

Research Article

Relationship Matrix Nonnegative Decomposition for Clustering

Ji-Yuan Pan and Jiang-She Zhang

Faculty of Science and State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an Shaanxi Province, Xi'an 710049, China

Correspondence should be addressed to Ji-Yuan Pan, panjiyuan@gmail.com

Received 18 January 2011; Revised 28 February 2011; Accepted 9 March 2011

Academic Editor: Angelo Luongo

Copyright © 2011 J.-Y. Pan and J.-S. Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nonnegative matrix factorization (NMF) is a popular tool for analyzing the latent structure of nonnegative data. For a positive pairwise similarity matrix, symmetric NMF (SNMF) and weighted NMF (WNMF) can be used to cluster the data. However, both of them are not very efficient for the ill-structured pairwise similarity matrix. In this paper, a novel model, called relationship matrix nonnegative decomposition (RMND), is proposed to discover the latent clustering structure from the pairwise similarity matrix. The RMND model is derived from the nonlinear NMF algorithm. RMND decomposes a pairwise similarity matrix into a product of three low rank nonnegative matrices. The pairwise similarity matrix is represented as a transformation of a positive semidefinite matrix which pops out the latent clustering structure. We develop a learning procedure based on multiplicative update rules and steepest descent method to calculate the nonnegative solution of RMND. Experimental results in four different databases show that the proposed RMND approach achieves higher clustering accuracy.

1. Introduction

Nonnegative matrix factorization (NMF) [1] has been introduced as an effective technique for analyzing the latent structure of nonnegative data such as images and documents. A variety of real-world applications of NMF has been found in many areas such as machine learning, signal processing [2-4], data clustering [5, 6], and computer vision [7].

Most applications focus on the clustering aspect of NMF [8, 9]. Each sample can be represented as a linear combination of clustering centroids. Recently, a theoretic analysis has shown the equivalence between NMF and K -means/spectral clustering [10]. Symmetric NMF (SNMF) [10] is an extension of NMF. It aims at learning clustering structure from the kernel matrix or pairwise similarity matrix which is positive semidefinite. When the similarity matrix is not positive semidefinite, SNMF is not able to capture the clustering structure

contained in the subspace associated with negative eigenvalues. In order to overcome the limitation, weighted NMF (SNMF) [10] is developed. In the WNMF model, the indefiniteness of the pairwise similarity matrix is passed onto a specific low-rank matrix. WNMF improves the clustering performance of SNMF. When a portion of data is labeled, it is desirable to incorporate the class labels information into WNMF in order to improve the clustering performance. To this end, a semisupervised NMF (SSNMF) [11] is studied by incorporating the domain knowledge into WNMF to extract more clustering structure information.

In SNMF, WNMF, and SSNMF, the low rank approximation to the pairwise similarity matrix is used. The goal is to learn the latent clustering structure by minimizing the reconstruction error. However, since there is no prior knowledge about the data, the kernel matrix is often obtained based on pairwise Euclidean distance in the high-dimensional space. It is more sensitive to the unexpected noise. Consequently, it may produce undesirable performances in clustering tasks by minimizing the objective function from the viewpoint of reconstruction in the form as SNMF, WNMF, and SSNMF. In this paper, we present a novel model, called relationship matrix nonnegative decomposition (RMND), for data clustering tasks. The RMND model is derived from the nonlinear NMF algorithms which take advantages of kernel functions in the high-dimensional feature space. RMND decomposes a pairwise similarity matrix into a product of a positive semidefinite matrix, a distribution matrix of similarity on latent features, and an encoding matrix. The positive semidefinite matrix pops out the clustering structure and is treated as a more convincing pairwise similarity matrix by an appropriate transformation. RMND learns the correct relationship matrix adaptively. Furthermore, according to the positive semidefiniteness, the SNMF formulation is incorporated in RMND, and then a more tractable representation of pairwise similarity matrix is obtained. We develop a learning procedure for RMND to discover the latent clustering structure. Experimental results show that the proposed RMND leads to significant improvements on clustering performance.

The rest of the paper is organized as follows: in Section 2, we briefly review the SNMF and WNMF. In Section 3, we present the proposed RMND model and its learning procedure. Some experimental results on several datasets are shown in Section 4. Finally, conclusions and final remarks are given in Section 5.

2. Symmetric NMF (SNMF) and Weighted NMF (WNMF)

A pairwise similarity matrix X is a nonnegative matrix, since the pairwise similarities between different objects cannot be negative. For the kernel case, $X = V^T V$ is the standard inner-product linear kernel matrix, where V is a nonnegative data matrix with size $p \times n$. And it can be extended to any other kernels. NMF technique is powerful to discover the latent structure in X . Since X is a symmetric matrix, Ding et al. [10] introduced the SNMF model as follows:

$$X \approx AA^T, \quad (2.1)$$

where A is a nonnegative matrix of size $n \times q$ whose rows denote the degrees of the samples related to the q centroids of clusters.

In (2.1), AA^T is a positive semidefinite matrix. When the similarity matrix X is indefinite, X has negative eigenvalues. AA^T will not provide a good approximation, since AA^T cannot absorb the subspace associated with negative eigenvalues. For kernel matrices,

decomposition in (2.1) is feasible. However, a large number of similarity matrices are nonnegative but not positive semidefinite matrix. Ding et al. [10] introduced another improved factorization model as

$$X \approx ASA^T, \quad (2.2)$$

where S is a nonnegative matrix of size $q \times q$ which inherits the indefiniteness of X . The detailed update rules for A and S can be found in [10].

