Sparse Concept Coding for Visual Analysis

Deng Cai Hujun Bao Xiaofei He
The State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
388 Yu Hang Tang Rd., Hangzhou, Zhejiang, China 310058
{dengcai, bao, xiaofeihe}@cad.zju.edu.cn

Abstract

We consider the problem of image representation for visual analysis. When representing images as vectors, the feature space is of very high dimensionality, which makes it difficult for applying statistical techniques for visual analysis. To tackle this problem, matrix factorization techniques, such as Singular Vector Decomposition (SVD) and Non-negative Matrix Factorization (NMF), received an increasing amount of interest in recent years. Matrix factorization is an unsupervised learning technique, which finds a basis set capturing high-level semantics in the data and learns coordinates in terms of the basis set. However, the representations obtained by them are highly dense and can not capture the intrinsic geometric structure in the data. In this paper, we propose a novel method, called Sparse Concept Coding *(SCC), for image representation and analysis. Inspired from the recent developments on manifold learning and sparse coding, SCC provides a sparse representation which can capture the intrinsic geometric structure of the image space. Extensive experimental results on image clustering have shown that the proposed approach provides a better representation with respect to the semantic structure.*

1. Introduction

Image representation is the fundamental problem in image processing. Researchers have long sought effective and efficient representations of images. For a give image database, we may have thousands of distinct features. However, the degree of the freedom of each image could be far less. Instead of the original feature space, one might hope to find a hidden semantic "concept" space to represent the images. The dimensionality of this "concept" space will be much smaller than the feature space. To achieve this goal, matrix factorization based approaches have attracted considerable attention in the last decades [1, 5, 10].

Given an image data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, each column of $\mathbf{X}$ corresponding to an image, the matrix factorization methods find two matrices $\mathbf{U} \in \mathbb{R}^{m\times k}$ and $\mathbf{A} \in \mathbb{R}^{k\times n}$ whose product can well approximate $\mathbf{X}$. Each column vector of $\mathbf{U}$ can be regarded as a basis vector corresponding to a certain semantic concept and each column vector of $\mathbf{A}$ is the representation of an image in this concept space.

One of the most well know matrix factorization methods is Latent Semantic Analysis (LSA) [5], which is fundamentally based on Singular Value Decomposition. LSA is optimal in the sense of reconstruction error and thus optimal for data representation when Euclidean structure is concerned. Another popular matrix factorization method is Non-negative Matrix Factorization (NMF) [10], which requires the factorization matrices (both $\mathbf{U}$ and $\mathbf{A}$) are non-negative. The non-negative constraints only allow additive combinations among different basis vectors and it is believed that NMF can learn a *parts-based* representation [10]. The effectiveness of NMF has been demonstrated in many image analysis tasks [1, 12] . Inspired by biological visual systems, people has been arguing sparse features of data points are useful for learning [18, 22, 24]. Sparse Coding (SC) [11, 16] is a recently popular matrix factorization method which requires the representation matrix $\mathbf{A}$ to be sparse. The sparseness of $\mathbf{A}$ indicates that each image will only relate to several concepts (with non-zero coefficients to the corresponding basis vectors). All these popular matrix factorization methods only consider the Euclidean structure of the image space.

Recent studies have shown that human generated image data is probably sampled from a submanifold of the ambient Euclidean space [2, 19, 21]. In fact, the image data cannot possibly "fill up" the high dimensional Euclidean space uniformly. Therefore, the intrinsic manifold structure needs to be considered while performing the matrix factorization.

Motivated from recent progresses in matrix factorization (sparse coding) and manifold learning, in this paper we propose a novel matrix factorization method, called *Sparse Concept Coding* (SCC), for image representation. SCC is a a two-step approach including basis learning and sparse representation learning. In the first step, SCC learns the basis by exploiting the intrinsic geometric structure of the data. By applying the spectral analysis on the nearest neighbor graph, SCC encodes the semantic structure in the basis vectors. In the second step, SCC uses the LASSO [9] to learn a sparse representation with respect to the learned basis for each image. As a result, SCC can have more discriminating power than the traditional matrix factorization approaches which only consider the Euclidean structure of the data.

The rest of the paper is organized as follows: in Section 2, we provide a brief review of matrix factorization. Our Sparse Concept Coding method is introduced in Section 3. The experimental results are presented in Section 4. Finally, we provide the concluding remarks in Section 5.

