Non-Negative Matrix Factorization with Constraints

Haifeng Liu Zhaohui Wu
College of Computer Science, Zhejiang University
Hangzhou, Zhejiang, China
{haifengliu,wzh}@zju.edu.cn

Abstract

Non-negative matrix factorization (NMF), as a useful decomposition method for multivariate data, has been widely used in pattern recognition, information retrieval and computer vision. NMF is an effective algorithm to find the latent structure of the data and leads to a parts-based representation. However, NMF is essentially an unsupervised method and can not make use of label information. In this paper, we propose a novel semi-supervised matrix decomposition method, called *Constrained Non-negative Matrix Factorization*, which takes the label information as additional constraints. Specifically, we require that the data points sharing the same label have the same coordinate in the new representation space. This way, the learned representations can have more discriminating power. We demonstrate the effectiveness of this novel algorithm through a set of evaluations on real world applications.

Introduction

Dimensionality reduction techniques have been receiving more and more attentions as fundamental tools for data representation (Lee and Seung 1999; He, Cai, and Min 2005; Min, Lu, and He 2004; He et al. 2005). Among them, matrix decomposition approaches have been developed by using different criteria. The most popular techniques include Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and Vector Quantization. Central to the matrix factorization is to find two or more matrix factors whose product is a good approximation to the original matrix. In real applications, the dimension of the decomposed matrix factors is usually much smaller than that of the original matrix. This gives rise to compact representations of the data points which can facilitate other learning tasks such as clustering and classification.

Among matrix factorization methods, Non-negative Matrix Factorization (NMF)(Lee and Seung 1999; Li and Ding 2006) specializes in that it enforces the constraint that the factor matrices must be non-negative, i.e., all elements must be equal to or greater than zero. This non-negative constraint leads NMF to a parts-based representation of the object in the sense that it only allows additive, not subtractive combination of the components. Therefore, it is an ideal dimensionality reduction algorithm for image processing, face recognition (Lee and Seung 1999; Li et al. 2001) and document clustering (Xu, Liu, and Gong 2003), where it is natural to consider the object as a combination of parts to form a whole. As in the scope of non-negative matrix factorization, the related work includes pLSA (Hofmann 2001), co-clustering (Dhillon, Mallela, and Modha 2003) etc.

NMF is an unsupervised learning algorithm. That is, NMF is inapplicable to many real-world problems where limited knowledge from domain experts is available. However, many machine learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy (Chapelle, Schölkopf, and Zien 2006; He 2010). The cost associated with the labeling process may render a fully labeled training set infeasible, whereas acquisition of a small set of labeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Therefore, It would be great benefit to extend the usage of NMF to a semi-supervised manner.

Recently, Cai et al. (Cai et al. 2008; 2009) proposed a Graph regularized NMF (GNMF) approach to encode the geometrical information of the data space. GNMF constructs a nearest neighbor graph to model the local manifold structure. When label information is available, it can be naturally incorporated into the graph structure. Specifically, if two data points share the same label, a large weight can be assigned to the edge connecting them. If two data points have the different labels, the corresponding weight is set to be 0. This gives rise to semi-supervised GNMF. The major disadvantage of this approach is that there is no theoretical guarantee that data points from the same class will be mapped together in the new representation space, and it remains unclear how to select the weights in a principled manner.

In this paper, we propose a novel matrix decomposition method, called *Constrained Non-negative Matrix Factorization* (CNMF), which takes the label information as additional *hard* constraints. The central idea of our approach is that the data points from the same class should be merged together in the new representation space. Thus, the obtained parts-based representation has the consistent label with the original data, and therefore can have more discriminating power. Another advantage of our approach is that it is parameter free, which avoids the cost of tuning parameters in order to get the best result. It makes our algorithm applicable to many real world applications easily and efficiently. We also discuss how to solve the corresponding optimization problem efficiently.

A Review of NMF

Non-negative Matrix Factorization (NMF) (Lee and Seung 2001) is an unsupervised learning algorithm used to decompose multivariate data under the constraint that all entries in the decomposed matrix factors have to be non-negative.

Suppose we have n data points $\{\mathbf{x}_i\}_{i=1}^n$. Each data point $\mathbf{x}_i \in \mathbb{R}^m$ is m-dimensional and is represented by a vector. The vectors are placed in the columns and the whole data set is represented by a matrix $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m \times n}$. NMF aims to find two non-negative matrix factors $\mathbf{U}$ and $\mathbf{V}$ where the product of the two factors is an approximation of the original matrix, represented as:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T \tag{1}$$

The approximation is quantified by a cost function which can be constructed by some distance measure. If we take the *Frobenius norm* as an example, the goal of NMF can be restated as follows: to factor $\mathbf{X}$ into an $m \times k$ matrix $\mathbf{U}$ and a $k \times n$ matrix $\mathbf{V}^T$ such that the following objective function is minimized:

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

This objective function is not convex in both variables $\mathbf{U}$ and $\mathbf{V}$. Thus, it is hard to find the global minima for $\mathcal{O}$. Lee and Seung proposed an iterative update algorithm (Lee and Seung 2001) to find the locally optimal solution for the above optimization problem.