3. Relationship Matrix Nonnegative Decomposition (RMND)

3.1. Proposed RMND Model

Both SNMF and WNMF are powerful methods for learning the clustering structure from a pairwise similarity matrix or a kernel matrix. If the data from different latent classes are well separated, X would be approximately a block diagonal matrix. It is easy to find the clustering structure by SNMF and WNMF in that case. Thus, AA^T and ASA^T would be approximately block diagonal matrices. SNMF and WNMF learn good approximation for X . This is the simplest case in data clustering. In many real applications, data from different latent classes often crossly corrupt. Then, X is not a block diagonal matrix although the data is rearranged appropriately. By minimizing the reconstruction error in (2.1) and (2.2), SNMF and WNMF would not find favorable clustering structure. The reason is AA^T and ASA^T would be approximately block diagonal matrices for good clustering. Consequently, it is desirable to build a new model for finding correct clustering structure and approximating X well at the same time.

Recently, NMF has been extended to nonlinear nonnegative component analysis algorithms (referred as KNMF) by Zafeiriou and Petrou [12]. KNMF is proposed to model efficiently the nonlinearities that are present in most real-life applications. The idea of KNMF is to perform NMF in the high-dimensional feature space. Specifically, KNMF is to find a set of nonnegative weights and nonnegative basis vectors such that the nonlinearly mapped training vectors can be written as linear combinations of nonlinear mapped nonnegative basis vectors. Let $\phi : \mathcal{R}_+^p \rightarrow F$ be a mapping that projects data v_i to a Hilbert space F of arbitrary dimensionality. KNMF attempts to find a set of q vectors $w_j \in \mathcal{R}_+^p$ and a set of nonnegative weights h_{ji} such that

$$V^\Phi \approx W^\Phi H, \quad (3.1)$$

where $V^\Phi = [\phi(v_1), \dots, \phi(v_n)]$ and $W^\Phi = [\phi(w_1), \dots, \phi(w_q)]$. The nonlinear mapping is related to a kernel function with the operation as $k(v_i, v_j) = \phi(v_i)^T \phi(v_j)$. The detailed algorithms for directly learning W and H can be found in [12].

In this paper, we focus on the convex nonlinear nonnegative component analysis algorithm (referred as CKNMF) in [12]. Instead of finding both W and H simultaneously, Zafeiriou and Petrou followed the similar lines as convex-NMF [9] and assumed that

the centroid $\phi(w_j)$ is in the space spanned by the columns of V^Φ . Formally, $\phi(w_j)$ can be written as

$$\phi(w_j) = m_{1j}\phi(v_1) + \cdots + m_{nj}\phi(v_n) = \sum_{l=1}^n m_{lj}\phi(v_l), \quad (3.2)$$

where $m_{ij} \geq 0$ and $\sum_{l=1}^n m_{lj} = 1$. This means that the centroid $\phi(w_j)$ can be interpreted as a convex weighted combination of certain data point $\phi(v_l)$. Using (3.2), approximation (3.1) is reformulated in the matrix form as

$$V^\Phi \approx V^\Phi MH \implies X \approx XMH, \quad (3.3)$$

where X is the kernel matrix with the entry $x_{ij} = k(v_i, v_j) = \phi(v_i)^T \phi(v_j)$. Equation (3.3) provides a new decomposition of kernel matrix. Each matrix has the explicit interpretation. X is the relationship matrix between different objects based on a certain kernel function, each column of M denotes the relationship distribution on certain latent feature according to the property of convex combinations, and H is the encoding coefficient matrix. In particular, we rewrite (3.3) in an entry form as

$$x_{ij} \approx \sum_{r=1}^n \sum_{s=1}^q x_{ir} m_{rs} h_{sj} = \sum_{s=1}^q (XM)_{is} h_{sj}. \quad (3.4)$$

It can be noted from (3.4) that $(XM)_{is}$ represents the weighted average relationship measure correlated to object i on the s th latent feature, and then the relationship measure between object i and j is a linear combination of the weighted average relationship measures on the latent features.

However, (3.3) or (3.4) is not convincing for clustering tasks, since the kernel matrix X cannot represent the relationship between different objects faithfully. It is more desirable to discover the latent relationship adaptively. Consequently, we replace the X in the right hand side in (3.3) by a latent relationship matrix

$$X \approx RMH, \quad (3.5)$$

where R denotes the correct relationship matrix. From (3.5), the correct relationship matrix R would be adaptively learned from the kernel matrix X . X is a linear transformation of R . A relationship matrix R , which pops out the latent clustering structure, is approximately a block diagonal matrix under suitable rearrangement on samples. It would be a positive semidefinite matrix. SNMF model is reasonable to learn a low rank representation of matrix R . Thus, we derive our new model, referred as relationship matrix nonnegative decomposition (RMND), as follows:

$$X \approx AA^T MH, \quad (3.6)$$

where A is a nonnegative matrix whose rows denote the degrees of the samples related to the centroids of clusters. The corresponding optimization problem of RMND is given by

$$\begin{aligned} \min_{A, M, H} \quad & D = \frac{1}{2} \|X - AA^T MH\|^2 \\ \text{s.t.} \quad & A \geq 0, \quad M \geq 0, \quad H \geq 0, \quad \sum_i m_{ij} = 1, \quad \forall j = 1, 2, \dots, q. \end{aligned} \quad (3.7)$$

