Locality-Constrained Concept Factorization

Haifeng Liu **Zheng Yang** **Zhaohui Wu**
College of Computer Science, Zhejiang University
Hangzhou, Zhejiang, China 310058
{haifengliu,11021059,wzh}@zju.edu.cn.

Abstract

Matrix factorization based techniques, such as nonnegative matrix factorization (NMF) and concept factorization (CF), have attracted great attention in dimension reduction and data clustering. Both of them are linear learning problems and lead to a sparse representation of the data. However, the sparsity obtained by these methods does not always satisfy locality conditions, thus the obtained data representation is not the best. This paper introduces a locality-constrained concept factorization method which imposes a locality constraint onto the traditional concept factorization. By requiring the concepts (basis vectors) to be as close to the original data points as possible, each data can be represented by a linear combination of only a few basis concepts. Thus our method is able to achieve sparsity and locality at the same time. We demonstrate the effectiveness of this novel algorithm through a set of evaluations on real world applications.

1 Introduction

Data representation is a fundamental problem in many real world applications such as pattern recognition, computer vision, image clustering etc. Especially linear data representations, such as principal component analysis (PCA), locality preserving projection (LPP) [He and Niyogi, 2003; He *et al.*, 2005], independent component analysis (ICA), nonnegative matrix factorization (NMF) [Lee and Seung, 1999] and concept factorization (CF) [Xu and Gong, 2004], have been widely used in these data analysis tasks.

Matrix factorization has been frequently used in linear data representation. Given a data matrix $\mathbf{X}$, the above algorithms aim to find two or more matrix factors whose product is a good approximation to the original matrix. One factor consists of a set of new basis vectors which reveals the latent semantic structure, and the other factor can be considered as the coefficients (also referred as encodings) where each data point is a linear combination of the found bases. In real applications, the dimension of the new basis vectors is usually much smaller than that of the original data matrix. This gives rise to a compact representation of the data points which can facilitate other learning tasks such as clustering and classification.

NMF and CF have attracted great attention as efficient matrix decomposition methods. NMF aims to decompose a data matrix $\mathbf{X}$ into two matrix factors $\mathbf{U}$ and $\mathbf{V}$ so that $\mathbf{UV}$ provides a good approximation to $\mathbf{X}$. The CF model is a variation of NMF in that each cluster is expressed by a linear combination of the data points and each data point is represented by a linear combination of the cluster centers. The major advantage of CF over NMF is that the NMF algorithm can only be performed in the original feature space of the data points, but the CF method can be performed on any data representation space, so that it can be kernelized and the powerful idea of the kernel method can be applied [Xu and Gong, 2004].

Since the dimension of the decomposed factors is much smaller than that of the original matrix, both NMF and CF map the data from high dimensional space to a low dimensional representation and obtain a sparse encoding of the data. However, the sparsity obtained by these methods does not always satisfy locality conditions. Since the local points share the greatest similarity, it would be more natural to represent the basis vectors using a few *nearby* anchor points, which leads to a more efficient representation of the data.

Recently, Cai et al. [Cai *et al.*, 2011] proposed a Locally Consistent Concept Factorization approach to encode the geometrical information of the data space, which can extract the document concepts with respect to the intrinsic manifold structure. This method is able to discover the local geometrical structure, but does not always satisfy the locality conditions as we mentioned before. As to add sparseness constraint to the matrix factorization, Hoyer [Hoyer, 2004] showed how explicitly incorporating the notion of 'sparseness' improves the found decompositions. However, there is no work that includes locality and sparsity constraints at the same time to the best of our knowledge.

In this paper, we introduce a novel matrix factorization algorithm, called *Locality-constrained Concept Factorization (LCF)* which imposes a locality constraint onto the traditional concept factorization. By requiring the concepts (basis vectors) to be as close to the original data points as possible, each data can be represented by a linear combination of only a few nearby basis concepts, thus achieving sparsity and locality at the same time.

To achieve this goal, we incorporate the idea of Local Co-

ordinate Coding [Yu *et al.*, 2009] into the original objective function of concept factorization. And we also propose a multiplicative algorithm to efficiently solve the corresponding optimization problem.

2 Background

Factorization of matrices is generally non-unique, and a number of different methods of doing so have been developed by incorporating different constraints.

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 an $m \times k$ matrix $\mathbf{U}$ and a $k \times n$ matrix $\mathbf{V}$ where the product of these two factors is an approximation to the original matrix, represented as $\mathbf{X} \approx \mathbf{UV}$. 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 $\mathbf{V}$: $\mathbf{x}_i \approx \sum_{j=1}^k \mathbf{u}_j v_{ji}$.