In the NMF factorization, each column vector of $\mathbf{U}$, $\mathbf{u}_i$, can be regarded as a basis and each data point $\mathbf{x}_i$ is approximated by a linear combination of these k bases, weighted by the components of $\mathbf{V}$. In other words, NMF maps each data $\mathbf{x}_i$ to $\mathbf{v}_i$ from m-dimensional space to k-dimensional space. The new representation space is spanned by the k bases $\mathbf{u}_i$. In real applications such as image processing (Lee and Seung 1999), face recognition (Li et al. 2001) (Liu, Zheng, and Lu 2003) and document clustering (Xu, Liu, and Gong 2003), we usually set $k \ll m$ and $k \ll n$. Then the high dimensional data can be represented by a set of low-dimensional vectors in the hope that the basis vectors can discover the latent semantic structure among the data set. Different from other dimension reduction algorithms such as PCA, LDA and LPP (He and Niyogi 2003), the non-negative constraints on $\mathbf{U}$ and $\mathbf{V}$ only permit the additive combination of the basis vectors, which is the reason why NMF is considered as the parts-based representation.

NMF with Constraints

NMF is an unsupervised learning algorithm. It can not be applied directly to the situation when the label information is available. In this section, we introduce a novel matrix decomposition method, called *Constrained Non-negative Matrix Factorization* (CNMF), which takes the label information as additional constraints. This method can guarantee that the data points sharing the same label can be mapped into the same class in the low-dimensional space. The algorithm presented in this paper is fundamentally motivated from semi-supervised graph embedding (He, Ji, and Bao 2009) which also consider label information as additional constraints.

The Objective Function

Consider a data set consisting of n data points $\{\mathbf{x}_i\}_{i=1}^n$, among which the label information is available for the first l data points $\mathbf{x}_1, \cdots, \mathbf{x}_l$, and the rest $n - l$ data points $\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n$ are unlabeled.

Suppose there are c classes. Each data point from $\mathbf{x}_1, \cdots, \mathbf{x}_l$ is labeled with one class. We first build an $l \times c$ indicator matrix $\mathbf{C}$ where $c_{i,j} = 1$ if $\mathbf{x}_i$ is labeled with the j-th class; $c_{i,j} = 0$ otherwise. With the indicator matrix $\mathbf{C}$, we define a label constraint matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \begin{pmatrix} \mathbf{C}_{l \times c} & 0 \\ 0 & \mathbf{I}_{n-l} \end{pmatrix}$$

where $\mathbf{I}_{n-l}$ is a $(n-l) \times (n-l)$ identity matrix. Recall that NMF maps each data point $\mathbf{x}_i$ to $\mathbf{v}_i$ from m-dimensional space to k-dimensional space. To incorporate the label information, we can impose the label constraints by introducing an auxiliary matrix $\mathbf{Z}$:

$$\mathbf{V} = \mathbf{A}\mathbf{Z} \tag{3}$$

From the above equation, it is easy to check that if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same label, then $\mathbf{v}_i = \mathbf{v}_j$.

With the label constraints, our CNMF algorithm reduces to minimize the following objective function

$$\begin{aligned} \mathcal{O} &= \|\mathbf{X} - \mathbf{U}\mathbf{V}^T\| \\ &= \|\mathbf{X} - \mathbf{U}(\mathbf{A}\mathbf{Z})^T\| \\ &= \|\mathbf{X} - \mathbf{U}\mathbf{Z}^T\mathbf{A}^T\| \end{aligned} \tag{4}$$

with the constraint that $u_{i,j}$ and $z_{i,j}$ are non-negative. Since $\mathbf{Z}$ is non-negative, it is easy to see that $\mathbf{V}$ is also non-negative.

The Algorithm

The objective function of CNMF in Eq. (4) is not convex in both variables $\mathbf{U}$ and $\mathbf{Z}$. It is, thus, unrealistic to find the global minima for $\mathcal{O}$. Int the following, we describe an iterative updating algorithm to obtain the local optima of $\mathcal{O}$.