The objective function D of RMND in (3.7) is not convex for A , M , and H simultaneously. Therefore, it is unrealistic to expect an algorithm to find the global minimum of D . As it is known that $AA^T MH = AA^T MUU^{-1}H$, where U is a diagonal matrix with $u_{jj} = \sum_i m_{ij}$, the normalization on M can be easily handled after M is updated. Therefore, we only consider the nonnegativity constraints on the factors. When A is fixed, let γ_{jk} and ψ_{jk} be the Lagrange multiplier for constraints $m_{jk} \geq 0$ and $h_{jk} \geq 0$, respectively. We define matrix $\Gamma = [\gamma_{jk}]$ and $\Psi = [\psi_{jk}]$, then the Lagrange multiplier \mathcal{L} is

$$\mathcal{L} = \frac{1}{2} \|X - AA^T MH\|^2 + \text{Tr}(\Gamma M^T) + \text{Tr}(\Psi H^T), \quad (3.8)$$

where $\|\cdot\|$ denotes the Euclidean norm and $\text{Tr}(\cdot)$ is the trace function. The partial derivatives of \mathcal{L} with respect to M and H are

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial M} &= AA^T AA^T MHH^T - AA^T XH^T + \Gamma, \\ \frac{\partial \mathcal{L}}{\partial H} &= M^T AA^T AA^T MH - M^T AA^T X + \Psi. \end{aligned} \quad (3.9)$$

Using the KKT conditions $\gamma_{jk} m_{jk} = 0$ and $\psi_{jk} h_{jk} = 0$, we get the following equations:

$$\begin{aligned} (AA^T AA^T MHH^T)_{jk} m_{jk} - (AA^T XH^T)_{jk} m_{jk} &= 0, \\ (M^T AA^T AA^T MH)_{jk} h_{jk} - (M^T AA^T X)_{jk} h_{jk} &= 0. \end{aligned} \quad (3.10)$$

The above equations lead to the following multiplicative update rules:

$$\begin{aligned} m_{jk} &\leftarrow m_{jk} \frac{(AA^T XH^T)_{jk}}{(AA^T AA^T MHH^T)_{jk}}, \\ h_{jk} &\leftarrow h_{jk} \frac{(M^T AA^T X)_{jk}}{(M^T AA^T AA^T MH)_{jk}}. \end{aligned} \quad (3.11)$$

For factor matrix A , the corresponding partial derivatives of D is

$$\frac{\partial D}{\partial A} = AA^T MHH^T M^T A + MHH^T M^T AA^T A - XH^T M^T A - MHXA. \quad (3.12)$$

Input: Positive matrix $X \in \mathcal{R}^{n \times n}$, and a positive integer q .
Output: Nonnegative factor matrices $A \in \mathcal{R}^{n \times q}$, $M \in \mathcal{R}^{n \times q}$ and $H \in \mathcal{R}^{q \times n}$.
Learning Procedure
(S1) Initialize A , M and H to random positive matrices, and normalize each column of A to one.
(S2) Repeat the iterations until convergence:
(1) $H \leftarrow H \otimes (M^T A A^T X) \oslash (M^T A A^T A A^T M H)$.
(2) $M \leftarrow M \otimes (A A^T X H^T) \oslash (A A^T A A^T M H H^T)$.
(3) $M \leftarrow M U^{-1}$ and $H \leftarrow U H$ where U is a diagonal matrix with $u_{jj} = \sum_i m_{ij}$.
(4) Repeatedly select small positive constant μ_A until the objective function is decreased.
(i) $\tilde{A} := A - \mu_A \frac{\partial D}{\partial A}$.
(ii) Project each column of \tilde{A} to be nonnegative vector with unit L_2 norm.
(5) $A := \tilde{A}$.
Above, \otimes and \oslash denote elementwise multiplication and division, respectively.

Algorithm 1: Relationship matrix nonnegative decomposition (RMND).

Our algorithm essentially takes a step in the direction of the negative gradient and, subsequently, projects onto the constraint space, making sure that the taken step is small enough that the objective function D is reduced at every step. The learning procedure for RMND can be summarized as Algorithm 1.

3.2. Computational Complexity Analysis

In this subsection, we discuss the extra computational cost of our proposed algorithm in comparison with SNMF and WNMF. We count the arithmetic operations for each algorithm. Based on the updating rules in [10], it is not hard to count the arithmetic operations of each iteration in SNMF and WNMF. For GNMF, the steepest descent method is used to update factor matrix A . We use the bipartition to determine the small positive constant μ_A . Let N_1 be the maximum iteration number in the steepest descent method. We summary the computational operation counts for each iteration in Table 1. Suppose that the algorithms stop after N_2 iterations and the overall cost for both SNMF and WNMF is

$$O(N_2 q n^2). \quad (3.13)$$

The overall cost for RMND is

$$O(N_2 N_1 q n^2). \quad (3.14)$$

Then, the overall cost of SNMF, WNMF, and RMND is related to $q n^2$, where n is the number of samples. Much time is needed for large-scale data clustering tasks. For RMND, its overall cost is also effected by the maximum iteration number N_1 in the steepest descent method. Nevertheless, RMND will be shown that it is capable of improving the clustering performance in Section 4. We will develop algorithms for fast convergence and low computational complexity in the future work.

