Improved Hypergraph Regularized Nonnegative Matrix Factorization with Sparse Representation

Sheng Huang\textsuperscript{a,b,*}, Hongxing Wang\textsuperscript{b}, Yongxin Ge\textsuperscript{b}, Luwen Huangfu\textsuperscript{c}, Xiaohong Zhang\textsuperscript{a,b}, Dan Yang\textsuperscript{b}

\textsuperscript{a}Ministry of Education Key Laboratory of Dependable Service Computing in Cyber Physical Society, Chongqing, 400044, China
\textsuperscript{b}School of Software Engineering at Chongqing University, Chongqing, 400044, China
\textsuperscript{c}Eller College of Management at University of Arizona, Tucson AZ, 85712, USA

Abstract

As a commonly used data representation technique, Nonnegative Matrix Factorization (NMF) has received extensive attentions in the pattern recognition and machine learning communities over decades, since its working mechanism is in accordance with the way how the human brain recognizes objects. Inspired by the remarkable successes of manifold learning, more and more researchers attempt to incorporate the manifold learning into NMF for finding a compact representation, which uncovers the hidden semantics and respects the intrinsic geometric structure simultaneously. Graph regularized Nonnegative Matrix Factorization (GNMF) is one of the representative approaches in this category. The core of such approach is the graph, since a good graph can accurately reveal the relations of samples which benefits the data geometric structure depiction. In this paper, we leverage the sparse representation to construct a sparse hypergraph for better capturing the manifold structure of data, and then impose the sparse hypergraph as a regularization to the NMF framework to present a novel GNMF algorithm called Sparse Hypergraph regularized Nonnegative Matrix Factorization (SHNMF). Since the sparse hypergraph inherits the merits of both the sparse representation and the hypergraph model, SHNMF enjoys more robustness and can better exploit the high-order discriminant manifold information for data representation. We apply our work to address the image clustering issue for evaluation. The experimental results on five popular image databases show the promising performances of the proposed approach in comparison with the state-of-the-art NMF algorithms.

Keywords: Nonnegative Matrix Factorization, Image Representation, Hypergraph Learning, Image Clustering, Dimensionality Reduction, Sparse Representation

1. Introduction

Nonnegative Matrix Factorization (NMF) is a considerable multivariate analysis technique for data analysis\textsuperscript{24,12,15}. It learns a part-based representation of the data via finding two nonnegative matrices whose product provides a good approximation to the original matrix. In NMF, the nonnegative constraints lead to a parts-based representation because they allow only additive, not subtractive, combinations. And such way is in accordance with the recognition way of
the human brain\textsuperscript{12,3}. Due to its good psychological and physiological interpretation, NMF has been in vogue for decades and successfully applied to a wide range of domains such as computer vision, machine learning and pattern recognition\textsuperscript{3,28,16,14,20,30}.

Recently, many studies indicate that the high-dimensional data actually resides on a low-dimensional manifold and such intrinsic geometric structure of data is very useful for discriminating data\textsuperscript{21,2}. Over decades, extensive classical manifold learning approaches have been presented, such as Locally Linear Embedding (LLE)\textsuperscript{21}, Laplacian Eigenmapping (LE)\textsuperscript{2} and Neighborhood Preserving Embedding (NPE)\textsuperscript{8}. Graph learning is a popular technique for manifold learning. In graph learning, the manifold learning issue is considered as a graph cut issue\textsuperscript{25}. The core of graph learning approaches is graph which encodes the relations of samples. During the graph partition, such relations of samples should be preserved as more as possible\textsuperscript{2,9}. And then the manifold of data can be well kept.