2. Background

Given a data set with high dimensionality, matrix factorization is a common approach to "compress" the data by finding a set of *basis* vectors and the *representation* with respect to the basis for each data point. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$ be the data matrix, matrix factorization can be mathematically defined as finding two matrices $\mathbf{U} \in \mathbb{R}^{m\times k}$ and $\mathbf{A} \in \mathbb{R}^{k\times n}$ whose product can best approximate $\mathbf{X}$:

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

Each column of $\mathbf{U}$ can be regarded as a basis vector which captures the higher-level features in the data and each column of $\mathbf{A}$ is the k-dimensional representation of the original inputs with respect to the new basis. From this sense, matrix factorization can also be regarded as a dimensionality reduction method since it reduces the dimension from m to k.

A common way to measure the approximation is by Frobenius norm of a matrix $\|\cdot\|$. Thus, the matrix factorization can be defined as the optimization problem as follows:

$$\min_{\mathbf{U},\mathbf{A}} \|\mathbf{X} - \mathbf{UA}\|^2 \tag{1}$$

Various matrix factorization algorithms add different constraints on the above optimization problem based on different goals.

Latent Semantic Analysis (LSA) is one of the most popular matrix factorization algorithms for image analysis [5]. It adds rank constraint ($rank(\mathbf{UA}) \leq k$) on the optimization problem in Eq. (1). LSA is fundamentally based on Singular Value Decomposition (SVD) [8]. Suppose the rank of $\mathbf{X}$ is r, the SVD decomposition of $\mathbf{X}$ is as follows:

$$\mathbf{X} = \mathbf{U\Sigma V}^T,$$

where $\mathbf{\Sigma} = diag(\sigma_1, \cdots, \sigma_r)$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are the singular values of $\mathbf{X}$, $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_r]$ and $\mathbf{u}_i$'s are called left singular vectors, $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_r]$ and $\mathbf{v}_i$'s are called right singular vectors. We have $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$. LSI uses the first k columns of $\mathbf{U}$ and $\mathbf{V}$. It can be proven that $\mathbf{U} \in \mathbb{R}^{m\times k}$ and $\mathbf{A} = \mathbf{\Sigma V}^T \in \mathbb{R}^{k\times n}$ is the optimal solution of the optimization problem (1) under the constraint $rank(\mathbf{UA}) \leq k$ [8].

Another popular matrix factorization algorithm is Non-negative Matrix Factorization (NMF) [10], which focuses on the analysis of data matrices whose elements are non-negative. NMF adds the non-negative constraint on both $\mathbf{U}$ and $\mathbf{A}$ in the optimization problem (1). The non-negative constraints on $\mathbf{U}$ and $\mathbf{A}$ only allow additive combinations among different basis vectors. For this reason, it is believed that NMF can learn a *parts-based* representation [10].

The representation matrix $\mathbf{A}$ learned in the above two methods is usually dense. Since each basis vector (column vector of $\mathbf{U}$) can be regarded as a concept, the denseness of $\mathbf{A}$ indicates that each images is a combination of *all* the concepts. This is contrary to our common knowledge since most of the images only include several semantic concepts. Sparse Coding (SC) [11, 16] is a recently popular matrix factorization method trying to solve this issue. Sparse coding adds the sparse constraint on $\mathbf{A}$, more specifically, on each column of $\mathbf{A}$, in the optimization problem (1). In this way, SC can learn a sparse representation. SC has several advantages for data representation. First, it yields sparse representations such that each data point is represented as a linear combination of a small number of basis vectors. Thus, the data points can be interpreted in a more elegant way. Second, sparse representations naturally make for an indexing scheme that would allow quick retrieval. Third, the sparse representation can be overcomplete, which offers a wide range of generating elements. Potentially, the wide range allows more flexibility in signal representation and more effectiveness at tasks like signal extraction and data compression. Finally, there is considerable evidence that biological vision adopts sparse representations in early visual areas [15]. The sparse coding approach is fundamentally different from those sparse subspace learning methods, *e.g.*, SparsePCA [25] and SparseLDA [14]. Instead of learning a sparse $\mathbf{A}$, the sparse subspace learning methods [14, 25] learn a sparse $\mathbf{U}$. The low dimensional representation matrix learned by these sparse subspace learning methods is still dense.

