Clustering performance on Reuters-21578 (Accuracy)

k	2	3	4	5	6	7	8	9	10	Avg.
KM	0.825	0.692	0.652	0.588	0.589	0.539	0.469	0.447	0.480	0.587
KM-NCW	**0.899**	0.853	0.802	0.707	0.693	0.618	0.572	0.568	0.596	0.701
AA	0.812	0.716	0.648	0.596	0.573	0.536	0.467	0.450	0.478	0.586
NC	0.898	0.832	0.774	0.724	0.698	0.658	0.575	0.585	0.613	0.706
NMF	0.828	0.712	0.683	0.580	0.582	0.535	0.448	0.457	0.490	0.591
NMF-NCW	0.894	0.846	0.786	0.720	0.701	0.667	0.581	0.575	0.625	0.711
CF	0.831	0.715	0.694	0.592	0.614	0.549	0.451	0.456	0.508	0.601
CF-NCW	0.893	0.852	0.795	0.720	0.708	0.673	0.594	0.587	0.617	0.715
LCCF	0.868	0.756	0.752	0.698	0.677	0.606	0.598	0.568	0.590	0.679
LCCF-NCW	0.894	**0.854**	**0.804**	**0.750**	**0.766**	**0.756**	**0.712**	**0.704**	**0.662**	**0.767**
following are kernelized version of each algorithm with kernel function $\langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$										
KM	0.742	0.669	0.616	0.584	0.542	0.512	0.508	0.470	0.435	0.564
KM-NCW	0.873	**0.862**	0.769	0.702	0.684	0.669	0.648	0.620	0.610	0.715
AA	0.744	0.680	0.605	0.604	0.560	0.541	0.556	0.526	0.471	0.587
NC	0.874	0.848	0.783	0.747	0.734	0.728	0.718	0.688	0.651	0.752
CF	0.794	0.708	0.673	0.627	0.610	0.564	0.568	0.538	0.555	0.626
CF-NCW	0.895	0.848	0.750	0.764	0.714	**0.740**	**0.762**	**0.714**	0.631	0.758
LCCF	0.868	0.756	0.752	0.698	0.677	0.606	0.598	0.568	0.590	0.679
LCCF-NCW	**0.908**	0.854	**0.807**	**0.789**	**0.770**	0.736	0.738	0.710	**0.670**	**0.776**

TABLE 7
Clustering performance on Reuters-21578 (Normalized Mutual Information)

k	2	3	4	5	6	7	8	9	10	Avg.
KM	0.428	0.409	0.483	0.428	0.482	0.460	0.393	0.393	0.478	0.439
KM-NCW	**0.627**	0.630	0.655	0.525	0.572	0.522	0.454	0.464	0.540	0.554
AA	0.368	0.353	0.431	0.381	0.430	0.424	0.349	0.363	0.435	0.393
NC	0.596	0.584	0.586	0.500	0.552	0.508	0.434	0.441	0.524	0.525
NMF	0.436	0.410	0.509	0.424	0.485	0.449	0.362	0.391	0.464	0.437
NMF-NCW	0.608	0.641	0.626	0.533	0.576	0.545	0.461	0.471	0.556	0.558
CF	0.440	0.414	0.521	0.425	0.494	0.456	0.363	0.396	0.476	0.443
CF-NCW	0.605	0.648	0.631	0.534	0.582	0.549	0.466	0.475	0.550	0.560
LCCF	0.503	0.452	0.553	0.485	0.519	0.485	0.447	0.445	0.513	0.489
LCCF-NCW	0.599	**0.652**	**0.668**	**0.556**	**0.623**	**0.601**	**0.538**	**0.533**	**0.567**	**0.593**
following are kernelized version of each algorithm with kernel function $\langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$										
KM	0.183	0.206	0.263	0.218	0.260	0.260	0.212	0.231	0.276	0.234
KM-NCW	0.509	0.590	0.539	0.446	0.506	0.468	0.424	0.412	0.474	0.485
AA	0.116	0.141	0.208	0.156	0.195	0.177	0.153	0.169	0.220	0.171
NC	0.490	0.517	0.528	0.433	0.504	0.479	0.442	0.437	0.458	0.476
CF	0.312	0.329	0.401	0.332	0.405	0.377	0.336	0.349	0.428	0.363
CF-NCW	0.567	0.529	0.488	0.446	0.484	0.490	0.528	0.472	0.427	0.492
LCCF	0.503	0.452	0.553	0.485	0.519	0.485	0.447	0.445	0.513	0.489
LCCF-NCW	**0.648**	**0.640**	**0.677**	**0.576**	**0.632**	**0.591**	**0.566**	**0.544**	**0.583**	**0.606**