The speciality of NMF is that it enforces that all entries of the factor matrices must be non-negative. One limitation of NMF is that the non-negative requirement is not applicable to applications where the data involves negative number. The second is that it is not clear how to effectively perform NMF in the transformed data space so that the powerful kernel method can be applied.

Concept Factorization is proposed striving to address the above problems while inheriting all the strengths of the NMF method. In the CF model, we rewrite the NMF model by representing each base (cluster center) $\mathbf{u}_j$ by a linear combination of the data points $\mathbf{u}_j = \sum_i w_{ij}\mathbf{x}_i$ where $w_{ij} \geq 0$. Let $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n\times k}$, CF tries to decompose the data matrix satisfied the following condition

$$\mathbf{X} \approx \mathbf{XWV}.$$

Using the Frobenius norm to qualify the approximation, CF tries to minimize the following objective function

$$\mathcal{O} = \|\mathbf{X} - \mathbf{XWV}\|^2 \tag{1}$$

3 Locality-constrained Concept Factorization

In this section, we introduce our *Locality-constrained Concept Factorization* (LCF) algorithm, which takes the locality constraint as an additional requirement. The algorithm presented in this paper is fundamentally motivated from the concept of local coordinate coding [Yu *et al.*, 2009].

3.1 The Objective Function

First we introduce the concept of coordinate coding.

Definition A coordinate coding is a pair (γ, C), where $C \subset \mathbb{R}^d$ is a set of anchor points, and γ is a map of $\mathbf{x} \in \mathbb{R}^d$ to $[\gamma_v(\mathbf{x})]_{v\in C} \in R^{|C|}$ such that $\sum_v \gamma_v(\mathbf{x}) = 1$. It induces the following physical approximation of $\mathbf{x}$ in $\mathbb{R}^d$: $\gamma(\mathbf{x}) = \sum_{v\in C} \gamma_v(\mathbf{x})v$.

According to this definition, the CF model can be considered as a coordinate coding where the basis vectors $\mathbf{u}_j = \sum_i w_{ij}\mathbf{x}_i$ are a set of anchor points, and each column of $\mathbf{V}$ contains the coordinates for each data point with respect to the anchor points. In order to add the local sparse constraint to the traditional concept factorization, we require that each original data point should be sufficiently *close* to only *a few* anchor points. This can be achieved by introducing the following term to measure the locality and sparsity penalty:

$$\mathcal{R} = \sum_{k=1}^K |v_{ki}| \left\|\mathbf{u}_k - \mathbf{x}_i\right\|^2 = \sum_{k=1}^K |v_{ki}| \left\|\sum_{j=1}^N w_{jk}\mathbf{x}_j - \mathbf{x}_i\right\|^2$$

The above constraint incurs a heavy penalty if $\mathbf{x}_i$ is far away from the anchor point $\mathbf{u}_j$ while its coordinate v_{ji} with respect to $\mathbf{u}_j$ is large. Therefore, by minimizing $\mathcal{R}$, we essentially try to formalize our intuition that $\mathbf{x}_i$ is close to the anchor points $\mathbf{u}_j$ as much as possible, otherwise its coordinate with respect to $\mathbf{u}_j$ tends to be zero.

With the locality constraint, our LCF algorithm reduces to minimize the following objective function:

$$\mathcal{O} = \|\mathbf{X} - \mathbf{XWV}\|^2 + \lambda \sum_{i=1}^N \sum_{k=1}^K |v_{ki}| \left\|\sum_{j=1}^N w_{jk}\mathbf{x}_j - \mathbf{x}_i\right\|^2 \tag{2}$$

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

3.2 The Algorithm

We introduce an iterative algorithm to find a local minimum for the optimization problem. The objective function can be rewritten as follows:

$$\begin{aligned}\mathcal{O} =& \|\mathbf{X} - \mathbf{XWV}\|^2 + \lambda \sum_{i=1}^N \sum_{k=1}^K |v_{ki}| \left\|\sum_{j=1}^N w_{jk}\mathbf{x}_j - \mathbf{x}_i\right\|^2 \\ =& \|\mathbf{X} - \mathbf{XWV}\|^2 + \lambda \sum_{i=1}^N \left\|(\mathbf{x}_i\mathbf{1}^T - \mathbf{XW})\mathbf{D}_i^{1/2}\right\|^2\end{aligned}$$