3. Sparse Concept Coding

All the three matrix factorization methods discussed above only consider the Euclidean structure of the image space. However, recent studies have shown that human generated image data is probably sampled from a submanifold of the ambient Euclidean space [2, 19, 21]. In fact, the human generated image data cannot possibly "fill up" the high dimensional Euclidean space uniformly. Therefore, the intrinsic manifold structure needs to be considered while learning the basis.

Combining with the idea behind the sparse coding approach, we believe that a good matrix factorization should consider the following two aspects:

- The learned basis should capture the intrinsic geometric structure of the data;
- And the representation under the new basis should be sparse.

Considering the above two aspects, we propose a novel matrix factorization method called Sparse Concept Coding (SCC) in this section. SCC is a two-step approach for matrix factorization. The first step is basis (concept) learning. By exploring the intrinsic geometric structure of the data, SCC can learn a basis which is optimal for concept expression. The second step is representation learning. SCC uses LASSO [9]which is a L1-regularized regression model, to learn a sparse representation for each image.

The goal of SCC is trying to solve the following optimization problem

$$\min_{\mathbf{U},\mathbf{A}} \|\mathbf{X}-\mathbf{U}\mathbf{A}\|^2 \tag{2}$$

where the basis $\mathbf{U}$ should capture the intrinsic geometric structure of $\mathbf{X}$ and each column vector of $\mathbf{A}$ should be sparse. In the next subsection, we begin with a discussion on how to learn the basis $\mathbf{U}$.

3.1. Basis Learning

Recently, various researchers [2, 19, 21] have considered the case when the data is drawn from sampling a probability distribution that has support on or near to a *submanifold* of the ambient space. In order to detect the underlying manifold structure, many *manifold learning* algorithms have been proposed [2, 19, 21]. There algorithms construct a nearest neighbor graph to model the local geometric structure and perform spectral analysis on the graph weight matrix. This way, these manifold learning algorithms can "unfold" the data manifold and provide the "flat" embedding for the data points.

Consider a graph with N vertices where each vertex corresponds to an image in the data set. The edge weight matrix $\mathbf{W}$ is usually defined as follows:

$$\mathbf{W}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in N_p(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_p(\mathbf{x}_s) \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $N_p(\mathbf{x}_i)$ denotes the set of p nearest neighbors of $\mathbf{x}_i$. Define a diagonal matrix $\mathbf{D}$ whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, we can compute the graph Lapalcian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [4].

The "flat" embedding for the data points which "unfold" the data manifold can be found by solving the following generalized eigen-problem [2]:

$$\mathbf{L}\mathbf{y} = \lambda\mathbf{D}\mathbf{y} \tag{4}$$

Let $\mathbf{Y} = [\mathbf{y}_1, \cdots, \mathbf{y}_k]$, $\mathbf{y}_i$'s are the eigenvectors of the above generalized eigen-problem with respect to the smallest eigenvalue. Each row of $\mathbf{Y}$ is the "flat" embedding for each data point.

Considering the manifold structure, our SCC tries to learn the basis $\mathbf{U}$ which can best fit $\mathbf{Y}$, which can be achieved by solving the optimization problem as follows:

$$\min_{\mathbf{U}} \|\mathbf{Y} - \mathbf{X}^T\mathbf{U}\|^2 + \alpha\|\mathbf{U}\|^2 \tag{5}$$

where the term $\alpha\|\mathbf{U}\|^2$ is to regularize the model to avoid over-fitting [9] and α is the regularization parameter. In statistics, this model is called *Ridge Regression* [9].

By taking the derivative of Eq. (5) with respect to $\mathbf{U}$ and setting it to zero, we can obtain the optimal solution:

$$\mathbf{U}^* = \left(\mathbf{X}\mathbf{X}^T + \alpha\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y} \tag{6}$$

where $\mathbf{I}$ is the identity matrix.

If the image data are of very high dimensionality, it is computational expensive to inverse the matrix $\mathbf{X}\mathbf{X}^T + \alpha\mathbf{I}$. In reality, we can use some iterative algorithms to directly solve the regression problem in Eq. (5) (*e.g.*, LSQR [17]).