In the last decade, some researchers are aware of the importance of manifold information and try to incorporate such desirable property into NMF\textsuperscript{3,3,28,7,34,27}. Graph regularized Nonnegative Matrix Factorization (GNMF) should be one of the representative approaches in this category\textsuperscript{3}. In GNMF, a graph is leveraged to depict the neighbourhood relation of data, and then introduced to NMF as a graph regularization for considering the local geometric structures preservation during the matrix factorization. The graph construction is very crucial for GNMF. Since the graph used in GNMF is the common graph whose edge only can connect two vertices, GNMF can only capture the simple pairwise relation of data and clearly neglect the high-order relation among the samples. To address this issue, Zeng et al presented the Hypergraph regularized Nonnegative Matrix Factorization (HNMF) for image clustering via using hypergraph instead of graph to depict the data structure\textsuperscript{33}. The reason why HNMF can address such problem is that hypergraph is a generalization of graph and its edge can connect any number of vertices\textsuperscript{35}. Generally speaking, a graph can only depict one relation (or measure) of samples. However, in most of time, it is very hard to pick up a suitable data relation to address different issues. Jim Jing-Yan Wang et al alleviated this problem by presenting Multiple Graph regularized Nonnegative Matrix Factorization (MultiGrNMF). In MultiGrNMF, multiple graphs, which encode different data relations, are combined as a combative graph regularization for constraining NMF. These graphs are deemed as a series of initial guesses of the optimal graph Laplacian. And, finally, the optimal combination of graphs can be conditionally learned by ensemble manifold regularization technique\textsuperscript{6}. The main drawback of MultiGrNMF over other GNMF algorithms is that it is very time consuming to construct multiple graphs. Moreover, from the perspective of graph embedding\textsuperscript{26}, many constrained NMF algorithms actually can be formulated as a specific GNMF algorithm\textsuperscript{13,29,22}.

In HNMF or GNMF algorithms, a good graph (or hypergraph) is an important key towards to the success, since a high-quality graph can well reveal the real relations among samples and these relations are the crucial cues for data analysis. In GNMF or graph learning approaches, the \textit{k}-nearest neighbour is the most commonly adopted method for graph (or hypergraph) construction. However, many studies have shown such graph construction fashion is sensitive to noise and often cannot correctly reflect the real relation of samples. Recently, motivated by the advancements of Sparse Representation (SR)\textsuperscript{11}, some researchers have attempted to leverage sparse representation for constructing
the high quality graph. The sparse representation-based graph is often called sparse graph or $L_1$-graph. Since it inherits the merits of SR which is very discriminative and robust to noise, the sparse graph works have already achieved remarkable success in many domains\textsuperscript{4,10}. We believe the success of sparse graph can be also applied to HMMF and GNMF. So, in this paper, we focus on applying sparse representation to construct a more robust and discriminative hypergraph for constraining NMF. We call such novel NMF algorithm Sparse Hypergraph regularized Nonnegative Matrix Factorization (SHNMF). Since the sparse representation-based hypergraph inherits the merits from both the hypergraph and the sparse representation, SHNMF enjoys more desirable properties and has a much better performance over the conventional GNMF and HNMF algorithms. We adopt five popular image databases to evaluate the data representational power of SHNMF. The experimental results demonstrate the superiority of SHNMF in comparison with the state-of-the-art NMF algorithms.

The rest of paper is organized as follows: the background knowledge and some basic notations are introduced in Section 2. Section 3 presents the methodology of our work; The experimental results are analyzed and discussed in section 4; the conclusion is finally summarized in section 5.

2. Preliminaries

Let us denote a set of $n$ samples as a $l \times n$-dimensional matrix $X = [x_1, \cdots, x_n]$ where $x_i$ is the $i$th sample which is corresponding to the $i$th column of matrix $X$. In the matrix factorization task, the sample matrix can be approximately factorized as $X \approx WU^T$ where the $l \times m$-dimensional matrix $W = [w_1, \cdots, w_m]$, $m \leq l$ is denoted as the bases (the basis vectors) while the $n \times m$-dimensional matrix $U = [u_1, \cdots, u_n]$ is denoted as its corresponding coefficients (loadings). For each sample, we have $x_i \approx Wu_i^T$. Clearly, such equation $X \approx WU^T$ represents the reconstruction process of sample matrix via using the bases and loadings. Its reverse process can be done as $u_i^T = W^{-1}x_i$.

2.1. Nonnegative Matrix Factorization (NMF)

Compared to the other matrix factorization techniques, Nonnegative Matrix Factorization (NMF) imposes the non-negativity constraints $W, U \geq 0$ to ensure that all entries of $W$ and $U$ are non-negative. Consequently, NMF only allows non-subtractive combinations. There are two cost functions can be defined to find an approximate factorization $X \approx WU^T$. The first one bases on the Euclidean distance and the second one bases on divergence. In this paper, we only introduce the Euclidean distance based version and the divergence based version can be referenced from their original papers. So, the NMF problem can be finally formulated as a following optimization problem:

$$\hat{W} = \arg\min_{W,U} \|X - WU^T\|^2, \quad s.t \quad W, U \geq 0$$

(1)

The above problem can be solved by using multiplicative updating rules\textsuperscript{24}. Furthermore, an additional constraint $\sum_i w_{ij} = 1$ is always imposed for stabilizing the computation, but this is not necessary.
2.2. Hypergraph

