Semi-Supervised Sparse Coding

Jim Jing-Yan Wang

University at Buffalo, The State University of New York, Buffalo, NY 14203, USA

Abstract

Sparse coding approximate the data sample as a sparse linear combination of some basic codewords, and use the sparse codes as new presentations. In this paper, we investigate learning discriminative sparse codes by sparse coding in a semi-supervised scene, where only a few training samples are labeled. By using the manifold structure spanned by the data set of both labeled and unlabeled samples, and the constrains provided by the labels of labeled samples, we learn the variable class labels for all the samples. Furthermore, to improve the discriminative ability of the learned sparse codes, we assume that the class labels could be predicted from the sparse codes directly using a linear classifier. By solving the codebook, sparse codes, class labels and the classifier parameters simultaneously in a unified objective function, we develop a semi-supervised sparse coding algorithm.

Keywords: Semi-Supervised Learning, Sparse Coding, Manifold Learning

1. Introduction

Sparse coding (Sc) has been a popular and effective data representation method for many applications, including pattern recognition, bioinformatics, computer vision, etc. Given a data sample with its feature vector, Sc tries to learn a codebook with some codeworks, and approximate the data sample as the linear combination of the codewords. Sc assume that only a few codewords in the codebook is enough to represent the
data sample, thus the combination coefficients should be sparse, i.e. most of the
coefficients are zeros, leaving only a few of them non-zeros. The linear combi-
nation coefficients of the data sample could be its new representation. Because
they are sparse, we also call the coefficient vector spares code. To solve the
sparse code, we usually minimize the approximation error with regard to the
codebook and the sparse code, and at the same time seek the sparsity of the
sparse code.

Although Sc has been used in many pattern recognition applications, such
as palmprint recognition [13], dynamic texture recognition [14], human action
recognition [15], speech recognition [16], digit recognition [16], and face recogni-
tion [17], in most cases, Sc is used as a unsupervised learning method. When Sc
is performed to the training data set, it is assumed that the class labels of the
training samples are unavailable. Then after the sparse codes are learned, they
will be used to learn a classifier. Thus the class labels are ignored during sparse
coding procedure. However, in most pattern recognition problems, the class
labels of the training samples are usually given. Thus its is necessary to
improve the discriminative ability of the learned sparse codes for the classifica-
tion purpose. To solve this problem, a few supervised Sc methods are proposed to
include the class labels during the coding of the samples. For example, Mairal et
al. [18] proposed to learn the sparse codes of the samples and a classifier in the
sparse code space simultaneously, by constructing and optimizing a unified ob-
jective function for the Sc parameters and the classification parameters. Wang
et al. [19] proposed the discriminative Sc method based on multi-manifolds,
by learning discriminative class-conditioned codebooks and sparse codes from
both data feature spaces and class labels. Though these methods used the class
labels, they requires that all the training samples are labeled. However, in some
real-world applications, there are only very few training samples labeled, while
the remaining most samples are unlabeled. Learning from such training set is
called semi-supervised learning [20, 21, 22, 23]. Semi-supervised learning, com-
pared to the supervised learning, can explores both the labels of the labeled
samples, and the distribution of the over all data set containing labeled and
unlabeled samples. When there are few labeled samples, they are not sufficient to learn an effective classifier using a supervised learning algorithm. In this case, it is necessary to include the unlabeled samples to explore the overall distribution. Many semi-supervised learning algorithms have been proposed to learn classifier from both labeled and unlabeled samples (inductive learning) \[24, 25\], or to learn the labels of the unlabeled samples from the labeled samples (transductive learning) \[26, 27, 28\]. However, surprisingly, no work has been done to learn discriminate sparse codes from partially labeled data set by utilizing both the labels and feature vectors of labeled samples, and the feature vectors of the unlabeled data samples. It is interesting to note that He et al. \[29\] proposed to use the Sc method to construct a sparse graph from the data set for the transductive learning problem, so that the class labels could be prorogated from the labeled samples to the unlabeled samples via the sparse code. However, we should note that during the sparse graph learning procedure using Sc, the class labels of the labeled samples are ignored. Thus in He et al.’s work \[29\], Sc is also performed in an unsupervised way. Similarly, Sc is also used to construct a sparse graph for transductive learning problem in \[30\].