Using the matrix property Tr$(\mathbf{AB})$ = Tr$(\mathbf{BA})$, the objective function $\mathcal{O}$ can be rewritten as:

$$\begin{aligned} \mathcal{O} &= \text{Tr}((\mathbf{X} - \mathbf{U}\mathbf{Z}^T\mathbf{A}^T)(\mathbf{X} - \mathbf{U}\mathbf{Z}^T\mathbf{A}^T)^T) \\ &= \text{Tr}((\mathbf{X}\mathbf{X}^T - 2\mathbf{X}\mathbf{A}\mathbf{Z}\mathbf{U}^T + \mathbf{U}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{U}^T)) \\ &= \text{Tr}(\mathbf{X}\mathbf{X}^T) - 2\text{Tr}(\mathbf{X}\mathbf{A}\mathbf{Z}\mathbf{U}^T) + \text{Tr}(\mathbf{U}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{U}^T) \end{aligned}$$

Let α_{ij} and β_{ij} be the Lagrange multiplier for constraint $u_{ij} \geq 0$ and $z_{ij} \geq 0$, respectively, and $\boldsymbol{\alpha} = [\alpha_{ij}]$, $\boldsymbol{\beta} = [\beta_{ij}]$. The Lagrange $\mathcal{L}$ is

$$\mathcal{L} = \mathcal{O} + \text{Tr}(\boldsymbol{\alpha}U^T) + \text{Tr}(\boldsymbol{\beta}Z^T) \tag{5}$$

Requiring that the derivatives of $\mathcal{L}$ with respect to $\mathbf{U}$ and $\mathbf{Z}$ vanish, we have:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = -2\mathbf{XAZ} + 2\mathbf{UZ}^T\mathbf{A}^T\mathbf{AZ} + \alpha = 0 \quad (6)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = -2\mathbf{A}^T\mathbf{X}^T\mathbf{U} + 2\mathbf{A}^T\mathbf{AZU}^T\mathbf{U} + \beta = 0 \quad (7)$$

Using the Kuhn-Tucker condition $\alpha_{ij}u_{ij} = 0$ and $\beta_{ij}z_{ij} = 0$, we get the following equations for u_{ij} and z_{ij}:

$$(\mathbf{XAZ})_{ij}u_{ij} - (\mathbf{UZ}^T\mathbf{A}^T\mathbf{AZ})_{ij}u_{ij} = 0 \quad (8)$$

$$(\mathbf{A}^T\mathbf{X}^T\mathbf{U})_{ij}z_{ij} - (\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ij}z_{ij} = 0 \quad (9)$$

These equations lead to the following updating rules:

$$u_{ij} \leftarrow u_{ij}\frac{(\mathbf{XAZ})_{ij}}{(\mathbf{UZ}^T\mathbf{A}^T\mathbf{AZ})_{ij}} \quad (10)$$

$$z_{ij} \leftarrow z_{ij}\frac{(\mathbf{A}^T\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ij}} \quad (11)$$

We have the following theorem regarding the above iterative updating rules.

Theorem 1. *The objective function $\mathcal{O}$ is nonincreasing under the update rules in Eq.* (10) *and* (11)*. The objective function is invariant under these updates if and only if* $\mathbf{U}$ *and* $\mathbf{Z}$ *are at a stationary point.*

Theorem 1 grantees the convergence of the iterations in Eq. (10) and (11) and therefore the final solution will be a local optima. In the following, we will give the proof of Theorem 1.

To prove Theorem 1, we use a similar auxiliary function as used in the Expectation-Maximization algorithm (Dempster, Laird, and Rubin 1977; Saul and Pereira 1977).

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

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

are satisfied.

Regarding the above auxiliary function, we have the following lemma, which will be used to prove the convergence of the objective function.

Lemma 2. *If G is an auxiliary function, then F is nonincreasing under the update*

$$x^{t+1} = \arg\min_x G(x, x'). \quad (12)$$

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

The equality $F(x^{t+1}) = F(x^t)$ holds only if x^t is a local minimum of $G(x, x^t)$. By iterating the updates in Eq. (12), the sequence of estimates will converge to a local minimum $x_{min} = \arg\min_x F(x)$. We will show this by defining an appropriate auxiliary function for the objective function $\|\mathbf{X} - \mathbf{UZ}^T\mathbf{A}^T\|$.

First, we prove the convergence of the update rule in Eq. (11). For any element z_{ab} in $\mathbf{Z}$, let $F_{z_{ab}}$ denote the part of $\mathcal{O}$ relevant to z_{ab}. Since the update is essentially elementwise, it is sufficient to show that each $F_{z_{ab}}$ is nonincreasing under the update step of (11). We prove this by defining the auxiliary function regarding z_{ab} as follows.