Table 1: Computational operation counts for each iteration in SNMF, WNMF, and RMND.

Method	Fladd*	fmlt*	fldiv°	overall
SNMF	$qn^2 + 2q^2n + 2qn$	$qn^2 + 2q^2n + 2qn$	qn	$O(qn^2)$
WNMF	$2qn^2 + 4q^2n + 2qn + 2q^3$	$2qn^2 + 4q^2n + 2qn + 2q^3 + q^2$	$qn + q^2$	$O(qn^2)$
RMND	$(1 + 4N_1)qn^2 + (10 + 9N_1)q^2n$ $+ (4 + 4N_1)qn + (4 + 3N_1)q^3$ $+ N_1q + N_1n^2$	$(1 + 4N_1)qn^2 + (10 + 9N_1)q^2n$ $+ (4 + 3N_1)qn + (4 + 3N_1)q^3 + N_1n^2$	$(3 + N_1)qn$	$O(N_1qn^2)$

* fladd: a floating-point addition. *fmlt: a floating-point multiplication. °fldiv: a floating-point division.

4. Numerical Experiments

We evaluate the performance of five different methods, RMND, K -means clustering, spectral clustering (SpeClus) [13], SNMF, and WNMF, in a task of data clustering. In RMND, once A is learned, we denote $\hat{X} = \hat{A}\hat{A}^T$, where \hat{A} is the modification of A with normalized rows. We apply K -means clustering on A , H and the factor matrices learned by SNMF and WNMF on \hat{X} . Finally, the best clustering results of RMND is obtained.

4.1. Datasets

We use five datasets for evaluating the clustering performance of algorithms. The detailed description for the datasets is listed below.

- (1) *JAFFE* [14] is a face database often used in the literature of face recognition. JAFFE database contains 213 face images from 10 different persons under varying in facial expression.
- (2) *Coil20* is a dataset consisting of 1440 images from 20 classes under varying in rotations (<http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php>). For simplicity, the first 10 classes are used in our experiments.
- (3) *reuters* dataset [15] is from the news articles of the Reuters newswire. Reuters-21578 corpus contains 21,578 documents in 135 categories. We use the data preprocessed by Cai et al. [15]. The documents with multiple category labels are discarded and the dataset with 8067 documents in the largest 30 categories is derived. Then, we randomly select at most 5 categories for efficiency.
- (4) *USPS* is a dataset of handwritten digits from 0 to 9 (<http://www.zjucadcg.cn/dengcai/Data/MLData.html>). These image data have been preprocessed by Cai et al. [15]. The *USPS* dataset used here consists of mixed-sign data. The first 100 samples from each class are used in our experiments.

4.2. Evaluation Metrics for Clustering and Kernel Function

In all these methods, we set q , the dimensionality of feature subspace, to be equal to the number of classes of datasets. Two performance measures (clustering accuracy and normalized mutual information) are used to evaluate the clustering performance of algorithms. If we denote the true label for the i th data to be c_i , and the estimated label \hat{c}_i , the clustering accuracy can be computed by $\sum_{i=1}^n (\delta(c_i, \hat{c}_i) / n)$, where $\delta(x, y) = 1$ for $x = y$ and $\delta(x, y) = 0$ for $x \neq y$. The clustering accuracy achieves maximum value 1 when clustering

results are perfect. Let T be the set of clusters obtained from the ground truth and T' obtained from our algorithm. The normalized mutual information measure is defined by

$$\text{NMI} = \frac{\text{MI}(T, T')}{\max(H(T), H(T'))}, \quad (4.1)$$

where $\text{MI}(T, T')$ is the mutual information between T and T' , $H(T)$ and $H(T')$ denote the entropies of T and T' , respectively. The value of NMI varies between 0 and 1. The greater the normalized mutual information, the better the clustering quality.

In our experiments, we use Gaussian kernel function to calculate the kernel matrix. And then, we evaluate the clustering performance of different algorithms. For comparison, the same kernel matrix has been used in our experiments. The Gaussian kernel function used here is as follows:

$$g(y, z) = e^{-\|y-z\|^2/2t^2}, \quad (4.2)$$

where t is a parameter. X is given by

$$x_{ij} = \begin{cases} e^{-\|v_i-v_j\|^2/2t^2}, & v_i \in N_l(v_j) \text{ or } v_j \in N_l(v_i), \\ 0, & \text{otherwise,} \end{cases} \quad (4.3)$$

where t is a heat kernel parameter [16] and $N_l(v_j)$ denotes a set of l nearest neighbors of v_j . For simplicity, we present a tunable way to set t as a square average distance between different samples [17]

$$t = \sqrt{\frac{m}{2n^2} \sum_{i,j} \|v_i - v_j\|^2}, \quad (4.4)$$

where m is a scale factor.

4.3. Experimental Results on the JAFFE Dataset

To demonstrate how our method improves the performance of data clustering, we firstly set the number of nearest neighbors $l = n - 1$, the scale factor $m = 1$, where n is the number of samples. Then, the pairwise similarity matrix X is the weighted adjacency matrix of the fully connected graph similar to those in spectral clustering. Figure 1 displays the pairwise similarity matrix obtained from the *JAFFE* dataset. It can be noted that X is ill structured. In order to discover the latent clustering structure, we apply RMND, SNMF, and WNMF algorithms to obtain the decomposition form of X , respectively. The factor matrices are randomly initialized by the values in the range $[0, 1]$. Figure 2 shows that objective function value decreases with increasing iteration number. It can be noted that the reconstruction error of RMND is smaller than those of SNMF and WNMF after 500 iterations. Figures 3, 4, and 5 display the estimated pairwise similarity matrix of SNMF, WNMF, and RMND algorithms, respectively. The estimated pairwise similarity matrix learned by RMND is more highly structured. RMND produces better representations of kernel matrix than SNMF and WNMF. SNMF and WNMF have similar representations of X when the algorithms are convergent.