To fill this gap, we propose the semi-supervised Sc method in this paper. Given a data set with only few of the samples labeled, besides conducting Sc for all the samples, we also assume that the class labels for all the samples could be learned from their sparse codes. To do this, we define variable class labels for all the samples, and a classifier to predict the variable class labels. The variable class labels learning are regularized by the manifold of the data set, and the labels of the labeled samples. To learn the codebook, sparse codes, variable class labels, and the classifier parameters simultaneously, we construct a unified objective for them. In the objective function, besides the approximation error term and the sparsity term for Sc, we also introduce the class label approximation error term, and the manifold regularization term for variable class labels. By optimizing this objective function, we try to predict the variable class label from the sparse codes, thus the learned sparse code is naturally discriminative since it has the ability to predict the class label. Moreover, the learning of the
class labels of the unlabeled samples is regularized by the known labels of the labeled samples, the sparse codes and the manifold structure of the data set. The contributions of this paper are two folds:

1. We proposed a discriminative Sc method which could learn from semi-supervised data set. It is a discriminative representation and both labeled and unlabeled data samples could be used to improve its discriminative ability.

2. Moreover, it is also an inductive learning method since it learns a codebook and a classifier from the semi-supervised training set, which could be further used to code and classifier the test samples.

The rest parts of this paper is organized as follows: in section 2 we introduce the proposed semi-supervised Sc method, and finally in section 3 the paper is concluded.

2. Proposed Method

In this section, we introduce the proposed semi-supervised learning method. An objective function is firstly constructed, and then an iterative algorithm is developed to optimized it.

2.1. Objective Function

We assume that we have a training data set of \( n \) training samples, denoted as \( \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \in \mathbb{R}^d \), where \( \mathbf{x}_i \) is the \( d \)-dimensional feature vector for the \( i \)-th sample. The data set is further denoted as a data matrix as \( X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n} \), where the \( i \)-th column is the feature vector of the \( i \)-th sample. We assume that we are dealing with a \( c \)-class semi-supervised classification problem, and only the first \( l \) samples are labeled, while the remaining samples are unlabeled. For a labeled sample \( \mathbf{x}_i \), we define a \( c \)-dimensional binary class label vector \( \hat{\mathbf{y}}_i \in \{1, 0\}^c \), with its \( \nu \)-th element equal to one if it is labeled to the \( \nu \)-th class, and the reminding elements equal to zeros. The class label vector set of the labeled samples are denoted as \( \{\hat{\mathbf{y}}_1, \cdots, \hat{\mathbf{y}}_l\} \in \mathbb{R}^c \), and they are further
organized as a matrix $\hat{Y}_i = [\hat{y}_1, \cdots, \hat{y}_l] \in \{1, 0\}^{c \times l}$, with its $i$-th column as the label vector of the $i$-th sample. To construct the objective function, we consider the following three problems:

- **Sparse coding**: Given a sample $x_i$, sparse coding tries to learn a codebook matrix $B = [b_1, \cdots, b_m] \in \mathbb{R}^{d \times m}$, where its columns are $m$ codewords, and a $m$-dimensional coding vector $s_i \in \mathbb{R}^m$, so that $x_i$ could be approximated as the linear combination of the codewords,

$$x_i \approx Bs_i$$

And at the same time, $s_i$ should be as sparse as possible. Thus we also call $s_i$ sparse code. The sparse code $s_i$ is a new representation of $x_i$. The sparse codes of the training samples are organized in a sparse code matrix $S = [s_1, \cdots, s_n] \in \mathbb{R}^{m \times n}$, with its $i$-th column as the sparse code of the $i$-th sample. To learn the codebook and the sparse codes from the training set, the following optimization problem is proposed,