Lemma 3. *Let F' denote the first order derivative with respective to* $\mathbf{Z}$*. The function*

$$\begin{aligned} G(z, z_{ab}^t) &= F_{z_{ab}}(z_{ab}^t) + F'_{z_{ab}}(z_{ab}^t)(z - z_{ab}^t) \\ &\quad + \frac{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab}}{z_{ab}^t}(z - z_{ab}^t)^2 \quad (13) \end{aligned}$$

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

Proof. Obviously, $G(z, z) = F_{z_{ab}}(z)$. According to the definition of auxiliary function, we only need to show that $G(z, z_{ab}^t) \geq F_{z_{ab}}(z)$. In order to do this, we compare $G(z, z_{ab}^t)$ in Eq. (13) with the Taylor series expansion of $F_{z_{ab}}(z)$

$$\begin{aligned} F_{z_{ab}}(z) &= F_{z_{ab}}(z_{ab}^t) + F'_{z_{ab}}(z - z_{ab}^t) \\ &\quad + \frac{1}{2}F''_{z_{ab}}(z - z_{ab}^t)^2, \quad (14) \end{aligned}$$

where F'' is the second order derivative with respect to $\mathbf{Z}$. It is easy to check that

$$F'_{z_{ab}} = \left(\frac{\partial \mathcal{O}}{\partial \mathbf{Z}}\right)_{ab} = (-2\mathbf{A}^T\mathbf{X}^T\mathbf{U} + 2\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab} \quad (15)$$

$$F''_{z_{ab}} = 2(\mathbf{A}^T\mathbf{A})_{aa}(\mathbf{U}^T\mathbf{U})_{bb} \quad (16)$$

Putting Eq. (16) into Eq. (14) and comparing Eq. (13) and (14), we can see that, instead of showing $G(z, z_{ab}^t) \geq F_{z_{ab}}(z)$, it is equivalent to show

$$\frac{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab}}{z_{ab}^t} \geq \frac{1}{2}F''_{z_{ab}} = (\mathbf{A}^T\mathbf{A})_{aa}(\mathbf{U}^T\mathbf{U})_{bb} \quad (17)$$

To prove the above inequality, we have

$$\begin{aligned} (\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab} &= \sum_{l=1}^k (\mathbf{A}^T\mathbf{AZ})_{al}(\mathbf{U}^T\mathbf{U})_{lb} \\ &\geq (\mathbf{A}^T\mathbf{AZ})_{ab}(\mathbf{U}^T\mathbf{U})_{bb} \\ &\geq \sum_{l=1}^k (\mathbf{A}^T\mathbf{A})_{al}z_{lb}^t(\mathbf{U}^T\mathbf{U})_{bb} \\ &\geq z_{ab}^t(\mathbf{A}^T\mathbf{A})_{aa}(\mathbf{U}^T\mathbf{U})_{bb} \end{aligned}$$

□

Then we define an auxiliary function for the update rule in Eq. (10). Similarly, let $F_{u_{ab}}$ denote the part of $\mathcal{O}$ relevant to u_{ab}. Then the auxiliary function regarding u_{ab} is defined as follows.

Lemma 4. *The function*

$$\begin{aligned} G(u, u_{ab}^t) &= F_{u_{ab}}(u_{ab}^t) + F'_{u_{ab}}(u_{ab}^t)(u - u_{ab}^t) \\ &\quad + \frac{(\mathbf{UZ}^T\mathbf{A}^T\mathbf{AZ})_{ab}}{u_{ab}^t}(u - u_{ab}^t)^2 \quad (18) \end{aligned}$$

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

The proof of Lemma 4 is essentially similar to the proof of Lemma 3 and is omitted here due to space limitation. With the above lemmas, now we give the proof of Theorem 1.

Proof of Theorem 1: Putting $G(z, z_{ab}^t)$ of Eq. (13) into Eq. (12), we get:

$$\begin{aligned} z_{ab}^{t+1} &= z_{ab}^t - z_{ab}^t \frac{F'_{z_{ab}}(z_{ab}^t)}{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab}} \\ &= z_{ab}^t \frac{(\mathbf{A}^T\mathbf{X}^T\mathbf{U})_{ab}}{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab}} \qquad (19) \end{aligned}$$

Since Eq. (13) is an auxiliary function, $F_{z_{ab}}$ is nonincreasing under this update rule, according to Lemma 3.