our knowledge about the two data sets. Since the content of each cluster in TDT2 is narrowly defined and the clusters in TDT2 are compact, it has high chance that each document share the same cluster membership with its p-nearest neighbors even when p is large. While in Reuters, documents in each cluster have a broader variety of content. Thus a document is unlikely to share the same cluster membership with its p-nearest neighbors when p is large. And as a result, the performance of LCCF on Reuters decreases dramatically as the p increases.

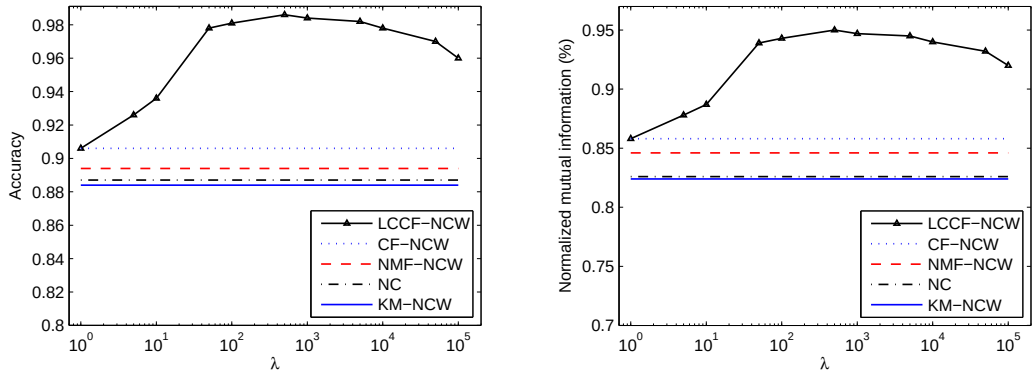
4.5 Convergence Study

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

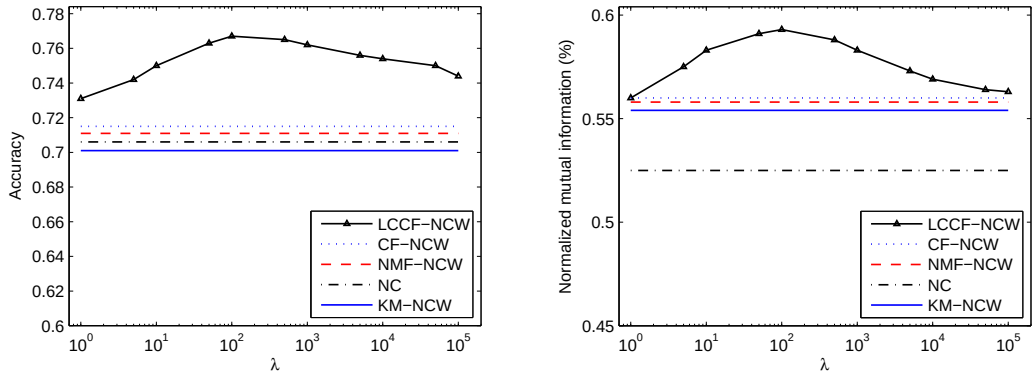
Figure 3 shows the convergence curves of NMF, CF and LCCF on both the two data sets. For each figure, the y-axis is the value of objective function and the x-axis denotes the iteration number. We can see that the multiplicative updating rules for all the three algorithms converge very fast, usually within 100 iterations.

5 Conclusion

In this paper, we have presented a novel clustering algorithm called Locally Consistent Concept Factorization (LCCF). LCCF models the data space as a sub-manifold embedded in the ambient space and performs the concept factorization on this manifold in question. As a result, LCCF can have more discriminating power than



(a) TDT2



(b) Reuters

Fig. 1. The performance of LCCF is very stable with respect to the parameter λ. LCCF achieves consistent good performance with the λ varying from 50 to 10000 on TDT2 data set and from 50 to 1000 on Reuters data set.

the ordinary NMF and CF approaches which only consider the Euclidean structure of the data. Experimental results on document clustering show that LCCF provides better representation in the sense of semantic structure.