$$\min_{B, S} \sum_{i=1}^n \left\{ \|x_i - Bs_i\|_2^2 + \alpha\|s_i\|_1 \right\}$$

s.t $\|b_k\|_2^2 \leq c$

where the first term $\|x_i - Bs_i\|_2^2$ is the approximation error term, the second term $\|s_i\|_1$ is introduced to encourage the sparsity of each $x_i$, and $\alpha$ is a trade-off parameter. Moreover, $\|b_k\|_2^2 \leq c$ is imposed to to reduce the complexity of each codeword.

- **Class Label Learning**: We also propose to learn the class label vectors from the sparse code space for all the training samples by a linear function. To do this, we introduce a variable label vector for each sample $x_i$ as $y_i \in \mathbb{R}^c$. Please note that we release its as a real value vector instead of a binary vector, and each element presents its membership of each class. The variable class label vector set for all the training samples are denoted as
\{y_1, \cdots, y_n\} \in \mathbb{R}^c$, and further organized as a variable class label matrix, 
\[ Y = [y_1, \cdots, y_n] \in \mathbb{R}^{c \times n} \]. We assume that its class label vector could be approximated from its sparse code by a linear classifier,

\[ y_i \approx Ws_i \] (3)

where \( W \in \mathbb{R}^{c \times m} \) is the classifier parameter matrix. To learn the class labels and the classifier parameter matrix, we propose the following optimization problem,

\[
\begin{aligned}
\min_{S, W, Y} & \sum_{i=1}^{n} \|y_i - Ws_i\|_2^2 \\
\text{s.t} & \|w_k\|_2^2 \leq e, k = 1, \cdots, m \\
& y_i = \hat{y}_i, i = 1, \cdots, l.
\end{aligned}
\] (4)

As we can see from the above objective function, we use the squared \( L_2 \) norm distance \( \|y_i - Ws_i\|_2^2 \) as the approximation error for the \( i \)-th sample. Moreover, \( \|w_k\|_2^2 \leq e \) constrain is introduced to reduce the complexity of the classifier, and \( y_i = \hat{y}_i, i = 1, \cdots, l \) constrains are introduced so that the learned labels could respect the known labels of the labeled samples.

**Manifold Label Regularization**: We also hope the learned class labels could respect the manifold structure of the data set. We assume that for each sample \( x_i \), its class label vector \( y_i \) could be reconstructed by the class labels of its nearest neighbors \( N_i \),

\[ y_i \approx \sum_{j \in N_i} A_{ij} y_j \] (5)

where \( A_{ij} \) is the reconstruction coefficient, which could be solved in the same way as Locally Linear Embedding (LLE) \( [31] \) by minimizing the reconstruction error in the original feature space,
\[
\min_{A_{ij}} \sum_{i=1}^{n} \left\| x_i - \sum_{j \in \mathcal{N}_i} A_{ij} x_j \right\|_2^2
\]
\[
s.t \ A_{ij} \geq 0, j \in \mathcal{N}_i, \sum_{j \in \mathcal{N}_i} A_{ij} = 1
\]
\[
A_{ij} = 0, j \notin \mathcal{N}_i
\]

With the solved reconstruction coefficient matrix \( A = [A_{ij}] \in \mathbb{R}^{n \times n} \), we regularize the class label learning with the following optimization problem,

\[
\min_{\mathcal{Y}} \sum_{i=1}^{n} \left\| \mathcal{Y}_i - \sum_{j \in \mathcal{N}_i} A_{ij} \mathcal{Y}_j \right\|_2^2
\]
\[
s.t \ \mathcal{Y}_i = \hat{\mathcal{Y}}_i, i = 1, \cdots, l.
\]

By doing this, we assume that label space and the data space share the same local linear reconstruction coefficients.