As a generalization of graph, hypergraph is an important tool for data representation. It depicts the structure of data via measuring the similarity between groups of points\textsuperscript{35}. The main difference between hypergraph and graph is that the edge of hypergraph can own any number of vertices while the one of graph can only connect two vertices. The edge of hypergraph is often called hyperedge. When the length of hyperedge is equal to two, the hypergraph is exactly equivalent to a common graph. Due to the aforementioned property of hyperedge, hypergraph enjoys a higher flexibility for depicting the high-order relation. In the real world, the relations among data should be more complex than the simple pairwise and naively squeezing such complex relations into pairwise ones will inevitably lead to loss of information which can be expected valuable for learning tasks. Take the article coauthor relationship as an example, an article is often corresponding to several authors. And, apparently, a common graph cannot intuitively describe such one to $n$ high-order relation. Even in GNMF algorithms, some works also empirically validate this argument. In the experiments of\textsuperscript{33}, Hypergraph regularized Nonnegative Matrix Factorization (HNMF) which utilizes hypergraph instead graph for considering the high-order relations among samples achieve a better performance in comparison with GNMF\textsuperscript{3}.

Here, we introduce some basic concepts and notations of hypergraph. Let $G(V, E)$ denote a hypergraph with vertex set $V = \{v_1, \ldots, v_n\}$ and hyperedge set $E = \{e_1, \ldots, e_s\}$ for depicting the sample matrix $X = [x_1, \ldots, x_n]$. In such hypergraph, each vertex is corresponding to a sample and each hyperedge is an arbitrary subset of $V$ which groups the samples by a relation. Generally speaking, a hyperedge is often associated with a weight $w(\cdot)$. Such weight essentially indicates the degree of necessity for preserving this hyperedge (keeping the corresponding sample relation) during the hypergraph partition. Given a vertex $v$, we denote its degree $d(v)$ as $d(v) = \sum_{e \ni v} w(e)$, for measuring the degree of importance of the vertex $v$ to the hypergraph $G$. Similarly, The degree of a hyperedge $e$ is denoted as $\delta(e) = |e|$ where $|\cdot|$ returns the number of elements. The degree of hyperedge can be deemed as the length of hyperedge. For $k$-uniform hypergraphs, the degrees of each hyperedge are equal, $\delta(e) = k$. In particular, for the case of ordinary graphs or 2-graphs, $\delta(e) = 2$. The vertex-edge incidence matrix $H$ is a $|V| \times |E|$ dimensional binary matrix, which is a common way to depict a hypergraph. Here we denote its $v, e$-th entry as $h(v, e)$. If $v \in e$, $h(v, e)$ is 1, otherwise it is 0. By these definitions, we have $d(v) = \sum_{e \ni v} w(e) h(v, e)$ and $\delta(e) = \sum_{v \in e} h(v, e)$.

After obtaining the hypergraph which encodes the relations among the samples, the data analysis task can be accomplished by grouping the samples based on their relations, which is also known as a hypergraph cut task from the perspective of graph learning. According to Zhou’s normalized hypergraph cut\textsuperscript{35}, we have the hypergraph structure loss function over a collection of hypergraph cuts denoted by a $n \times m$-dimensional matrix $F = [f_1, \ldots, f_m]$,

$$
\hat{F} = \arg \min_{F} \Omega(F, G) := \frac{1}{2} \sum_{e \in E} \sum_{(u, v) \in e} \frac{w_e}{\delta(e)} \left\| \frac{F(u)}{\sqrt{d(u)}} - \frac{F(v)}{\sqrt{d(v)}} \right\|^2,
$$

(2)

where a hypergraph cut $f_i$ is a $n$-dimensional vector and each of its element reflects the probability of a sample (a vertex) belonging to a partitioned subgraph. $F(\cdot)$ returns a $m$-dimensional vector whose elements are the ones of different hypergraph cuts corresponding to the given vertex. Equation 2 aims at learning a collection of hypergraph cuts
which can preserve the hyperedges as more as possible during the hypergraph partition. In addition, the hyperedges encode the relations between samples and the obtained subgraphs are actually the clusters of data. Therefore, the data geometric structure can be preserved in the learned hypergraph cut space whose dimensionality is much lower than the one of the sample space.