There is a parameter λ which controls the smoothness of our LCCF model. LCCF boils down to original CF 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.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 60905001 and 90920303, National Key Basic Research Foundation of China under Grant 2009CB320801, the U.S. National Science Foundation under Grants IIS-08-42769, IIS-09-05215 and the Air Force Office of Scientific Research MURI award FA9550-08-1-0265. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

Appendix A (Proofs of Theorem 1)

The objective function $\mathcal{O}$ of GNMF in Eq. (10) is certainly bounded from below by zero. To prove Theorem 1, we need to show that $\mathcal{O}$ is nonincreasing under the update steps in Eq. (17) and (18). Since the second term of $\mathcal{O}$ is only related to $\mathbf{V}$, we have exactly the same update formula for $\mathbf{U}$ in LCCF as the original CF. Thus, we can use the convergence proof of CF to show that $\mathcal{O}$ is nonincreasing under the update step in Eq. (17). Please see [14][24] for details.

Now we only need to prove that $\mathcal{O}$ is nonincreasing under the update step in Eq. (18). We will follow the similar procedure described in [14]. Our proof will make use of an auxiliary function similar to that used in the Expectation-Maximization algorithm [10]. We begin with the definition of the *auxiliary function*.

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

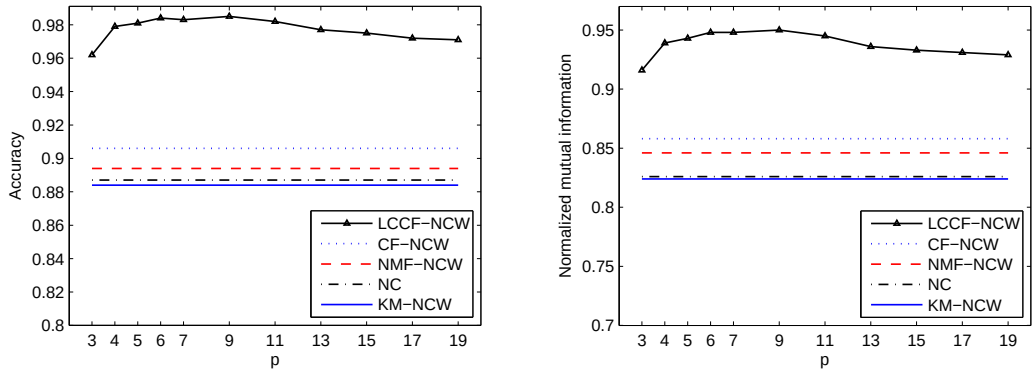
$$G(v, v') \geq F(v), \qquad G(v, v) = F(v)$$

are satisfied.

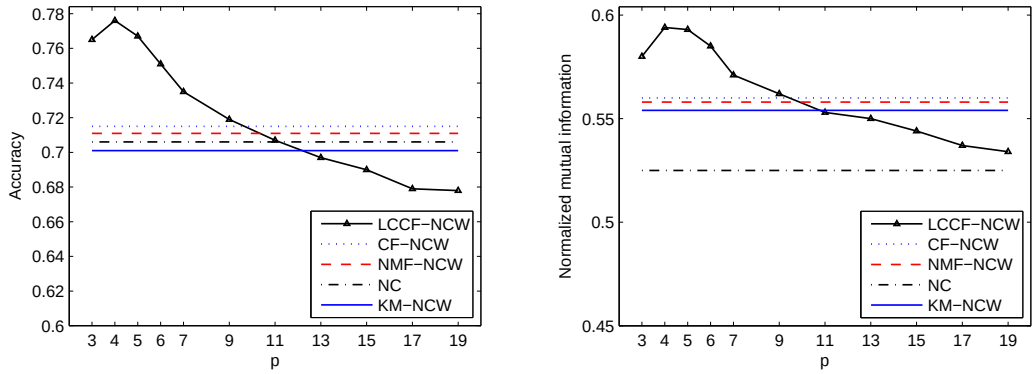
The auxiliary function is very useful because of the following lemma.