Let s denote the average number of non-zero entries in each image (each column of $\mathbf{X}$), $s \le m$. Compute $\mathbf{U}^*$ by solving Eq. (5) with LSQR method needs $O(kns)$ time [17]. Considering $\mathbf{W}$ is a p-nearest neighbor graph, we need $O(n^2s + n^2p)$ time to compute it. And we need $O(knp)$ time to compute the smallest k eigenvectors of the eigen-problem (4) [20]. Considering $k \ll n$, our SCC algorithm needs $O(n^2s + n^2p)$ to compute the basis $\mathbf{U}$.

3.2. Sparse Representation Learning

After we obtain the basis $\mathbf{U}$, the representation $\mathbf{A}$ can be computed column by column independently through the following minimization problem.

$$\min_{\mathbf{a}_i} \|\mathbf{x}_i - \mathbf{U}\mathbf{a}_i\|^2 + \beta|\mathbf{a}_i| \tag{7}$$

where $\mathbf{x}_i$ and $\mathbf{a}_i$ are the i-th columns of $\mathbf{X}$ and $\mathbf{A}$, respectively. $|\mathbf{a}_i|$ denote the L1-norm of $\mathbf{a}_i$ and this L1-norm regularization term is added to ensure the sparseness of $\mathbf{a}_i$. In statistics, the above L1-regularized regression problem is called LASSO [9].

The optimization problem in Eq. (7) has the following equivalent formulation:

$$\begin{aligned} &\min_{\mathbf{a}_i} \|\mathbf{x}_i - \mathbf{U}\mathbf{a}_i\|^2 \\ &s.t. \quad |\mathbf{a}_i| \le \gamma \end{aligned} \tag{8}$$

The Least Angel Regression (LARs) algorithm [6] can be used to solve the optimization problem in Eq. (8). Instead of setting the parameter γ, LARs provides another choice to control the sparseness of $\mathbf{a}_i$ by specifying the cardinality (the number of non-zero entries) of $\mathbf{a}_i$. LARs can compute the entire solution path (the solutions with all the possible cardinality on $\mathbf{a}_i$) of problem (8). in $O(k^3 + mk^2)$.

Overall, the time complexity of our SCC algorithm is

$$O(n^2 s + n^2 p + k^3 + mk^2).$$

Comparing with the traditional sparse coding approaches [11, 16], SCC has its own advantages as follows:

- Traditional sparse coding approaches consider the factorization on Euclidean space, whereas SCC learns the basis by exploiting the intrinsic geometric of the data. The basis learned by SCC can have more discriminating power.
- Traditional sparse coding approaches iteratively find the **U** and **V**, which is very time consuming. Our SCC approach only needs to solve a sparse eigen-problem and two regression problems, which is very efficient.

4. Experimental Results

Previous studies show that matrix factorization and sparse coding approaches are very powerful on clustering [23]. It can achieve similar or better performance than most of the state-of-the-art clustering algorithms. In this section, we also evaluate our SCC algorithm on clustering problems.

The COIL20 image library[1] is used in our experiment. COIL20 contains 32×32 gray scale images of 20 objects viewed from varying angles and each object has 72 images.

4.1. Evaluation Metric

The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. Two metrics, the accuracy (AC) and the normalized mutual information metric (NMI) are used to measure the clustering performance [3]. Given a data point $\mathbf{x}_i$, let r_i and s_i be the obtained cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

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

where N is the total number of samples and $\delta(x, y)$ is the delta function that equals 1 if $x = y$ and equals 0 otherwise, and map(r_i) is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [13].

Let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric MI(C, C') is defined as follows:

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

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

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

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

4.2. Compared Algorithms

In our experiments, we compared five matrix factorization algorithms and a baseline method as follows:

- Baseline method which simply performs clustering in the original feature space.
- Latent Semantic Analysis [5] which is based on Singular Value Decomposition (SVD).
- Nonnegative Matrix Factorization (NMF) [10].
- Sparse Coding (SC) approach [16]. We use an efficient space coding algorithm which is introduced in [11].
- Laplacian Sparse Coding (LapSC) approach [7], which is an manifold (geometric) extension of traditional SC by using a graph regularizer.
- Space Concept Coding (SCC), which is the new approach proposed in this paper.

After the matrix factorization, we have a low dimensional representation of each image. The clustering is then performed in this low dimensional space.