Similarly, putting $G(u, u_{ab}^t)$ of Eq. (18) into Eq. (12), we get:

$$\begin{aligned} u_{ab}^{t+1} &= u_{ab}^t - u_{ab}^t \frac{F'_{u_{ab}}(z_{ab}^t)}{(\mathbf{A}^T\mathbf{AZU}^T\mathbf{U})_{ab}} \\ &= u_{ab}^t \frac{(\mathbf{XAZ})_{ab}}{(\mathbf{UZ}^T\mathbf{A}^T\mathbf{AZ})_{ab}} \qquad (20) \end{aligned}$$

Since Eq. (18) is an auxiliary function, $F_{u_{ab}}$ is nonincreasing under this update rule, according to Lemma 4.

Experimental Results

In this section, we investigate the use of our proposed CNMF algorithm for data clustering. We begin with a description of the data sets used in our experiments.

Data Sets

The experiments are conducted on two data sets. One is the AT&T database [1], and the other is the Yale Face database [2]. The AT&T database consists of ten different images for each of 40 distinct subjects, thus 400 images in total. The Yale Database contains 165 grayscale images of 15 individuals. All images demonstrate variations in lighting condition (left-light, centerlight, right-light), facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

In all the experiments, images are preprocessed so that faces are located. Original images are first normalized in scale and orientation such that the two eyes are aligned at the same position. Then the facial areas were cropped into the final images for clustering. Each image is 32 × 32 pixels with 256 gray levels per pixel.

Evaluation Metrics

We use two metrics to evaluate the clustering performance (Xu, Liu, and Gong 2003; Cai et al. 2008). The result is evaluated by comparing the cluster label of each sample with the label provided by the data set. One metric is accuracy (AC), which is used to measure the percentage of correct labels obtained. Given a data set containing n images, for each sample image m_i, let l_i be the cluster label we obtained by applying different algorithms and r_i be the label provided by the data set. The accuracy (AC) is defined as:

$$AC = \frac{\sum_{i=1}^{n} \delta(r_i, map(l_i))}{n} \qquad (21)$$

where $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(l_i)$ is the mapping function that maps each cluster label l_i to the equivalent label from the data set. The best mapping can be found by using the Kuhn-Munkres algorithm (Lovasz and Plummer 1986).

The second metric is the normalized mutual information ($\widehat{MI}$). In clustering applications, mutual information is used to measure how similar two sets of clusters are. Given two sets of image clusters $\mathcal{C}$ and $\mathcal{C}'$, their mutual information metric $MI(\mathcal{C}, \mathcal{C}')$ is defined as:

$$MI(\mathcal{C}, \mathcal{C}') = \sum_{c_i \in \mathcal{C}, c'_j \in \mathcal{C}'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \qquad (22)$$

where $p(c_i)$, $p(c'_j)$ denote the probabilities that an image arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ denotes the joint probability that this arbitrarily selected image belongs to the cluster c_i as well as c'_j at the same time. $MI(\mathcal{C}, \mathcal{C}')$ takes values between zero and $\max(H(\mathcal{C}), H(\mathcal{C}'))$, where $H(\mathcal{C})$ and $H(\mathcal{C}')$ are the entropies of $\mathcal{C}$ and $\mathcal{C}'$, respectively. It reaches the maximum $\max(H(\mathcal{C}), H(\mathcal{C}'))$ when the two sets of image clusters are identical and it becomes zero when the two sets are completely independent. One important character of $MI(\mathcal{C}, \mathcal{C}')$ is that the value keeps the same for all kinds of permutations. In our experiments, we use the normalized metric $\widehat{MI}(\mathcal{C}, \mathcal{C}')$ which takes values between zero and one:

$$\widehat{MI}(\mathcal{C}, \mathcal{C}') = \frac{MI(\mathcal{C}, \mathcal{C}')}{\max(H(\mathcal{C}), H(\mathcal{C}'))} \qquad (23)$$

Performance Evaluations and Comparisons

We compare the following algorithms:

- Our proposed Constrained Non-negative Matrix Factorization (**CNMF**).
- Non-negative Matrix Factorization based clustering (**NMF**). We implemented a normalized cut weighted version of NMF as suggested in (Xu, Liu, and Gong 2003).
- Non-negative Tensor Factorization (**NTF**) (Shashua and Hazan 2005). NTF is an extension of NMF to tensor data. In NTF, each face image is represented as a second order tensor, rather than a vector.
- Graph regularized Non-negative Matrix Factorization (**GNMF**) (Cai et al. 2008) which encode the geometrical information of the data space into matrix factorization.
- Semi-supervised Graph regularized Non-negative Matrix Factorization on Manifold (**SemiGNMF**)(Cai et al. 2008). This method incorporate the label information into the graph structure by modifying the weight matrix.