After several deductions, Equation 2 can be eventually formulated as a matrix form:

\[ \Omega(F, G) = \text{Trace}[F^T (I - D_v^{-1/2} H W D_v^{-1} H^T D_v^{-1/2}) F] \]
\[ = \text{Trace}[F^T (I - A) F] = \text{Trace}(F^T L F), \quad (3) \]

where \( \text{Trace}(\cdot) \) returns the trace of a matrix and the matrix \( L = I - A = I - D_v^{-1/2} H W D_v^{-1} H^T D_v^{-1/2} \) is the Laplacian matrix of hypergraph \( G \). \( A = D_v^{1/2} H W D_v^{-1} H^T D_v^{1/2} \) can be deemed as a normalized affinity matrix encoded the structure of hypergraph \( G \). \( D_v, D, \) and \( W \) are the diagonal matrix forms of \( \delta(e), d(v) \) and \( w_i \) respectively.

2.3. Sparse Representation

Sparse Representation (SR) is a constrained linear regression model which has achieved extensive attentions in dozens of areas. SR imposes a \( L_1 \)-norm regularization as the sparsity constraint to the linear regression model to learn the \( L_1 \)-norm regularized regression coefficients. Such sparsity constraint endows SR the strong discriminating capability and the robustness to noise, since only the most relevant train samples are compulsively selected for reconstructing the query sample. Given a query sample \( x_i \), we can use the rest of samples \( X_{\neq i} = [x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n] \) to represent it via using sparse representation as follows:

\[ \hat{C}_i = \arg \min_{C_i} ||C_i||_1, \text{ s.t. } ||x_i - X_{\neq i} C_i^T||^2 \leq \epsilon \]
\[ \Rightarrow \hat{C}_i = \arg \min_{C_i} (1 - \beta) ||x_i - X_{\neq i} C_i^T||^2 + \beta ||C_i||_1 \quad (4) \]

where \( \epsilon \) is the measurement noise and \( \beta \) is a parameter in the range \([0, 1]\) which is used to control the trade off between the reconstruction error and the sparsity. The \((n - 1)\)-dimensional vector \( C_i = [c_{i1}, \cdots, c_{(i-1)}, c_{(i+1)}, \cdots, c_m] \) is the regression coefficients with respect to the query sample \( x_i \) where its element \( c_{ij} \) encodes the correlation between \( x_i \) and \( x_j \).

3. Methodology

3.1. Sparse Hypergraph

In this section, we introduce how to leverage the sparse representation algorithm to construct a high quality hypergraph for better representing the data structure. The regression coefficients learned by SR are actually the correlations of samples and such correlation is natural similarity measure.

For each pair of samples, we can obtain two correlations via using Equation 4. Then, we define the similarity of these two samples as the mean of the absolute values of their correlations to make sure the nonnegativity and symmetry of their similarity. Mathematically speaking, the similarity between \( x_i \) and \( x_j \) can be denoted as follows:

\[ s_{ij} = s_{ji} = \frac{|c_{ij}| + |c_{ji}|}{2}. \quad (5) \]
Since the self-similarity cannot be computed by SR, we define the self-similarity of a sample as the sum of the similarities between this sample and the rest samples, $s_i = \sum_{t \neq i} s_{it}$. According to the obtained similarities among all the samples, it is not hard to construct a similarity matrix (or affinity matrix) $S$ where $s_{ij}$ is the $ij$-th element of $S$. Then, the normalization of the sample similarities can be done by $S = \sqrt{MS} \sqrt{M}$ where $n \times n$-dimensional matrix $M$ is diagonal matrix whose $i$-th diagonal element is the sum of elements in the $i$-th row of $S$, $m_i = \sum_s s_{is}$.

After obtaining the similarities, we can follow Section 2.2 to define a hypergraph $G_S$ for depicting the samples where its affinity matrix $A_S$ is exactly the aforementioned normalized $S$. In this sparse hypergraph, we can consider each sample as a centroid and select its top $k - 1$ most similar samples to define a $k$-length hyperedge. Therefore, for a data collection constructed by $n$ samples, we can obtain $n$ hyperedges. The weight of the hyperedge $e$ is defined as the mean of the similarities between each two samples in the hyperedge,

$$w = \frac{\sum_{\{va, vb\} \in e} s_{ab}}{l}$$

(6)

where $l$ is the number of pairs of vertices in the hyperedge $e$. Since this hypergraph is constructed based on sparse representation, we call it sparse hypergraph.