where $\mathbf{D}_i = diag(|\mathbf{v}_i|) \in \mathbb{R}^{K\times K}$. Using the matrix property $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$, $\|\mathbf{A}\|^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{A})$ and $\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$, we have

$$\begin{aligned}\mathcal{O} =& \mathrm{Tr}\Big((\mathbf{X} - \mathbf{XWV})(\mathbf{X} - \mathbf{XWV})^T \\ &+ \lambda \sum_{i=1}^N (\mathbf{x}_i\mathbf{1}^T - \mathbf{XW})\mathbf{D}_i(\mathbf{x}_i\mathbf{1}^T - \mathbf{XW})^T\Big) \\ =& \mathrm{Tr}\Big(\mathbf{XX}^T - 2\mathbf{XWVX}^T + \mathbf{XWVV}^T\mathbf{W}^T\mathbf{X}^T \\ &+ \lambda \sum_{i=1}^N (\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i\mathbf{1}\mathbf{x}_i^T - 2\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i\mathbf{W}^T\mathbf{X}^T \\ &+ \mathbf{XWD}_i\mathbf{W}^T\mathbf{X}^T)\Big)\end{aligned} \tag{3}$$

Let ψ_{jk} and ϕ_{ki} be the Langrange multiplier for constraints $w_{jk} \geq 0$ and $v_{ki} \geq 0$, respectively. We define matrix $\Psi =$

$[\psi_{jk}]$ and $\Phi = [\phi_{ki}]$, then the Langrange $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} =&\mathrm{Tr}\Big(\mathbf{X}\mathbf{X}^T - 2\mathbf{X}\mathbf{W}\mathbf{V}\mathbf{X}^T + \mathbf{X}\mathbf{W}\mathbf{V}\mathbf{V}^T\mathbf{W}^T\mathbf{X}^T \\ &+ \lambda\sum_{i=1}^{N}(\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i\mathbf{1}\mathbf{x}_i^T - 2\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i\mathbf{W}^T\mathbf{X}^T \\ &+ \mathbf{X}\mathbf{W}\mathbf{D}_i\mathbf{W}^T\mathbf{X}^T)\Big) + \mathrm{Tr}(\Psi\mathbf{W}^T) + \mathrm{Tr}(\Phi\mathbf{V}^T)\end{aligned}$$

Define $\mathbf{K} = \mathbf{X}^T\mathbf{X}$, define a column vector $\mathbf{a} = diag(\mathbf{K}) \in \mathbb{R}^N$. Let $\mathbf{A} = (\mathbf{a}, \cdots, \mathbf{a})^T$ be a $K \times N$ matrix whose rows are $\mathbf{a}^T$. Define a column vector $\mathbf{b} = diag(\mathbf{W}^T\mathbf{K}\mathbf{W}) \in \mathbb{R}^K$. Let $\mathbf{B} = (\mathbf{b}, \cdots, \mathbf{b})$ be a $K \times N$ matrix whose columns are $\mathbf{b}$. The partial derivatives of $\mathcal{L}$ with respect to $\mathbf{W}$ and $\mathbf{V}$ are:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{W}} =&2\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T - 2\mathbf{K}\mathbf{V}^T \\ &+ \lambda\sum_{i=1}^{N}(-2\mathbf{X}^T\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i + 2\mathbf{K}\mathbf{W}\mathbf{D}_i) + \Psi\end{aligned}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} =2\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V} - 2\mathbf{W}^T\mathbf{K} + \lambda(\mathbf{A} - 2\mathbf{W}^T\mathbf{K} + \mathbf{B}) + \Phi$$

Using the Karush-Kuhn-Tucker conditions (also known as the KKT conditions) $\psi_{jk}w_{jk} = 0$ and $\phi_{ki}v_{ki} = 0$, we get the following equations:

$$\begin{aligned}&(\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T)_{jk}w_{jk} - (\mathbf{K}\mathbf{V}^T)_{jk}w_{jk} \\ &+ \lambda(\sum_{i=1}^{N}\mathbf{K}\mathbf{W}\mathbf{D}_i)_{jk}w_{jk} - \lambda(\sum_{i=1}^{N}\mathbf{X}^T\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i)_{jk}w_{jk} = 0\\ &2(\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V})_{ki}v_{ki} - 2(\mathbf{W}^T\mathbf{K})_{ki}v_{ki} \\ &\qquad + \lambda(\mathbf{A} - 2\mathbf{W}^T\mathbf{K} + \mathbf{B})_{ki}v_{ki} = 0\end{aligned}$$