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

$$v^{(K+1)} = \arg\min_{v} G(v, v^{(K)}) \tag{31}$$

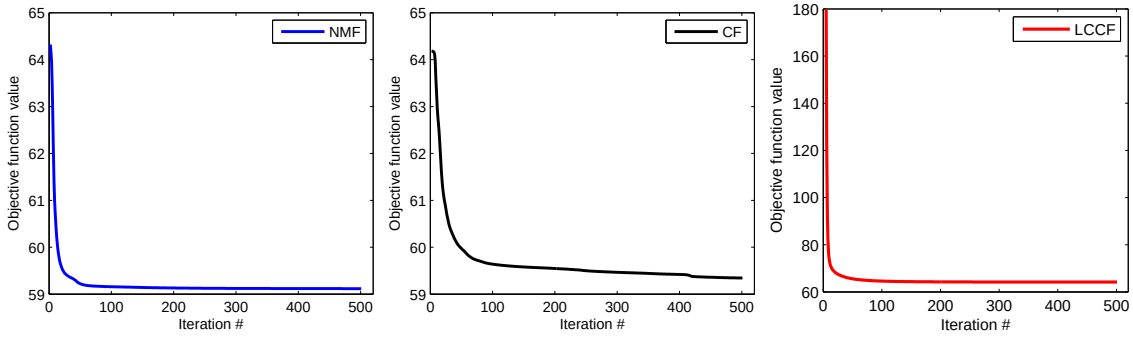


(a) TDT2

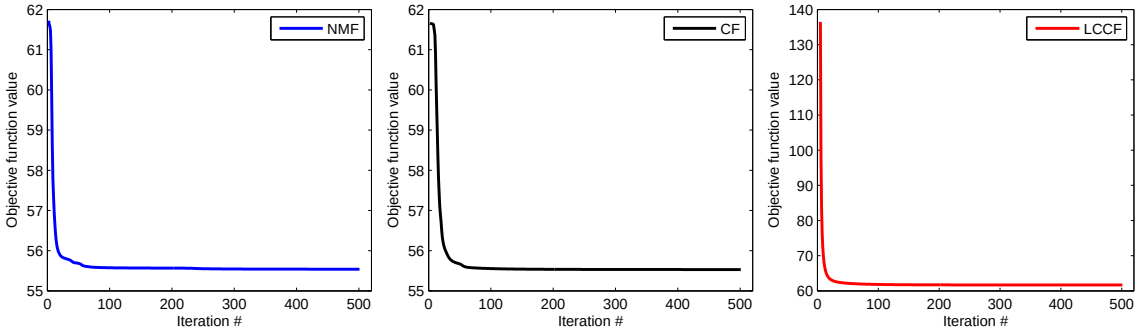


(b) Reuters

Fig. 2. The performance of LCCF decreases as the p increases. The curve is more sensitive with p on Reuters data set because the documents in each cluster of Reuters have a broader variety of content.



(a) Reuters



(b) TDT2

Fig. 3. Convergence curve of NMF, CF and LCCF

Proof:

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

■ □

Now we will show that the update step for $\mathbf{V}$ in Eq. (18) is exactly the update in Eq. (31) with a proper auxiliary function.

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

$$F'_{ab} = \left(\frac{\partial \mathcal{O}}{\partial \mathbf{V}}\right)_{ab} = \left(-2\mathbf{KW} + 2\mathbf{VW}^T\mathbf{KW} + 2\lambda\mathbf{LV}\right)_{ab} \tag{32}$$

$$F''_{ab} = 2\left(\mathbf{W}^T\mathbf{KW}\right)_{bb} + 2\lambda\mathbf{L}_{aa} \tag{33}$$

Since our update is essentially element-wise, it is sufficient to show that each F_{ab} is nonincreasing under the update step of Eq. (18).

Lemma 4: Function

$$\begin{aligned} G(v, v_{ab}^{(K)}) =& F_{ab}(v_{ab}^{(K)}) + F'_{ab}(v_{ab}^{(K)})(v - v_{ab}^{(K)}) \\ &+ \frac{\left(\mathbf{VW}^T\mathbf{KW}\right)_{ab} + \lambda\left(\mathbf{DV}\right)_{ab}}{v_{ab}^{(K)}}(v - v_{ab}^{(K)})^2 \end{aligned} \tag{34}$$