4.3. Clustering Results

We use k-means as the clustering algorithm. k-means can be performed on the original feature space (baseline) or the "concept" space (matrix factorization methods). In order to randomize the experiments, we evaluate the clustering performance with different number of clusters (k = 4, 6, $\cdots$, 18, 20). For each given cluster number k (except 20), 20 tests were conducted on different randomly chosen classes, and the average performance as well as the standard deviation was computed over these 20 tests. In each

[1] http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Table 1. Clustering performance

K	Accuracy (%)						Normalized Mutual Information (%)					
	Baseline	SVD	NMF	SC	LapSC	SCC	Baseline	SVD	NMF	SC	LapSC	SCC
4	83.0±15.2	83.1±15.0	81.0±14.2	78.5±15.4	79.6±16.7	**90.5±13.9**	74.6±18.3	74.4±18.2	71.8±18.4	70.8±18.0	73.3±19.1	**87.6±17.4**
6	74.5±10.3	75.5±12.2	74.3±10.1	73.0±11.2	76.3±11.5	**92.4±6.1**	73.2±11.4	73.1±12.1	71.9±11.6	72.0±11.7	76.6±11.0	**90.2±7.8**
8	68.6±5.7	70.4±9.3	69.3±8.6	68.2±10.3	72.0±9.1	**87.5±11.0**	71.8±6.8	72.8±8.3	71.0±7.4	70.64±9.0	76.3±7.1	**88.7±8.3**
10	69.6±8.0	70.8±7.2	69.4±7.6	72.6±9.0	76.8±7.4	**86.1±7.4**	75.0±6.2	75.1±5.2	73.9±5.7	76.56±6.0	81.2±5.4	**89.3±4.2**
12	65.0±6.8	64.3±4.6	69.0±6.3	67.0±7.5	72.2±5.2	**83.0±7.1**	73.1±5.6	72.5±4.6	73.3±5.5	73.86±5.5	79.5±4.6	**87.6±5.2**
14	64.0±4.9	67.3±6.2	67.6±5.6	66.5±5.3	72.7±6.4	**81.3±4.4**	73.3±4.2	74.9±4.9	73.8±4.6	74.38±3.5	80.3±4.4	**87.5±3.4**
16	64.0±4.9	64.1±4.9	66.0±6.0	65.1±4.2	71.8±3.9	**80.0±4.9**	74.6±3.1	74.5±2.7	73.4±4.2	74.82±3.1	80.5±2.6	**87.2±2.8**
18	62.7±4.7	62.3±4.3	62.8±3.7	61.6±5.4	69.4±3.5	**77.4±3.6**	73.7±2.6	73.9±2.5	72.4±2.4	73.12±2.3	79.4±2.2	**86.3±1.9**
20	63.7	64.3	60.5	64.4	71.1	**83.4**	73.4	74.5	72.5	73.62	79.8	**88.3**
Avg	68.3	69.1	68.9	68.5	73.5	**84.6**	73.6	74.0	72.7	73.3	78.5	**88.1**

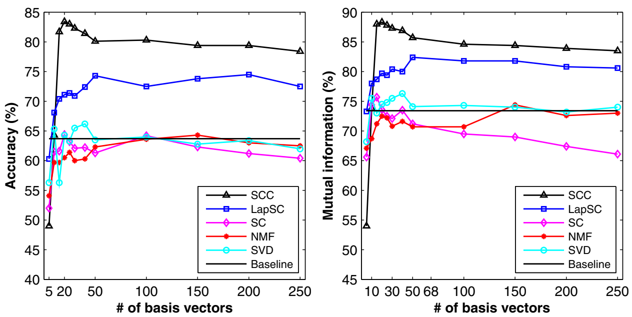

Figure 1. The performance of five matrix factorization methods vs. the number of basis vectors

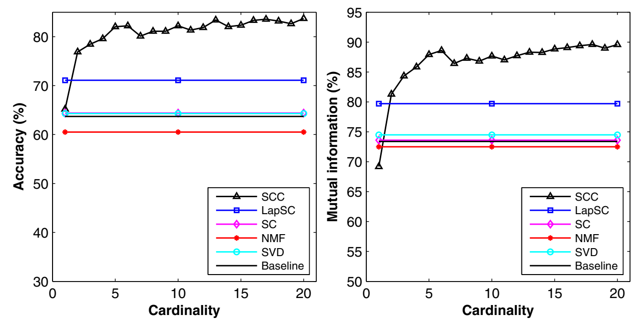

Figure 2. The performance of SCC vs. the cardinality of the low dimensional representation.

test, k-means algorithm was applied 10 times with different random starting points and the best result in terms of the objective function of k-means was recorded.