3.2. Sparse Hypergraph regularized Nonnegative Matrix Factorization (SHNMF)

We aim at learning a collection of nonnegative bases which can not only minimize the reconstruction error but also preserve the relations among samples encoded in sparse hypergraph. From the perspective of graph embedding, the loading matrix of these bases actually can be deemed as the hypergraph cut matrix of sparse hypergraph, $U = F$, where $F$ is the hypergraph cut matrix of sparse hypergraph $G_S$. Therefore, the optimal nonnegative bases $\hat{W}$ can be achieved via minimizing the following objective,

$$\hat{W} = \arg\min_{W,U} \|X - WU^T\|^2 + \lambda \Omega(F, G_S)$$

(7)

$$= \|X - WU^T\|^2 + \lambda \text{Trace}(U^T L_S U)$$

where $L_S = I - A_S$ is the Laplacian matrix of the sparse hypergraph $G_S$ and $A_S$ is the affinity matrix encoded sparse hypergraph structure (The detailed computation procedure of $A_S$ has been introduced in Section 3.1). Since this work is based on the sparse hypergraph, we call it Sparse Hypergraph Nonnegative Matrix Factorization (SHNMF). The objective function of SHNMF can be further deduced as follows:
\[ \mathcal{F} = \|X - WU^T\|^2 + \lambda \text{Trace}\{U^T(I - A_S)U\} \]
\[ = \text{Trace}(X - WU^T)(X - WU^T)^T + \lambda \text{Trace}\{U^T(I - A_S)U\} \]
\[ = \text{Trace}(XX^T) - 2\text{Trace}(XUW^T) + \text{Trace}(WU^TUW^T) + \lambda \text{Trace}(U^TU) - \lambda \text{Trace}(U^T A_S U) \] (8)

3.3. Updating Rules Minimizing Objective (8)

Let \( \theta \) and \( \phi \) be the Lagrangian multipliers for constraints \( W_{ij} \geq 0 \) and \( U_{ij} \geq 0 \) respectively. So, the Lagrangian of SHNMF \( \mathcal{J} \) is denoted as follows:
\[ \mathcal{J} = \mathcal{F} + \text{Tr}(\Theta W^T) + \text{Tr}(\Phi U^T) \] (9)
where \( \Theta = [\theta_{ij}] \) and \( \Phi = [\phi_{ij}] \). The derivatives of this Lagrangian with respect to \( W \) and \( U \) are:
\[ \frac{\partial \mathcal{J}}{\partial W} = 2WU^TU - 2XU + \Theta, \] (10)
\[ \frac{\partial \mathcal{J}}{\partial U} = 2UW^TW - 2X^TW + 2\lambda U - 2\lambda A_S U + \Phi. \] (11)

We can get the following equations via using Karush–Kuhn–Tucker conditions, \( \theta_{ij}w_{ij} = 0 \) and \( \phi_{ij}h_{ij} = 0 \):
\[ (WU^TU)_{ij}w_{ij} - (XU)_{ij} = 0, \] (12)
\[ (X^TW + \lambda A_S U)_{ij}h_{ij} - (UW^TW + \lambda U)_{ij} = 0. \] (13)

Therefore, the multiplicative updating rules of SHNMF with respect to \( W \) and \( U \) are
\[ w_{ik} \leftarrow \frac{(XU)_{ik}}{(WU^TU)_{ik}}, \] (14)
\[ u_{jk} \leftarrow \frac{(X^TW + \lambda A_S U)_{jk}}{(UW^TW + \lambda U)_{jk}}. \] (15)

Then, we can iteratively and alternatively learn \( U \) and \( W \) via giving the initial guesses of them. The learned \( H \) is the nonnegative representation of data and the learned \( W \) is the bases of SHNMF subspace.

3.4. Convergence Proof of Updating Rules

According to \(^3\), the multiplicative updating rules in Equations 14 and 15 are exactly special cases of gradient descent with an automatic step parameter selection. The advantage of multiplicative updating rules is the guarantee of nonnegativity of \( W \) and \( U \). In this section, we will prove the multiplicative updating rules in Equations 14 and 15 converge to a local optimum. Before presenting the proofs, a definition and a lemma with its corresponding proof given by Lee and Seung \(^24\) should be introduced first.
Definition 1. \( G(u, u') \) is an auxiliary function for \( P(u) \) if the conditions
\[
G(u, u') \geq P(u), \quad G(u, u) = P(u)
\]
be satisfied.