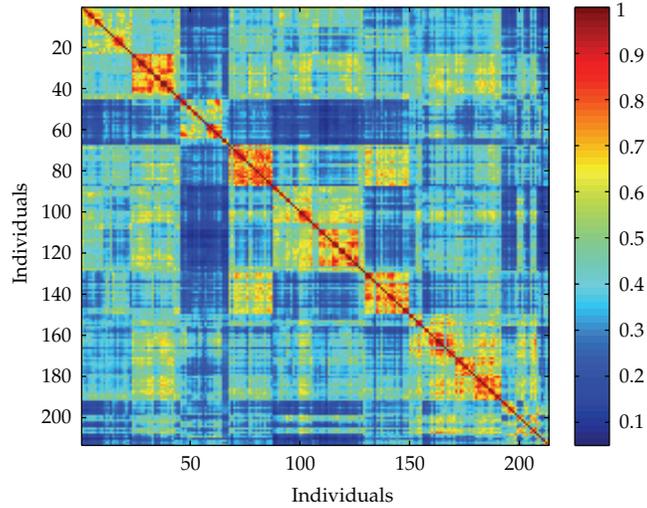


Figure 1: The pairwise similarity matrix X obtained from the *JAFFE* dataset.

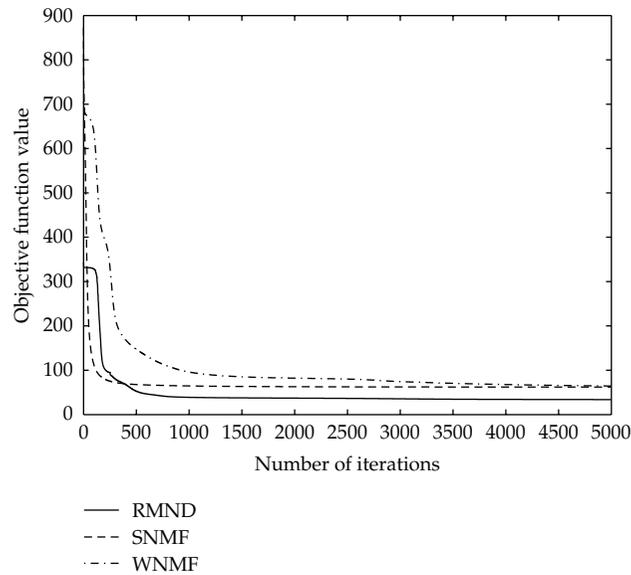


Figure 2: Objective function values as a function of the number of iterations for RMND, SNMF, and WNMF on the *JAFFE* dataset.

4.4. Clustering Performance Comparison

Tables 2, 3, and 4 show the experimental results on the *Coil20*, *Reuters*, and *USPS* datasets with the number of nearest neighbors $l = n - 1$ and the scale factor $m = 1$, respectively. The evaluations are conducted with different numbers of classes, ranging from 4 to 12 for the *Coil20* dataset, 2 to 5 for the *Reuters* dataset, and 2 to 10 for the *USPS* dataset. The cluster number indicates the class number used for experiments. In our experiments, the first k classes in the database are used. For each given class number k , 20 independent tests are

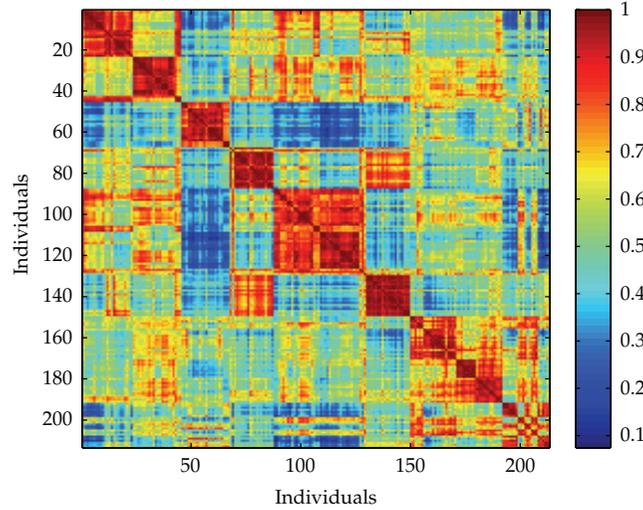


Figure 3: Estimated pairwise similarity matrix by applying SNMF to the *JAFFE* dataset.