The above equations lead to the following update rules:

$$w_{jk} \leftarrow w_{jk}\frac{(\mathbf{K}\mathbf{V}^T + \lambda\sum_{i=1}^{N}\mathbf{X}^T\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i)_{jk}}{(\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T + \lambda\sum_{i=1}^{N}\mathbf{K}\mathbf{W}\mathbf{D}_i)_{jk}} \quad (4)$$

$$v_{ki} \leftarrow v_{ki}\frac{2(\lambda+1)(\mathbf{W}^T\mathbf{K})_{ki}}{(2\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V} + \lambda\mathbf{A} + \lambda\mathbf{B})_{ki}} \quad (5)$$

We have the following theorem regarding the above iterative updating rules. Theorem 1 guarantees the convergence of the iterations in Eq. (4) and (5) and therefore the final solution will be a local optimum.

Theorem 1. *The objective function $\mathcal{O}$ in Eq. (2) is nonincreasing under the update rules in Eq. (4) and Eq. (5). The objective function is invariant under these updates if and only if $\mathbf{W}$ and $\mathbf{V}$ are at a stationary point.*

To prove Theorem 1, we use an auxiliary function similar to that used in the Expectation-Maximization algorithm.To make the proof complete, we restate the definition of auxiliary function and its property which will be used to prove the algorithm convergence.

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.

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

$$x^{t+1} = \arg\min_x G(x, x'). \quad (6)$$

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. (6), the sequence of estimates will converge to a local minimum $x_{min} = \arg\min_x F(x)$. Next, we will define an auxiliary function for our objective function and use Lemma 2 to show that the minimum of the objective function is exactly our update rule, thereby Theorem 1 is proved.

First, we prove the convergence of the update rule in Eq. (4). Considering any element w_{ab} in W, we use $F_{w_{ab}}$ to denote the part of $\mathcal{O}$ which is only relevant to w_{ab}. Since the update is essentially element-wise, it is sufficient to show that each $F_{w_{ab}}$ is nonincreasing under the update step of (4). We prove this by defining the auxiliary function G for $F_{w_{ab}}$ as follows.

Lemma 3. *The function*

$$\begin{aligned}G(w, w_{ab}^{(t)}) &= F_{w_{ab}}(w_{ab}^{(t)}) + F'_{w_{ab}}(w_{ab}^{(t)})(w - w_{ab}^{(t)}) \\ &+ \frac{(\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T)_{ab} + \lambda\sum_{i=1}^{N}(\mathbf{K}\mathbf{W}\mathbf{D}_i)_{ab}}{w_{ab}^{(t)}}(w - w_{ab}^{(t)})^2\end{aligned} \quad (7)$$

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

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

$$\begin{aligned}F_{w_{ab}}(w) =&F_{w_{ab}}(w_{ab}^{(t)}) + F'_{w_{ab}}(w_{ab}^{(t)})(w - w_{ab}^{(t)}) \\ &+ \Big(\mathbf{K}\mathbf{V}\mathbf{V} + \lambda\sum_{i=1}^{N}\mathbf{K}\mathbf{D}_i\Big)_{ab}(w - w_{ab}^{(t)})^2\end{aligned}$$

Since

$$\frac{\partial^2\mathcal{O}}{\partial W^2} = 2\mathbf{K}\mathbf{V}\mathbf{V} + 2\lambda\sum_{i=1}^{N}\mathbf{K}\mathbf{D}_i$$

and

$$\begin{aligned}&(\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T)_{ab} + \lambda\sum_{i=1}^{N}(\mathbf{K}\mathbf{W}\mathbf{D}_i)_{ab} \\ =&\sum_k(\mathbf{K}\mathbf{W})_{ak}(\mathbf{V}\mathbf{V}^T)_{kb} + \lambda\sum_{i=1}^{N}\sum_k(\mathbf{K}\mathbf{W})_{ak}(\mathbf{D}_i)_{kb} \\ \geq&(\mathbf{K}\mathbf{W})_{ab}(\mathbf{V}\mathbf{V}^T)_{bb} + \lambda\sum_{i=1}^{N}(\mathbf{K}\mathbf{W})_{ab}(\mathbf{D}_i)_{bb} \\ \geq&\sum_k(\mathbf{K})_{ak}w_{kb}^{(t)}(\mathbf{V}\mathbf{V}^T)_{bb} + \lambda\sum_{i=1}^{N}\sum_k(\mathbf{K})_{ak}w_{kb}^{(t)}(\mathbf{D}_i)_{bb} \\ \geq&w_{ab}^{(t)}\Big((\mathbf{K})_{aa}(\mathbf{V}\mathbf{V}^T)_{bb} + \lambda\sum_{i=1}^{N}(\mathbf{K})_{aa}(\mathbf{D}_i)_{bb}\Big) \\ \geq&\frac{1}{2}w_{ab}^{(t)}F''_{w_{ab}}\end{aligned}$$