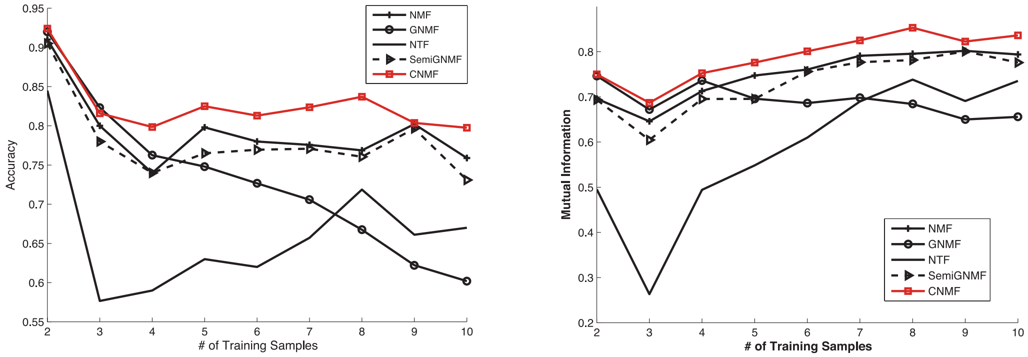
As we mentioned before, there is no parameter in our approach. For other algorithms, the parameters are set to be the values that each algorithm can achieve its best results.

[1] http://www.uk.research.att.com/facedatabase.html

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Table 1: Clustering Results Comparison on the AT&T database

k	Accuracy (%)					Normalized Mutual Information (%)				
	NMF	GNMF	NTF	SemiGNMF	CNMF	NMF	GNMF	NTF	SemiGNMF	CNMF
2	91.0 ± 11.0	92.0 ± 11.6	84.5 ± 11.7	90.5 ± 11.4	**92.4 ± 9.5**	69.6 ± 32.5	74.7 ± 34.6	49.5 ± 29.5	69.4 ± 34.5	**75.0 ± 30.5**
3	80.0 ± 13.1	**82.3 ± 12.0**	57.7 ± 10.3	78.0 ± 14.3	81.6 ± 12.5	64.6 ± 19.2	67.2 ± 18.2	26.3 ± 17.4	60.5 ± 23.0	**68.7 ± 16.7**
4	74.0 ± 10.8	76.3 ± 14.1	59.0 ± 9.1	74.0 ± 11.4	**79.8 ± 8.0**	71.3 ± 8.7	73.6 ± 14.6	49.4 ± 11.0	69.5 ± 9.7	**75.3 ± 7.8**
5	79.8 ± 10.2	74.8 ± 12.2	63.0 ± 11.2	76.5 ± 10.0	**82.5 ± 8.6**	74.7 ± 8.2	69.6 ± 13.4	54.8 ± 13.8	69.5 ± 10.9	**77.6 ± 9.5**
6	78.0 ± 11.0	72.7 ± 18.0	62.0 ± 9.5	77.0 ± 10.1	**81.3 ± 8.0**	76.0 ± 9.4	68.6 ± 16.4	61.0 ± 8.1	75.6 ± 9.2	**80.1 ± 7.8**
7	77.6 ± 9.3	70.6 ± 6.4	65.7 ± 8.1	77.1 ± 6.8	**82.4 ± 5.6**	79.1 ± 6.6	69.8 ± 6.6	69.0 ± 6.0	77.7 ± 4.1	**82.5 ± 4.6**
8	76.9 ± 7.6	66.8 ± 12.3	71.9 ± 7.8	76.0 ± 6.3	**83.7 ± 7.9**	79.6 ± 6.3	68.4 ± 13.7	73.8 ± 6.7	78.2 ± 5.8	**85.3 ± 6.4**
9	80.2 ± 7.9	62.2 ± 9.2	66.1 ± 11.7	79.7 ± 6.6	**80.4 ± 6.9**	80.2 ± 7.7	65.0 ± 8.4	69.1 ± 9.9	80.1 ± 6.8	**82.3 ± 5.5**
10	75.9 ± 7.3	60.2 ± 10.4	67.0 ± 6.4	73.1 ± 8.2	**79.8 ± 5.5**	79.4 ± 5.3	65.6 ± 7.9	73.5 ± 4.7	77.6 ± 6.3	**83.6 ± 3.9**
Avg.	79.3 ± 9.8	73.1 ± 11.8	66.3 ± 9.5	78.0 ± 9.5	**82.7 ± 8.1**	74.9 ± 11.5	69.2 ± 14.9	58.5 ± 11.9	73.1 ± 12.3	**78.9 ± 10.3**



(a) Accuracy vs. number of clusters

(b) Mutual Information vs. number of clusters

Figure 1: Clustering Performance on AT&T Database

The evaluations are conducted with different cluster numbers k ranging from two to ten. For the given cluster number k, we randomly choose k clusters from the data set, and repeat this process ten times. The average clustering performance is recorded over the ten tests. For fixed chosen clusters, we apply different matrix/tensor factorization algorithms to obtain new representations. K-means is then applied in the new representation spaces 20 times with different start points and the best result in terms of the objective function of K-means is recorded. Two different images are randomly selected from each cluster with labels.