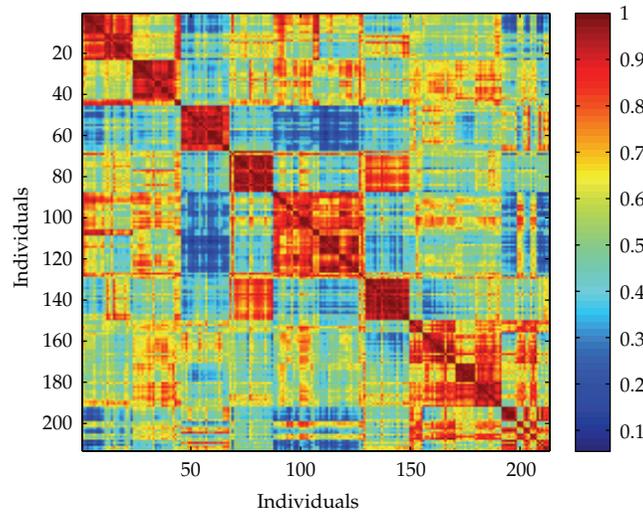


Figure 4: Estimated pairwise similarity matrix by applying WNMF to the *JAFFE* dataset.

conducted under different initializations. For the *Reuters* dataset, different randomly chosen classes are used. And the average performance is calculated over these 20 tests. For each test, *K*-means algorithm is applied 5 times with different start points, and the best result in terms of the objective function of *K*-means is recorded. From these tables, it can be noticed that RMND yields the best average clustering results on the three datasets although the clustering accuracy of RMND is a little smaller than those of other methods for certain cases of class numbers.

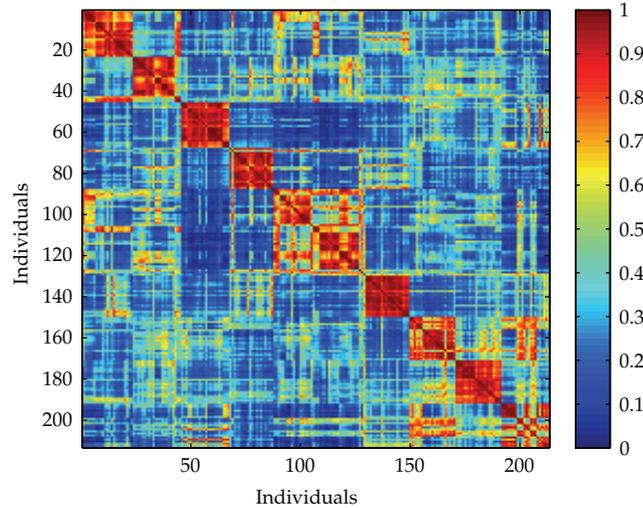


Figure 5: Estimated pairwise similarity matrix by applying RMND to the *JAFFE* dataset.

Table 2: Clustering accuracy and normalized mutual information of K -means, spectral clustering (SpeClus), SNMF, WNMF, and RMND on the *Coil20* dataset.

Cluster number	Clustering accuracy					Normalized mutual information				
	K -means	SpeClus	SNMF	WNMF	RMND	K -means	SpeClus	SNMF	WNMF	RMND
4	0.6602	0.6736	0.6700	0.6682	0.6811	0.5173	0.5494	0.5245	0.5326	0.5578
6	0.4807	0.4816	0.4670	0.4852	0.5054	0.4572	0.4669	0.4283	0.4342	0.4456
8	0.5128	0.5412	0.4967	0.4702	0.5444	0.5312	0.5632	0.5346	0.4940	0.5634
10	0.5355	0.5594	0.5543	0.5104	0.5532	0.6048	0.6002	0.6080	0.5632	0.6055
avg.	0.5473	0.5640	0.5470	0.5335	0.5710	0.5276	0.5449	0.5135	0.5060	0.5431

Table 3: Clustering accuracy and normalized mutual information of K -means, spectral clustering (SpeClus), SNMF, WNMF, and RMND on the *Reuters* dataset.

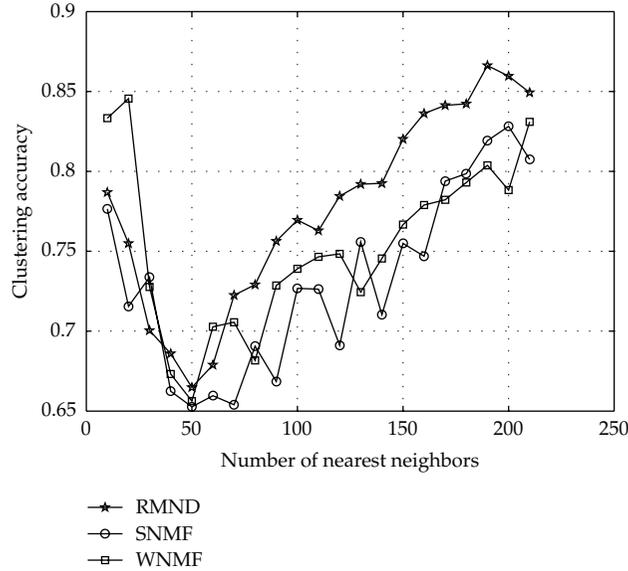
Cluster number	Clustering accuracy					Normalized mutual information				
	K -means	SpeClus	SNMF	WNMF	RMND	K -means	SpeClus	SNMF	WNMF	RMND
2	0.6325	0.7909	0.7665	0.7023	0.8457	0.1621	0.4119	0.3322	0.2584	0.5294
3	0.5536	0.6926	0.6857	0.6538	0.8022	0.2427	0.4999	0.4916	0.4391	0.6242
4	0.5301	0.6216	0.5766	0.5375	0.6634	0.2893	0.5064	0.4138	0.3097	0.5291
5	0.4366	0.5888	0.5635	0.4632	0.5817	0.2325	0.5567	0.4754	0.2950	0.5047
ave.	0.5382	0.6735	0.6506	0.5892	0.7233	0.2328	0.4937	0.4283	0.3246	0.5469

4.5. Clustering Performance Evaluation on Various Pairwise Similarity Matrix

In graph embedding methods, the pairwise similarity matrix (also referred as affinity matrix) has been widely used. In this subsection, we test our algorithm under different adjacency graph constructions to show how the different graph structures will affect the clustering

Table 4: Clustering accuracy and normalized mutual information of K -means, spectral clustering (SpeClus), SNMF, WNMF, and RMND on the *USPS* dataset.