Thus, $G(w, w_{ab}^{(t)}) \geq F_{w_{ab}}(w)$. □

Then we define an auxiliary function for the update rule in Eq. (5). Similarly, let $F_{v_{ab}}$ denote the part of $\mathcal{O}$ relevant to v_{ab}. Then the auxiliary function regarding v_{ab} is defined as follows.

Lemma 4. *Function*

$$G(v, v_{ab}^{(t)}) = F_{v_{ab}}(v_{ab}^{(t)}) + F'_{v_{ab}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) + \frac{(\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{V} + \frac{1}{2}\lambda\mathbf{A} + \frac{1}{2}\lambda\mathbf{B})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \quad (8)$$

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

The proof 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: From Lemma 3 we know that $G(w, w_{ab}^{(t)})$ is an auxiliary function for $F_{w_{ab}}$, and from Lemma 4 we know that $G(v, v_{ab}^{(t)})$ is an auxiliary function for $F_{v_{ab}}$. According to Lemma 2, by solving $w^{(t+1)} = \arg\min_w G(w, w_{ab}^{(t)})$, We obtain

$$w_{ab}^{(t+1)} = w_{ab}^{(t)} \frac{(\mathbf{K}\mathbf{V}^T + \lambda \sum_{i=1}^N \mathbf{X}^T\mathbf{x}_i\mathbf{1}^T\mathbf{D}_i)_{ab}}{(\mathbf{K}\mathbf{W}\mathbf{V}\mathbf{V}^T + \lambda \sum_{i=1}^N \mathbf{K}\mathbf{W}\mathbf{D}_i)_{ab}} \quad (9)$$

and by solving $v^{(t+1)} = \arg\min_z G(v, v_{ab}^{(t)})$, We obtain

$$v_{ab}^{(t+1)} = v_{ab}^{(t)} \frac{2(\lambda+1)(\mathbf{W}^T\mathbf{K})_{ab}}{(2\mathbf{W}^T\mathbf{K}\mathbf{W}\mathbf{V} + \lambda\mathbf{A} + \lambda\mathbf{B})_{ab}} \quad (10)$$

which are exactly the same updates as in Eq. (4) and (5). Therefore, the objective function $\mathcal{O}$ in Eq. (2) is nonincreasing under these updates. □

4 Experiments

In this section, we show the data clustering performance of the proposed method, and compare the results with four other related methods using the same data set. The algorithms that we evaluated are listed below:

- Traditional K-means (**Kmeans**).
- Nonnegative Matrix Factorization (**NMF**) [Lee and Seung, 2001].
- Concept Factorization (**CF**) [Xu and Gong, 2004].
- Non-negative Matrix Factorization with Sparseness Constraints (**SparseNMF**) [Hoyer, 2004].
- Our proposed Locality-constrained concept Factorization (**LCF**).

We use two metrics to evaluate the clustering performance. One metric is accuracy (AC), which is used to measure the percentage of correct labels obtained. 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. The definitions of these two metrics can be found in [Xu and Gong, 2004]. The performance is evaluated by comparing the cluster label of each sample with the label provided by the data set.

Table 1: Statistics of the two data sets

dataset	size(N)	dimensionality(M)	of classes(K)
Yale	165	1024	15
ORL	400	1024	40

The evaluations were conducted for the cluster numbers ranging from 2 to 10. For each given cluster number k, we randomly chose k clusters and ran the test 10 times, and the final scores were obtained by calculating the average and variance over the 10 test runs. Since the evaluated methods whose clustering results depend on the initialization, each test run consists of 10 sub-runs from which we chose the best result to report.

In the experiments, the parameters were set to be the values that each algorithm can achieve its best results. For our LCF algorithm, the regularization parameter is set to be $\lambda = 0.3$.

4.1 Data Sets

The experiments are conducted on two data sets. One is the Yale Database, and the other is Cambridge ORL face database. The important statistics of these data sets are described below (also summarized in Table 1):

- The Yale database contains 165 gray scale images of 15 individuals. All images demonstrate variations in lighting condition (left-light, center-light,right-light), facial expression (normal, happy, sad, sleepy, surprised and wink), and with/without glasses.
- The ORL database contains ten different images of each of 40 distinct subjects, thus 400 images in total. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

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 of 32×32 pixels with 256 gray levels per pixel.