Fig. 1 and 2 show the plots of accuracy and normalized mutual information versus the number of clusters for different algorithms. As can be seen, our proposed CNMF algorithm consistently outperforms all the other algorithms. NMF, GNMF, and SemiGNMF perform comparably to one another. SemiGNMF fails to make full use of the label information, and in some cases performs even worse than GNMF and NMF. This is because there is no theoretical guarantee for SemiGNMF that data points sharing the same label can be mapped sufficiently close to one another.

Table 1 and 2 show the detailed clustering accuracy (normalized mutual information), as well as the standard deviation. The last row of each table shows the average accuracy (normalized mutual information) over k. On AT&T data set, comparing to the second best algorithm, i.e. NMF, CNMF achieves 3.4% improvement in accuracy and 4.0% improvement in normalized mutual information. On Yale data set, comparing to the second best algorithm, i.e. GNMF, CNMF achieves 2.0% improvement in accuracy and 2.3% improvement in normalized mutual information.

Conclusions

In this paper, we have presented a novel matrix factorization method, called Constrained Non-negative Matrix Factorization (CNMF), which makes use of both labeled and unlabeled data points. CNMF imposes the label information to the objective function as hard constraints. This way, the new representations of the data points can have more discriminating power. Moreover, our algorithm is parameter free. Thus, CNMF can be easily applied to a wide range of practical problems. The experimental results on two standard face databases have demonstrated the effectiveness of our approach.

Acknowledgements

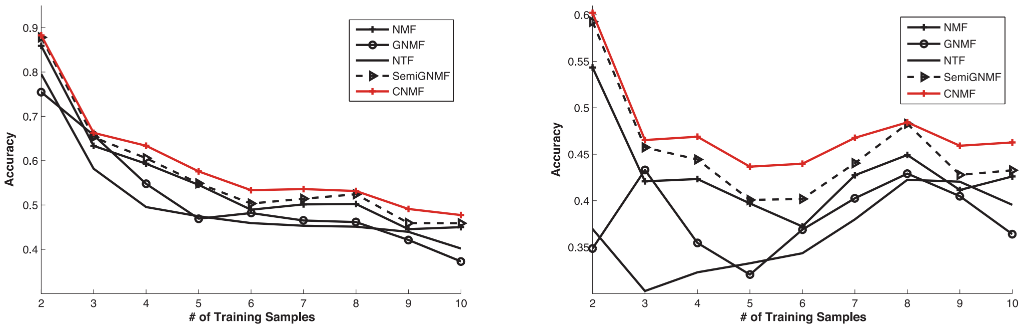
This work was supported by Science Foundation of Chinese University under grant 2009QNA5024.

References

Cai, D.; He, X.; Wu, X.; and Han, J. 2008. Non-negative matrix factorization on manifold. In *Proc. 8th IEEE International Confer-*