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

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

$$\begin{aligned} F_{ab}(v) =& F_{ab}(v_{ab}^{(K)}) + F'_{ab}(v_{ab}^{(K)})(v - v_{ab}^{(K)}) \\ &+ \left[\left(\mathbf{W}^T\mathbf{KW}\right)_{bb} + \lambda\mathbf{L}_{aa}\right](v - v_{ab}^{(K)})^2 \end{aligned} \tag{35}$$

with Eq. (34) to find that $G(v, v_{ab}^{(K)}) \geq F_{ab}(v)$ is equivalent to

$$\frac{\left(\mathbf{VW}^T\mathbf{KW}\right)_{ab} + \lambda\left(\mathbf{DV}\right)_{ab}}{v_{ab}^{(K)}} \geq \left(\mathbf{W}^T\mathbf{KW}\right)_{bb} + \lambda\mathbf{L}_{aa}. \tag{36}$$

We have

$$\left(\mathbf{VW}^T\mathbf{KW}\right)_{ab} = \sum_{l=1}^{k} v_{al}^{(K)}\left(\mathbf{W}^T\mathbf{KW}\right)_{lb} \geq v_{ab}^{(K)}\left(\mathbf{W}^T\mathbf{KW}\right)_{bb} \tag{37}$$

and

$$\begin{aligned} \lambda\left(\mathbf{DV}\right)_{ab} &= \lambda\sum_{j=1}^{M}\mathbf{D}_{aj}v_{jb}^{(K)} \geq \lambda\mathbf{D}_{aa}v_{ab}^{(K)} \\ &\geq \lambda\left(\mathbf{D} - \mathbf{S}\right)_{aa}v_{ab}^{(K)} = \lambda\mathbf{L}_{aa}v_{ab}^{(K)} \end{aligned} \tag{38}$$

.

Thus, Eq. (36) holds and $G(v, v_{ab}^{(K)}) \geq F_{ab}(v)$. ■ □

We can now demonstrate the convergence of Theorem 1:

Proof: **of Theorem 1** Replacing $G(v, v_{ab}^{(K)})$ in Eq. (31) by Eq. (34) results in the update rule:

$$\begin{aligned} v_{ab}^{(K+1)} &= v_{ab}^{(K)} - v_{ab}^{(K)}\frac{F'_{ab}(v_{ab}^{(K)})}{2\left(\mathbf{VW}^T\mathbf{KW}\right)_{ab} + 2\lambda\left(\mathbf{DV}\right)_{ab}} \\ &= v_{ab}^{(K)}\frac{\left(\mathbf{KW} + \lambda\mathbf{SV}\right)_{ab}}{\left(\mathbf{VW}^T\mathbf{KW} + \lambda\mathbf{DV}\right)_{ab}} \end{aligned} \tag{39}$$

Since Eq. (34) is an auxiliary function, F_{ab} is nonincreasing under this updating rule. ■ □

Appendix B (Weighted LCCF)

In this appendix, we give the solution to the weighted LCCF. Let each data point has weight γ_j and $\mathbf{z}_j^T$ is j-th row vector of $\mathbf{V}$, the weighted objective function is:

$$\begin{aligned} \mathcal{O} &= \sum_{j=1}^{N}\gamma_j\left(\mathbf{x}_j - \mathbf{XWz}_j\right)^T\left(\mathbf{x}_j - \mathbf{XWz}_j\right) + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{LV}) \\ &= \mathrm{Tr}\left((\mathbf{X} - \mathbf{XWV}^T)\Gamma\left(\mathbf{X} - \mathbf{XWV}^T\right)^T\right) + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{LV}) \\ &= \mathrm{Tr}\left(\left(\mathbf{X}\Gamma^{1/2} - \mathbf{XWV}^T\Gamma^{1/2}\right)\left(\mathbf{X}\Gamma^{1/2} - \mathbf{XWV}^T\Gamma^{1/2}\right)^T\right) \\ &\quad + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{LV}) \\ &= \mathrm{Tr}\left(\left(\mathbf{X}\Gamma^{1/2} - \mathbf{XWV}^T\Gamma^{1/2}\right)^T\left(\mathbf{X}\Gamma^{1/2} - \mathbf{XWV}^T\Gamma^{1/2}\right)\right) \\ &\quad + \lambda\,\mathrm{Tr}(\mathbf{V}^T\mathbf{LV}) \\ &= \mathrm{Tr}\left(\left(\mathbf{I} - \mathbf{W}'\mathbf{V}'^T\right)^T\mathbf{K}'\left(\mathbf{I} - \mathbf{W}'\mathbf{V}'^T\right)\right) + \lambda\,\mathrm{Tr}(\mathbf{V}'^T\mathbf{L}'\mathbf{V}') \end{aligned}$$

where Γ is the diagonal matrix consists of γ_j, $\mathbf{W}' = \Gamma^{-1/2}\mathbf{W}$, $\mathbf{V}' = \Gamma^{1/2}\mathbf{V}$, $\mathbf{L}' = \Gamma^{-1/2}\mathbf{L}\Gamma^{-1/2}$ and $\mathbf{K}' = \Gamma^{1/2}\mathbf{K}\Gamma^{1/2}$. Notice that the above equation has the same form as Eq. (11) in Section 3, so the same algorithm can be used to find the solution.