4.2 Clustering Results

Fig. 1 and 2 show the plots of accuracy and normalized mutual information versus the number of clusters for different algorithms on the Yale dataset. As can be seen, our proposed LCF algorithm consistently outperforms all the other algorithms. The detailed clustering results are shown in Table 2. The last row shows the average accuracy (normalized mutual information) over k. Comparing to the best algorithm other than our proposed LCF i.e. SparseNMF, our algorithm LCF achieves 4.41% improvement in accuracy and 5.69% improvement in normalized mutual information.

Table 2: Clustering Results Comparison on the Yale database

k	Kmeans	NMF	CF	SparseNMF	LCF
	Accuracy(%)				
2	78.18 ± 16.86	73.18 ± 18.00	71.82 ± 20.21	79.09 ± 17.51	$\mathbf{85.00 \pm 17.01}$
3	59.09 ± 12.14	62.73 ± 11.42	58.79 ± 9.70	63.33 ± 9.91	$\mathbf{76.36 \pm 17.29}$
4	51.36 ± 7.20	56.14 ± 5.75	53.18 ± 9.55	56.82 ± 7.87	$\mathbf{58.64 \pm 9.52}$
5	50.36 ± 3.36	52.55 ± 5.30	52.36 ± 5.74	54.18 ± 6.94	$\mathbf{58.18 \pm 7.18}$
6	46.52 ± 6.85	49.55 ± 8.91	48.18 ± 8.10	51.06 ± 8.63	$\mathbf{52.42 \pm 10.74}$
7	43.25 ± 6.42	46.49 ± 5.97	47.14 ± 5.92	47.66 ± 5.29	$\mathbf{49.48 \pm 4.84}$
8	44.20 ± 3.78	45.00 ± 3.56	46.36 ± 2.91	47.05 ± 2.84	$\mathbf{49.43 \pm 5.69}$
9	43.74 ± 7.03	43.23 ± 4.33	44.65 ± 6.32	42.32 ± 4.50	$\mathbf{47.68 \pm 5.71}$
10	39.45 ± 4.49	41.36 ± 5.77	43.91 ± 5.19	41.00 ± 3.34	$\mathbf{45.00 \pm 7.50}$
Avg.	50.68 ± 7.57	52.25 ± 7.67	51.82 ± 8.18	53.61 ± 7.43	$\mathbf{58.02 \pm 9.50}$
	Normalized Mutual Information(%)				
2	40.58 ± 39.97	32.53 ± 38.16	34.66 ± 43.79	42.85 ± 40.46	$\mathbf{56.78 \pm 39.45}$
3	38.49 ± 20.50	34.28 ± 16.27	31.99 ± 20.51	32.70 ± 15.86	$\mathbf{54.24 \pm 30.02}$
4	31.00 ± 6.92	32.78 ± 9.04	31.48 ± 11.96	35.88 ± 10.04	$\mathbf{36.13 \pm 9.92}$
5	39.08 ± 7.99	40.11 ± 8.46	38.13 ± 8.51	40.21 ± 7.76	$\mathbf{42.91 \pm 10.41}$
6	40.03 ± 11.26	37.74 ± 10.81	38.99 ± 11.40	39.83 ± 11.50	$\mathbf{42.43 \pm 11.94}$
7	36.39 ± 6.24	36.89 ± 6.36	40.46 ± 5.32	40.73 ± 5.37	$\mathbf{42.41 \pm 4.47}$
8	40.78 ± 4.09	41.07 ± 3.64	40.48 ± 2.26	40.90 ± 3.08	$\mathbf{43.78 \pm 3.70}$
9	41.58 ± 7.28	40.94 ± 5.24	40.90 ± 7.06	40.86 ± 3.54	$\mathbf{43.12 \pm 5.09}$
10	39.04 ± 5.58	39.83 ± 4.59	41.85 ± 4.96	41.07 ± 2.63	$\mathbf{44.45 \pm 7.32}$
Avg.	38.55 ± 12.20	37.35 ± 11.40	37.66 ± 12.86	39.45 ± 11.14	$\mathbf{45.14 \pm 13.59}$