Lemma 1. If \( G \) is an auxiliary function, then \( P \) is non increasing under the update
\[
u^{t+1} = \arg\min_u G(u, u')
\]
Proof. \( P(u^{t+1}) \leq G(u^{t+1}, u') \leq G(u', u') = P(u') \)

Definition 1 and Lemma 1 are important tools for proving the convergence of SHNMF algorithm. Now, the proof of the convergence of SHNMF can be translated into the proof of the following theorem.

Theorem 1. The objective function in Equation 8 is no increasing with the updating rules in Equations 14 and 15. The objective function is invariant under these updates if and only if \( W \) and \( U \) are at a stationary point.

Now, we will show that the updating step for \( U \) in Equation 15 is exactly the update in Equation 17 with a proper auxiliary function. The objective function of SHNMF can be rewritten as follows:
\[
F = ||X - WU^T||^2 + \lambda \text{Trace}(U^T L_S U)
\]
\[
= \sum_{i=1}^{l} \sum_{j=1}^{n} (x_{ij} - \sum_{k=1}^{m} w_{ik} u_{jk})^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{n} u_{ik} (L_S)_{ij} u_{jk}
\]
The corresponding first-order and second-order derivatives of the objective function of SHNMF with respect to \( u_{ij} \) are as follows:
\[
F'_{ij} = 2(UW^T W + \lambda L_S U - X^T W)_{ij}
\]
\[
F''_{ij} = 2(W^T W)_{ii} + 2\lambda (L_S)_{ii}
\]
Since the update is element wise, so it is sufficient to show that each \( F_{ij} \) is no increasing under update of Equation 15.

Lemma 2. Function
\[
G(u, u'_j) = F_{ij}(u'_j) + F'_{ij}(u'_j)(u - u'_j) + \frac{(UW^T W)_{ij} + \lambda (U)_{ij}}{u_{ij}}(u - u'_j)^2
\]
is an auxiliary function for \( F_{ij} \), the part of \( F \), which is only relevant to \( u_{ij} \).

Proof. Since \( G(u, u) = F_{ij}(u) \) is obvious, we need only show that \( G(u, u'_j) \geq F_{ij}(u) \). To do this, we compare the Taylor series expansion of \( F_{ij}(u) \),
\[
G(u, u'_j) = F_{ij}(u'_j) + F'_{ij}(u'_j)(u - u'_j) + [(W^T W)_{ii} + \lambda (L_S)_{ij}](u - u'_j)^2,
\]
With Equation 21 to find that \( G(u, u'_ij) \geq F_{ij}(u) \) is equivalent to
\[
\frac{(UW^TW)_{ij} + \lambda(U)_{ij}}{u'_{ij}} \geq (W^TW)_{ii} + \lambda(L_S)_{jj}.
\] (22)

We have \((UW^TW)_{ij} = \sum_{m=1}^n u'_m(W^TW)_{mj} \geq u'_{ij}(W^TW)_{jj}\) and \(\lambda U_{ij} = \lambda \sum_{b=1}^l I_{ib}u'_{bj} \geq \lambda(I - A_S)_{ij}u'_{ij} = \lambda(L_S)_{jj}u'_{ij}\).

Thus, we have \(G(u, u'_ij) \geq F_{ij}(u)\) and Lemma 2 holds.

Currently, we can lead to the proofs of Theorem 1.

**Proof of Theorem 1.** According to Lemmas 1 and 2, we can replace Equation 17 by Equation 21 to get the following updating rule for \(U\)
\[
u_{ij}^{t+1} = u'_{ij} - u'_{ij} \frac{F'_{ij}(v'_{ij})}{2(UW^TW)_{ij} + 2\lambda U_{ij}} = u'_{ij} \frac{(X^TW + \lambda A_S U)_{ij}}{(UW^TW + \lambda U)_{ij}}.
\] (23)

Since Equation 21 is an auxiliary function, \(F_{ij}\) is nonincreasing under this updating rule.

With regard to the convergence proof of the updating rule for \(W\) in Equation 14, its proof can directly refer to the proof of Theorem 1 in the paper of Lee and Seung\(^{12}\), since the updating rule for \(W\) is exactly as same as the updating rule for \(W\) in the NMF algorithm.

3.5. Computational Complexity Analysis

The lasso is one of the most common ways to solve the sparse representation problem. Take it as a case, the computational complexity of the sparse hypergraph construction step is \(O(nl^3 + n^2l^2)\) where \(n\) is the number of samples and \(l\) is the dimension of samples. However, there also exist lots of other more efficient approaches to address such sparse representation issue. So the computational complexity of this step can be further reduced. Supposing the multiplicative updates stops after \(t\) iterations, the overall cost for NMF (both formulations) is \(O(tnm)\) where \(m\) is the number of factors (the number of bases of SHNMF). Therefore, the total overall cost of SHNMF is \(O(tnm + nl^3 + n^2l^2)\).