For matrix factorization methods, how to determine the number of basis vectors is still an open problem. In this experiment, we report the performances of all the matrix factorization methods with the number of basis vectors equal to the number of clusters. There are two parameters in the basis learning phase of SCC: the number of nearest neighbors p and the ridge regularization parameter α. These two parameters were empirically set as 5 and 0.1, respectively. In the sparse representation learning phase of SCC, the parameter is the cardinality (the number of non-zero entries) of each low dimensional vector. We empirically let the cardinality equal to half of the number of the basis vectors.

Table (1) shows the clustering results of the five compared methods. In all the cases on both two data sets, our SCC method significantly outperforms its competitors. By simply using k-means on the low dimensional sparse representation, SCC achieves very impressive clustering performance. LapSC is the second best method. By incorporating a graph regularizer, LapSC gains significant improvement over the traditional sparse coding approach (SC). When the number of basis vectors is fixed as the number of clusters, the SVD method fails to gain significant improvement over the baseline approach.

Figure (1) shows how the performances of the five matrix factorization methods vary with the number of learned basis vectors. We simply reported the clustering results using the entire data set. As we can see, the performances of all the four methods drastically decrease as the number of basis vectors decreases (smaller than the number of classes). This result is expected since for a data set containing c class, there is at least c concepts. Thus, the "concept" space is with at least c dimensions. It is interesting to see that SCC reaches its highest performance with exactly c basis vectors, which might suggests that SCC can really capture the concept structure of the image space. As the number of basis vectors increases, the performance of SCC slightly decreases. In the entire scope ($c \sim 250$), the performances of all the five methods are relatively stable and our SCC significantly outperforms the other methods.

Figure (2) shows how the performance of SCC varies with the cardinality parameter. We have 20 basis vectors and each image is represented as a 20-dimensional vector in the SCC "concept" space. The cardinality denotes the number of non-zero entries of each 20-dimensional vector. As we can see, the performance of SCC is very stable with respect to the cardinality as long as the cardinality is not too small. The smallest value for cardinality which maintains the high performance is 5. In other words, despite the high dimensionality of the original feature space (1024), each image can be represented as a 20-dimensional *sparse* vector in the SCC "concept" space. And each sparse vector has only 5 non-zero entries. This significant property differentiates SCC from SVD and NMF. The low dimensional representations found by SVD and NMF are dense, which suggests that every image contains all the concepts. This is contrary to our common knowledge since most of the images only contain several concepts.

5. Conclusions

In this paper, we have presented a novel matrix factorization method for image representation called Sparse Concept Coding (SCC). SCC is a two-step approach including a basis learning stage and a sparse representation learning stage. In the first stage, SCC learns the basis by exploiting the intrinsic geometric structure of the data. By applying the spectral analysis on the nearest neighbor graph, SCC encodes the semantic structure in the basis vectors. In the second stage, SCC uses the LASSO to learn a sparse representation with respect to the learned basis for each image. As a result, SCC can have more discriminating power than the traditional matrix factorization approaches (SVD, NMF and SC) which only consider the Euclidean structure of the data. Experimental results on image clustering show that SCC provides better representation in the sense of semantic structure.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants 60905001 and 60875044, National Basic Research Program of China (973 Program) under Grant 2011CB302206 and Fundamental Research Funds for the Central Universities.

References

[1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1475–1490, 2004.

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

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

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

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

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

[7] S. Gao, I. Tsang, L. Chia, and P. Zhao. Local features are not lonely–Laplacian sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, pages 3555–3561, 2010.

[8] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

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

[11] H. Lee, A. Battle, R. Raina, , and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2006.

[12] S. Z. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pages 207–212, 2001.

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

[14] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 641–648, 2006.

[15] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[16] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.

[17] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proc. 2007 Int. Conf. Machine Learning (ICML'07)*, 2007.

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

[20] G. W. Stewart. *Matrix Algorithms Volume II: Eigensystems*. SIAM, 2001.

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

[22] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Learning Locality-constrained Linear Coding for Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, 2010.

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

[24] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1794–1801, 2009.

[25] H. Zou, T. Hastie, and R. Tibshirani. Sparse principle component analysis. *JCGS*, 15(2):262–286, 2006.