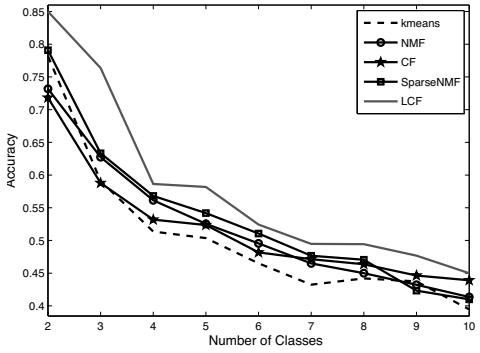


Figure 1: Accuracy on the Yale database

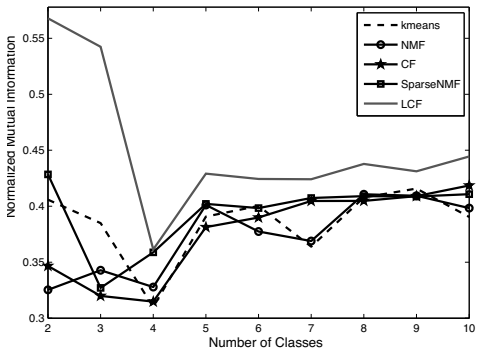


Figure 2: Mutual Information on the Yale database

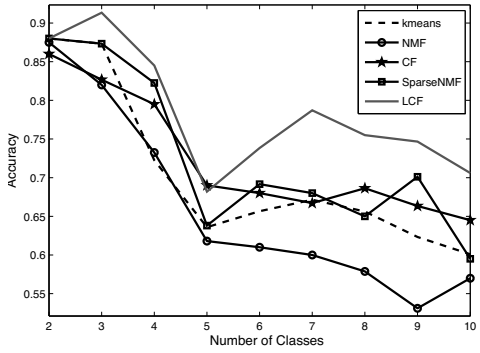


Figure 3: Accuracy on the ORL database

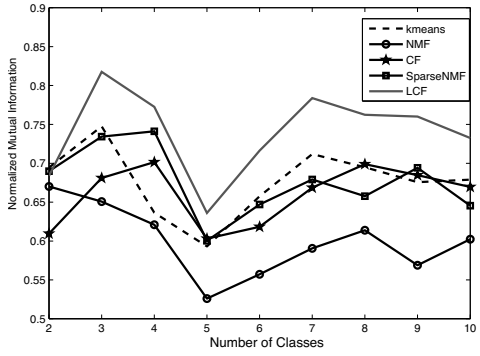


Figure 4: Mutual Information on the ORL database

Fig. 3 and Fig. 4 show the graphical clustering results for the ORL data set. LCF obtains the best result for most of the cases. SparseNMF fails to consider the locality condition, and in some cases performs even worse than Kmeans. Table 3 shows the detailed clustering accuracy and normalized mutual information. Comparing to the best algorithm other than our proposed LCF algorithm, i.e., SparseNMF, LCF achieves 5.8% improvement in accuracy. For normalized mutual information, LCF achieves 6.4% improvement.

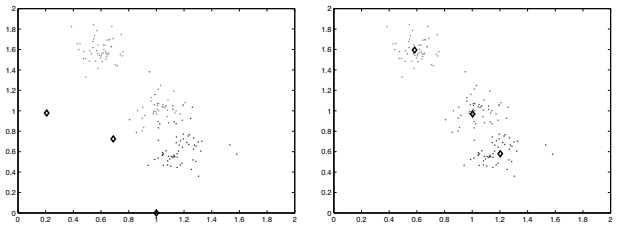
4.3 Learning the Overcomplete Basis

Usually, matrix factorization methods are used for dimension reduction in many applications. However, [Lee *et al.*, 2007] shows that in some cases, it is desirable to learn the overcomplete basis. Here we evaluate the performance of our al-