References

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, Cambridge, MA, 2001. 1, 3

[2] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 3

[3] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouimet. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*, 2003. 6

[4] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences*, 101(12):4164–4169, 2004. 2

[5] D. Cai, X. He, and J. Han. Document clustering using locality preserving indexing. *IEEE Transactions on Knowledge and Data Engineering*, 17(12):1624–1637, December 2005. 1, 7

[6] D. Cai, X. He, X. Wu, and J. Han. Non-negative matrix factorization on manifold. In *Proc. Int. Conf. on Data Mining (ICDM'08)*, 2008. 2

[7] F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *Regional Conference Series in Mathematics*. AMS, 1997. 3

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2nd edition, 2001. 4

[9] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990. 1

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 10

[11] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556, 2004. 7

[12] J. Kivinen and M. K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. In *STOC '95: Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 209–218, 1995. 4

[13] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. 2

[14] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*. 2001. 2, 10

[15] X. Li and Y. Pang. Deterministic column-based matrix decomposition. *IEEE Transactions on Knowledge and Data Engineering*, 22(1):145–149, 2010. 2

[16] N. K. Logothetis and D. L. Sheinberg. Visual object recognition. *Annual Review of Neuroscience*, 19:577–621, 1996. 1

[17] L. Lovasz and M. Plummer. *Matching Theory*. Akadémiai Kiadó, North Holland, Budapest, 1986. 7

[18] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, Cambridge, MA, 2001. 1

[19] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 1

[20] F. Sha, Y. Lin, L. K. Saul, and D. D. Lee. Multiplicative updates for nonnegative quadratic programming. *Neural Computation*, 19(8):2004–2031, 2007. 5

[21] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 7

[22] G. W. Stewart. *Matrix Algorithms Volume I: Basic Decompositions*. SIAM, 1998. 4

[23] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1

[24] W. Xu and Y. Gong. Document clustering by concept factorization. In *Proc. 2004 Int. Conf. on Research and Development in Information Retrieval (SIGIR'04)*, pages 202–209, Sheffield, UK, July 2004. 1, 2, 3, 4, 5, 6, 7, 10

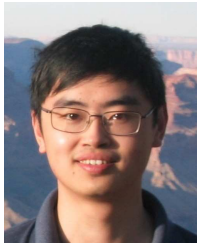
[25] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proc. 2003 Int. Conf. on Research and Development in Information Retrieval (SIGIR'03)*, pages 267–273, Toronto, Canada, Aug. 2003. 1, 2, 6, 7

[26] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao. Binary sparse nonnegative matrix factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(5):772–777, 2009. 2

[27] H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14*, pages 1057–1064. MIT Press, Cambridge, MA, 2001. 7

[28] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, 2003. 3

Deng Cai is an Associate Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the PhD degree in computer science from The University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining and information retrieval.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. 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, CA. His research interests include machine learning, information retrieval, and computer vision.

Jiawei Han is a Professor in the Department of Computer Science at the University of Illinois. He has been working on research into data mining, data warehousing, stream data mining, spatiotemporal and multimedia data mining, information network analysis, text and Web mining, and software bug mining, with over 400 conference and journal publications. He has chaired or served in over 100 program committees of international conferences and workshops and also served or is serving on the editorial boards for Data Mining and Knowledge Discovery, IEEE Transactions on Knowledge and Data Engineering, Journal of Computer Science and Technology, and Journal of Intelligent Information Systems. He is currently the founding Editor-in-Chief of ACM Transactions on Knowledge Discovery from Data (TKDD). Jiawei has received IBM Faculty Awards, the Outstanding Contribution Award at the International Conference on Data Mining (2002), ACM SIGKDD Innovation Award (2004), and IEEE Computer Society Technical Achievement Award (2005). He is a Fellow of ACM and IEEE. His book "Data Mining: Concepts and Techniques" (Morgan Kaufmann) has been used worldwide as a textbook.