Cluster number k	Clustering accuracy					Normalized mutual information				
	K -means	SpeClus	SNMF	WNMF	RMND	K -means	SpeClus	SNMF	WNMF	RMND
2	0.9650	0.9650	0.9650	0.9660	0.9900	0.8135	0.8135	0.9650	0.9660	0.9900
3	0.8867	0.8467	0.8700	0.8402	0.8912	0.6917	0.6380	0.6689	0.6656	0.7189
4	0.8015	0.7850	0.8057	0.7934	0.8421	0.6430	0.5904	0.6012	0.6118	0.6634
5	0.8392	0.7700	0.7864	0.7686	0.8354	0.6826	0.6147	0.6252	0.6195	0.6586
6	0.7330	0.7300	0.7206	0.6928	0.7325	0.6140	0.6110	0.5921	0.5633	0.6061
7	0.7194	0.7186	0.7155	0.6787	0.7331	0.6041	0.6063	0.5925	0.5656	0.6116
8	0.7199	0.7302	0.7194	0.7025	0.7463	0.6239	0.6320	0.6232	0.5987	0.6494
9	0.6932	0.7358	0.6937	0.6702	0.7007	0.5902	0.6406	0.5910	0.5640	0.5922
10	0.6343	0.6860	0.6527	0.6216	0.6725	0.5556	0.6023	0.5567	0.5333	0.5605
ave.	0.7769	0.7741	0.7699	0.7482	0.7938	0.6465	0.6388	0.6462	0.6320	0.6723

**Figure 6:** Clustering accuracies derived by applying RMND, SNMF, and WNMF on the *JAFFE* database versus different number of nearest neighbors.

performance. The number of nearest neighbors used in this paper defines the locality of graph. In Figures 6 and 7, we show the relationship between the average clustering accuracies and normalized mutual information versus different numbers of nearest neighbors under 20 independent runs and the scale factor $m = 1$, respectively. As can be seen, RMND performs better when the number of nearest neighbors is larger than 60 and the maximum achieved clustering accuracy is 86.62% when 190 nearest neighbors are used in (4.3). The normalized mutual information is better after $l > 150$, and the maximum normalized mutual information is 0.8429 at $l = 210$. For SNMF and WNMF, the best clustering accuracies are 84.55% and 82.82%, respectively, and the best normalized mutual information are 0.8373 and 0.8214,

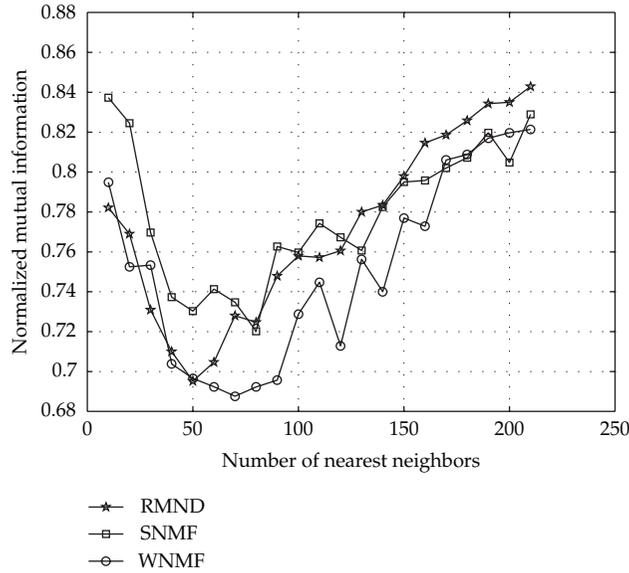


Figure 7: Normalized mutual information derived by applying RMND, SNMF, and WNMF on the *JAFFE* database versus different number of nearest neighbors.

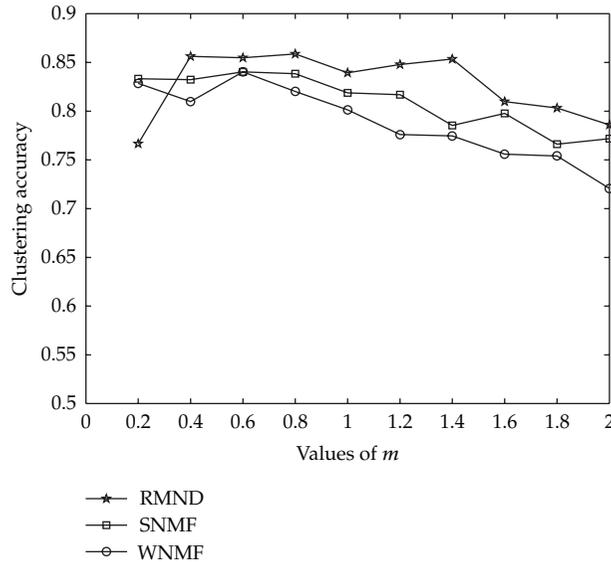


Figure 8: Clustering accuracies derived by applying RMND, SNMF, and WNMF on the *JAFFE* database versus different scale factor m .

respectively. This implies that RMND is more suitable to discover the clustering structure contained in smoothed pairwise similarity matrix.