Table 3: Clustering Results Comparison on the ORL database

k	Kmeans	NMF	CF	SparseNMF	LCF
	Accuracy(%)				
2	88.00 ± 17.64	87.50 ± 17.64	86.00 ± 16.09	88.00 ± 17.49	$\mathbf{88.00 \pm 17.78}$
3	87.33 ± 10.73	82.00 ± 11.27	82.67 ± 12.36	87.33 ± 10.93	$\mathbf{91.33 \pm 9.80}$
4	72.25 ± 13.53	73.25 ± 8.37	79.50 ± 9.34	82.25 ± 7.94	$\mathbf{84.50 \pm 9.92}$
5	63.60 ± 11.62	61.80 ± 7.40	$\mathbf{69.00 \pm 8.26}$	63.80 ± 7.77	68.20 ± 8.27
6	65.67 ± 7.04	61.00 ± 7.57	68.00 ± 7.74	69.17 ± 5.83	$\mathbf{73.83 \pm 6.99}$
7	67.14 ± 13.19	60.00 ± 6.23	66.71 ± 7.06	68.00 ± 6.07	$\mathbf{78.71 \pm 11.00}$
8	65.63 ± 7.57	57.88 ± 5.76	68.63 ± 6.29	65.00 ± 6.98	$\mathbf{75.50 \pm 6.20}$
9	62.33 ± 7.44	53.11 ± 4.55	66.33 ± 7.32	70.11 ± 6.35	$\mathbf{74.67 \pm 8.18}$
10	60.10 ± 7.23	57.00 ± 6.15	64.50 ± 6.17	59.50 ± 7.79	$\mathbf{70.60 \pm 6.62}$
Avg.	70.23 ± 10.66	65.95 ± 8.33	72.37 ± 8.96	72.57 ± 8.57	$\mathbf{78.37 \pm 9.42}$
	Normalized Mutual Information(%)				
2	$\mathbf{69.45 \pm 42.12}$	67.02 ± 40.95	60.95 ± 40.11	68.99 ± 42.71	68.38 ± 41.04
3	74.75 ± 17.48	65.08 ± 15.94	68.11 ± 18.03	73.43 ± 17.50	$\mathbf{81.76 \pm 17.12}$
4	63.64 ± 16.27	62.09 ± 9.01	70.15 ± 11.54	74.12 ± 9.17	$\mathbf{77.26 \pm 13.09}$
5	59.30 ± 12.75	52.59 ± 9.10	60.30 ± 8.93	60.03 ± 9.42	$\mathbf{63.61 \pm 10.83}$
6	65.75 ± 6.22	55.72 ± 5.45	61.83 ± 7.03	64.70 ± 7.13	$\mathbf{71.63 \pm 8.32}$
7	71.19 ± 10.56	59.06 ± 6.53	66.86 ± 7.62	67.92 ± 7.45	$\mathbf{78.39 \pm 10.46}$
8	69.48 ± 6.22	61.38 ± 4.54	69.89 ± 4.74	65.77 ± 5.71	$\mathbf{76.23 \pm 5.03}$
9	67.56 ± 7.50	56.88 ± 5.32	68.47 ± 5.68	69.42 ± 5.19	$\mathbf{76.01 \pm 7.92}$
10	67.89 ± 6.54	60.23 ± 6.54	66.96 ± 5.40	64.55 ± 6.59	$\mathbf{73.25 \pm 6.10}$
Avg.	67.67 ± 13.96	60.01 ± 11.49	65.95 ± 12.12	67.66 ± 12.32	$\mathbf{74.06 \pm 13.32}$

gorithm in this aspect. To show the performance for learing the overcomplete basis of our proposed algorithm, 150 data points from mixture of three Gaussians in a 2- dimensional space are randomly generated. NMF and LCF are conducted to obtain three basis vectors and cluster these data points into 3 clusters. Fig 5 shows that our LCF obtains better results than NMF. The bases obtained by NMF are far away from the original points. However, since we add a locality constraint, the three bases obtained by LCF exactly reside in the cluster centers, which leads to a better data representation.



(a) NMF clustering results (b) LCF clustering results

Figure 5: Experiments on learning the overcomplete basis. The black diamonds are the bases learned by each algorithm.

5 Conclusions

In this paper we proposed a novel matrix factorization method, called Locality-constrained Concept Factorization (LCF). This method enforces a locality constraint onto the traditional concept factorization. By requiring the concepts (basis vectors) to be as close to the original data points as possible, each data can be represented by a linear combination of only a few nearby basis concepts, thus achieving sparsity and locality at the same time. The experimental results on two standard face databases have demonstrated the effectiveness of our approach over other matrix factorization techniques, especially for the data clustering applications.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under grant 2009QNA5024.

References

[Cai *et al.*, 2011] Deng Cai, Xiaofei He, and Jiawei Han. Locally consistent concept factorization for document clustering. *IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2010.165*, 2011.

[He and Niyogi, 2003] Xiaofei He and Partha Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems 16*. 2003.

[He *et al.*, 2005] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):328–340, 2005.

[Hoyer, 2004] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, December 2004.

[Lee and Seung, 1999] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. In *Nature*, pages 788–791, 1999.

[Lee and Seung, 2001] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.

[Lee *et al.*, 2007] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2007.

[Xu and Gong, 2004] Wei Xu and Yihong Gong. Document clustering by concept factorization. In *Proc. 2004 Annual ACM SIGIR Conference*, 2004.

[Yu *et al.*, 2009] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems*, 2009.