4. Experimental Results

In this section, we conduct several experiments for validating our work on five popular image databases, namely JAFFE\(^5\), COIL20\(^{19}\), ORL\(^{23}\), AR\(^17\) and Yale\(^1\). Several state-of-the-art NMF algorithms are used for comparison. They are Nonnegative Matrix Factorization (NMF)\(^{12}\), Graph regularized Nonnegative Matrix Factorization (GNMF)\(^3\), Orthogonal Nonnegative Matrix Factorization (ONMF)\(^{14}\), Local Nonnegative Matrix Factorization (LNMF)\(^{13}\) and Hypergraph regularized Nonnegative Matrix Factorization (HNMF)\(^{33}\).

The JAFFE face database contains 213 images of 7 facial expressions posed by 10 Japanese females\(^5\). The size of each image is 64 \times 64 pixels in our experiments. The COIL-20 database has 20 objects and each object is represented in the database by 72 images obtained by the rotation of the object through 360\(^\circ\) in 5\(^\circ\) steps (1440 images in total)\(^{19}\). The size of each image is 32\times32 pixels in our experiments. The ORL database contains 400 face images from 40 subjects\(^{23}\). Each subject has ten images acquired at different times. In this database, the subjects’ facial expressions
and facial details are varying. The size of the face image is $32 \times 32$ pixels. Compared to the other three databases, this database is much smaller and we use it to learn the optimal parameters. The AR database consists of more than 4,000 face images of 126 subjects. The database characterizes divergence from ideal conditions by incorporating various facial expressions, luminance alterations, and occlusion modes. Following paper, a subset contains 1,680 images with 120 subjects are constructed in our experiment. All these images are $50 \times 40$ pixels. The Yale face database has 165 grayscale images of 15 individuals. Every subject involves different facial expressions and configurations. And the size of each image is $32 \times 32$ pixels.

### 4.1. Experimental Settings

The grayscale are directly used as the features on all five image databases. Among the compared approaches, GNMF supports several graph settings. We adopt two commonly used ones which are named Binary Graph regularized Nonnegative Matrix Factorization (GNMF$_B$ in short) and Cosine distance-based Graph regularized Nonnegative Matrix Factorization (GNMF$_C$ in short). In NMF algorithms, the learned loading matrix $U$ is considered as the new representation of data. We follow the experimental settings in which evaluate the representational power of NMF algorithms via clustering. In the experiments, the Clustering Accuracy and Normalized Mutual Information (NMI) as leveraged as two evaluation metrics. After representing the data using NMF algorithms, $k$-means is leveraged for performing the clustering and the cluster number for each database is fixed to its category number. With regard to the parameters of the compared approaches, they are all well tuned for presenting a fair comparison.

### 4.2. Image Clustering

We apply SHNMF and the compared approaches to address the image clustering issue for evaluating their data representational powers. Table 1 reports the clustering performances of different approaches on five image databases. From the observations, SHNMF outperforms the compared approaches under both two of the evaluation metrics. For examples, the clustering accuracy gains of SHNMF over HNMF on AR, ORL, COIL20, JAFFE and Yale databases are 33.71%, 9.50%, 4.10%, 2.23%, and 1.21% respectively. AR, COIL20 and ORL databases are relatively larger.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Clustering Accuracy</th>
<th>Normalized Mutual Information (NMI)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AR</td>
<td>ORL</td>
</tr>
<tr>
<td>NMF$^{24}$</td>
<td>24.29</td>
<td>51.25</td>
</tr>
<tr>
<td>LNMF$^{13}$</td>
<td>26.79</td>
<td>36.25</td>
</tr>
<tr>
<td>ONMF$^{14}$</td>
<td>26.29</td>
<td>52.00</td>
</tr>
<tr>
<td>GNMF$^C$</td>
<td>24.50</td>
<td>54.75</td>
</tr>
<tr>
<td>GNMF$_B$</td>
<td>28.86</td>
<td>58.25</td>
</tr>
<tr>
<td>HNMF$^{33}$</td>
<td>33.86</td>
<td>61.25</td>
</tr>
<tr>
<td>SHNMF</td>
<td><strong>67.57</strong></td>
<td><strong>70.75</strong></td>
</tr>
</tbody>
</table>
image databases. So the experimental results actually show a good property of SHNMF that SHNMF performs much better on a larger database. Comprehensively speaking, GNMF often performs much better than the conventional NMF algorithms while HNMF performs much better than GNMF. Clearly, these phenomena validate the facts that the manifold (or geometric) information can benefit the data representation and hypergraph enjoys the advantages in manifold structure depiction over graph. Such phenomena also reveal the reason why SHNMF defeats HNMF with a huge advantage is the excellent data manifold structure depiction capacity of sparse hypergraph over the common hypergraph and graph.