Note that the choice of parameters in (4.4) is still an open problem. To this end, we explore the range of possible values of the scale factor m to determine the heat kernel parameter. Specifically, m is taken from $\{0.2, 0.4, \dots, 2.0\}$. Figures 8 and 9 show the clustering

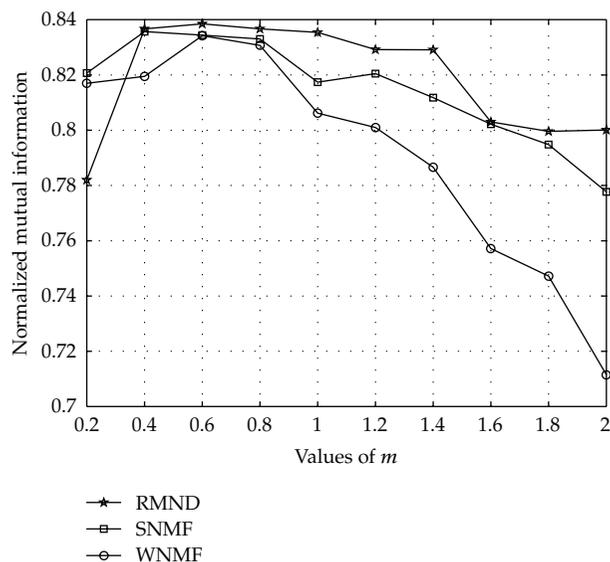


Figure 9: Normalized mutual information derived by applying RMND, SNMF, and WNMF on the *JAFFE* database versus different scale factor m .

accuracies and normalized mutual information on the *JAFFE* dataset under different scale factors. $n - 1$ nearest neighbors are used in this experiment. As m increases, the performance decreases. The reason might be that the difference between different pairwise similarity is small for larger value of m . The pairwise similarity matrix becomes more and more ill structured. Nevertheless, RMND leads to better clustering performance compared with SNMF and WNMF.

5. Conclusions and Future Work

We have presented a novel relationship matrix nonnegative decomposition (RMND) model for data clustering task. The RMND model is formulated by decomposing a pairwise similarity into a product of three low-rank nonnegative matrices which have explicit interpretation. The correct relationship matrix is adaptively learned from the pairwise similarity matrix by RMND. We develop a learning procedure based on multiplicative update rules and steepest descent method to calculate the nonnegative solution of RMND. Extensive numerical experiments confirm that (1) RMND provides a favorable low-rank representation of pairwise similarity matrix. (2) By using an appropriate kernel function, the ability of RMND, SNMF, and WNMF to deal with mixed-signed data makes them useful for many applications in contrast to original NMF. (3) RMND improves the clustering performance of SNMF and WNMF.

Further future work includes the following topics. The first is to develop algorithms for fast convergence and better solution in terms of minimizing the objective function. The second is to investigate the ability of RMND on different kinds of kernel functions.

Acknowledgments

This work was supported by the National Basic Research Program of China (973 Program) (Grant no. 2007CB311002), the National Natural Science Foundation of China (Grant nos. 60675013, 10531030 and 61075006) and Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20100102120048). The authors would like to thank the helpful comments which lead to substantial improvements of the paper.

References

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.
- [2] I. Buciu, N. Nikolaidis, and I. Pitas, "A comparative study of NMF, DNMF and LNMF algorithms applied for face recognition," in *Proceedings of the IEEE-EURASIP, Control Communications and Signal Processing*, 2006.
- [3] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403–415, 2006.
- [4] P. O. Hoyer, "Nonnegative sparse coding," in *Proceedings of the IEEE Workshop Neural Networks for Signal Processing*, 2002.
- [5] W. Xu, X. Liu, and Y. Gong, "Document clustering based on nonnegative matrix factorization," in *Proceedings of the ACM SIGIR Conference Research and Development in Information Retrieval (SIGIR '03)*, Toronto, Canada, 2003.
- [6] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *IEEE Signal Processing Letters*, vol. 17, no. 1, pp. 4–7, 2010.
- [7] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [8] C.-H. Zheng, D.-S. Huang, L. Zhang, and X.-Z. Kong, "Tumor clustering using nonnegative matrix factorization with gene selection," *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 4, pp. 599–607, 2009.
- [9] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, 2010.
- [10] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the SIAM Data Mining Conference*, pp. 606–610, 2005.
- [11] Y. Chen, M. Rege, M. Dong, and J. Hua, "Non-negative matrix factorization for semi-supervised data clustering," *Knowledge and Information Systems*, vol. 17, no. 3, pp. 355–379, 2008.
- [12] S. Zafeiriou and M. Petrou, "Nonlinear non-negative component analysis algorithms," *IEEE Transactions on Image Processing*, vol. 19, no. 4, pp. 1050–1066, 2010.
- [13] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [14] M. Lyons, S. Akamausu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proceedings of 3rd IEEE Conference Face and Gesture Recognition*, pp. 200–205, 1998.
- [15] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
- [16] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.
- [17] R. He, B. G. Hu, X. T. Yuan, and W. S. Zeng, "Principal component analysis based on nonparametric maximum entropy," *Neurocomputing*, pp. 1840–1852, 2010.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