Table 2: Clustering Results Comparison on the Yale Face database

k	Accuracy (%)					Normalized Mutual Information (%)				
	NMF	GNMF	NTF	SemiGNMF	CNMF	NMF	GNMF	NTF	SemiGNMF	CNMF
2	85.9 ± 12.9	75.5 ± 17.8	79.5 ± 11.8	87.8 ± 11.8	**88.3 ± 10.2**	54.3 ± 32.5	34.9 ± 34.3	37.0 ± 30.2	59.3 ± 30.2	**60.2 ± 25.8**
3	63.3 ± 9.1	65.8 ± 10.1	58.2 ± 21.6	65.3 ± 5.9	**66.3 ± 6.3**	42.1 ± 15.9	43.3 ± 17.9	30.3 ± 33.0	45.7 ± 12.3	**46.5 ± 13.1**
4	59.3 ± 9.1	54.8 ± 9.5	49.5 ± 10.1	60.5 ± 8.8	**63.4 ± 6.4**	42.3 ± 11.7	35.5 ± 13.9	32.3 ± 15.8	44.4 ± 12.5	**46.9 ± 9.2**
5	54.5 ± 7.8	46.9 ± 2.7	47.5 ± 4.9	54.9 ± 3.7	**57.6 ± 4.8**	39.7 ± 9.2	32.1 ± 6.4	33.3 ± 7.2	40.1 ± 5.7	**43.7 ± 6.3**
6	48.9 ± 6.6	48.2 ± 6.4	45.9 ± 7.3	50.3 ± 6.1	**53.3 ± 6.5**	37.2 ± 7.0	36.9 ± 8.0	34.4 ± 8.4	40.2 ± 7.0	**44.0 ± 7.7**
7	50.1 ± 5.9	46.5 ± 7.3	45.3 ± 6.5	51.4 ± 4.5	**53.6 ± 5.5**	42.7 ± 6.7	40.3 ± 8.7	38.0 ± 8.1	44.0 ± 5.5	**46.8 ± 5.5**
8	50.2 ± 6.5	46.1 ± 4.8	45.1 ± 3.3	52.4 ± 4.9	**53.1 ± 4.6**	44.9 ± 7.2	42.9 ± 5.7	42.3 ± 5.5	48.3 ± 4.4	**48.4 ± 4.4**
9	44.5 ± 5.7	42.1 ± 6.3	43.9 ± 5.3	45.9 ± 5.4	**49.1 ± 4.0**	41.2 ± 5.6	40.5 ± 6.8	42.0 ± 7.0	42.8 ± 5.2	**45.9 ± 5.0**
10	45.0 ± 3.6	37.3 ± 5.4	40.2 ± 5.5	45.9 ± 4.1	**47.7 ± 3.0**	42.6 ± 3.5	36.4 ± 5.7	39.6 ± 3.9	43.3 ± 3.7	**46.3 ± 2.9**
Avg.	55.8 ± 7.5	51.5 ± 7.8	50.6 ± 8.5	57.2 ± 6.1	**59.2 ± 5.7**	43.0 ± 11.0	38.1 ± 11.9	36.6 ± 13.2	45.3 ± 9.6	**47.6 ± 8.9**



(a) Accuracy vs. number of clusters

(b) Mutual Information vs. number of clusters

Figure 2: Clustering Performance on Yale Database

ence on Data Mining, 63–72.

Cai, D.; He, X.; Wang, X.; Bao, H.; and Han, J. 2009. Locality preserving nonnegative matrix factorization. In *Proc. International Joint Conference on Artificial Intelligence*.

Chapelle, O.; Schölkopf, B.; and Zien, A., eds. 2006. *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

Dempster, A.; Laird, N.; and Rubin, D. 1977. Maximum likelihood from incomplete data via the em algorithm. In *Joural the Royal Statistics Society*, 1–38.

Dhillon, I. S.; Mallela, S.; and Modha, D. S. 2003. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*.

He, X., and Niyogi, P. 2003. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*.

He, X.; Yan, S.; Hu, Y.; Niyogi, P.; and Zhang, H.-J. 2005. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(3):328–340.

He, X.; Cai, D.; and Min, W. 2005. Statistical and computational analysis of locality preserving projection. In *Proceedings of the* 22^{nd} *International Conference on Machine Learning*.

He, X.; Ji, M.; and Bao, H. 2009. Graph embedding with constraints. In *Proc. 21st International Joint Conference on Artificial Intelligence*.

He, X. 2010. Laplacian regularized D-optimal design for active learning and its application to image retrieval. *IEEE Trans. on Image Processing* 19(1):254–263.

Hofmann, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning* 177–196.

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

Lee, D. D., and Seung, H. S. 2001. Algorithms for non-negative matrix factorization. In *NIPS 13*.

Li, T., and Ding, C. 2006. The relationships among various nonnegative matrix factorization methods for clustering. In *Proc. 6th IEEE International Conference on Data Mining*.

Li, S.; Hou, X.; Zhang, H.; and Cheng, Q. 2001. Learning spatially localized, parts-based representation. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 207–212.

Liu, W.; Zheng, N.; and Lu, X. 2003. Non-negative matrix factorization for visual coding. In *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*.

Lovasz, L., and Plummer, M. 1986. *Matching Theory*. North Holland, Budapest: Akadémiai Kiadó.

Min, W.; Lu, K.; and He, X. 2004. Locality pursuit embedding. *Pattern Recognition* 37(4):781–788.

Saul, L., and Pereira, F. 1977. Aggregate and mixed-order markov models for statistical language processing. In *Proc. the Second Conference on Empirical Methods in Nature Language Processing*, 81–89. ACL Press.

Shashua, A., and Hazan, T. 2005. Non-negative tensor factorization with applications to statistics and computer vision. In *Proc. International Conference on Machine Learning (ICML)*, 793–800.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *Proc. 2003 Annual ACM SIGIR Conference*.