Figure 1: The impacts of $\beta$ and $\lambda$ to the clustering performances of SHNMF.

4.3. Parameters Discussion

The SHNMF algorithm involves three parameters, $\beta$, $\lambda$ and $k$. The parameter $\beta$ in Equation 4 is used to control the sparsity of sparse representation for better measuring the correlations of samples. The parameter $\lambda$ in Equation 7 is used to reconcile the reconstruction error and manifold information preservation. The parameter $k$ is the hyperedge length in the sparse hypegraph construction step. We conduct several experiments to empirically study the impacts of these three parameters to the performance of SHNMF. Figure 1 plots the clustering performance of SHNMF (the average of the clustering accuracy and NMI) under different values of $\lambda$ and $\beta$ on five image databases. From the observations in Figure 1, it is not hard to find that a greater $\lambda$ can benefit the performance of SHNMF more. Clearly, this phenomenon emphasizes the importance of the manifold structure in the data representation. Different to the $\lambda$, the observations in Figure 1 show that the optimal values of $\beta$ for different databases are different. We attribute this to the reasons that $\beta$ is used to control the sparsity of regression which is highly related to the dimension and size of data while these factors of involved image databases are totally different. The optimal values of $\beta$ for JAFFE, COIL20, AR, ORL and Yale databases are $10^{-3}$, $10^{-1}$, $10^{-4}$, $10^{-5}$ and $10^{-6}$ respectively. The hyperedge length $k$ is an important parameter which can deeply influence the quality of sparse hypergraph. However it has numerous possible values and it is impossible for us to evaluate each possible value. So, in our experiments, we follow the hypergraph parameter study fashion in\textsuperscript{32} which defines a hyperedge length candidate collection for each database first and then conducts experiments to empirically pick up the optimal hyperedge length from the candidates. Table 2 lists the hyperedge length candidate collections of SHNMF and HNMF respectively. And it also reports their optimal hyperedge lengths.
Table 2: The optimal hyperedge length $k$ of SHNMF and HNMF on different databases. (Here, the hyperedge length is excluding the centroid itself, $k_{HNMF}$ is the optimal $k$ of HNMF and $k_{SHNMF}$ is the optimal $k$ of SHNMF).

<table>
<thead>
<tr>
<th>Database</th>
<th>The hyperedge length candidates</th>
<th>$k_{HNMF}$</th>
<th>$k_{SHNMF}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAFFE</td>
<td>[2, 4, 6, 8, 10, 12, 14, 16, 18]</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>COIL20</td>
<td>[3, 5, 10, 20, 30, 36, 54, 72]</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>AR</td>
<td>[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>ORL</td>
<td>[2, 3, 4, 5, 6, 7, 8, 9, 10]</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Yale</td>
<td>[2, 3, 4, 5, 6, 7, 8, 9, 10]</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

for different databases. From the observations, the optimal $k$ of SHNMF is more stable. In the most of time, the optimal $k$ is equal to 4.

5. Conclusion

In this paper, we presented a new GNMF algorithm called Sparse Hypergraph regularized Nonnegative Matrix Factorization (SHNMF). In SHNMF, the sparse representation is leveraged to construct a sparse hypergraph for regularizing the NMF algorithm. Since the sparse hypergraph inherits the merits of both the sparse representation and the hypergraph learning, SHNMF is more robust and can better capture the high-order discriminative manifold information for data representation. We empirically evaluated SHNMF on five popular image databases, and the experimental results demonstrate the superiority of SHNMF over the compared NMF algorithms. Clearly, there still exist a lot of interesting works can further done based on our proposed work. For example, we can follow the idea of MultiGrNMF to construct multiple hypergraphs which encode different feature information and metric information for regularizing the NMF algorithm.

Acknowledgement

The work described in this paper was partially supported by National Natural Science Foundations of China (NO.61173131), Fundamental Research Funds for the Central Universities (No. 106112015CDJRC091101).

References