The overall optimization problem is formulated by combining the three problems in (2), (4) and (7), and the following optimization problem is obtained,

\[
\min_{B,S,Y,W} \sum_{i=1}^{n} \left\{ \| x_i - B s_i \|_2^2 + \alpha \| s_i \|_1 + \beta \| \mathcal{Y}_i - W s_i \|_2^2 + \gamma \left\| \mathcal{Y}_i - \sum_{j \in \mathcal{N}_i} A_{ij} \mathcal{Y}_j \right\|_2^2 \right\}
\]
\[
s.t. \ \| b_k \|_2^2 \leq c, \| w_k \|_2^2 \leq c, k = 1, \cdots, m,
\]
\[
\mathcal{Y}_i = \hat{\mathcal{Y}}_i, i = 1, \cdots, l.
\]

where \( \beta \) and \( \gamma \) are the tradeoff parameters. Please note that in this formulation, we do not use the class labels to regularize the sparse codes directly. Instead, a classifier is learned to learn the class label from the sparse codes, so that the class labels, the classifiers, and the sparse codes could be learned together and regularize each other.
2.2. Optimization

It is hard to find a close form solution for the problem in (8). Thus we use the alternative optimization strategy to optimize it in an iterative algorithm. In each iteration, the variables are optimized by turn. When one of the variables is optimized, the others are fixed.

2.2.1. Optimizing $B$ and $W$

We first discuss the optimization of $B$ and $W$. As we show later, they could be solve together as different parts of an generalized codebook. By removing the terms irrelevant to $B$ and $W$, and fixing $S$ and $Y$, we obtain the following optimization problem,

$$\min_{B,W} \sum_{i=1}^{n} \left\{ \|x_i - Bs_i\|^2_2 + \beta \|y_i - Ws_i\|^2_2 \right\}$$

$$= \|X - BS\|^2_2 + \left\| \sqrt{\beta}Y - \sqrt{\beta}WS \right\|^2_2$$

s.t. $\|b_k\|^2_2 \leq c, \|w_k\|^2_2 \leq e, k = 1, \cdots, m. \tag{9}$

We define an extended data matrix by catenating $X$ and $Y$ as $\tilde{X} = \begin{bmatrix} X \\ \sqrt{\beta}Y \end{bmatrix}$, and an extended codebook matrix by catenating $B$ and $W$ as $\tilde{B} = \begin{bmatrix} B \\ \sqrt{\beta}W \end{bmatrix}$. Moreover, we combine the two constrains $\|b_k\|^2_2 \leq c$ and $\|w_k\|^2_2 \leq e$ to one single constrain $\|b_k\|^2_2 + \beta \|w_k\|^2_2 \leq c + \beta e$. This constrain could be rewritten as $\|\begin{bmatrix} b_k \\ \sqrt{\beta}w_k \end{bmatrix}\|^2_2 \leq (c + \beta e)$, where $\tilde{b}_k$ is the $k$-th column of the $\tilde{B}$ matrix. In this way, the optimization is rewritten as

$$\min_{\tilde{B}} \left\| \tilde{X} - \tilde{BS} \right\|^2_2$$

s.t. $\|\tilde{b}_k\|^2_2 \leq (c + \beta e), k = 1, \cdots, m. \tag{10}$

This problem could be solved using the Lagrange dual method proposed in \[32\]. After $\tilde{B}$ is solved, $B$ and $W$ could be recovered from it as
\[ B = \tilde{B}_{1, \ldots, d}, \]
\[ W = \frac{1}{\sqrt{\beta}} \tilde{B}_{d+1, \ldots, d+c}, \]

where \( \tilde{B}_{1, \ldots, d} \) is the first \( d \) rows of matrix \( \tilde{B} \), and \( \tilde{B}_{d+1, \ldots, d+c} \) is the \( d+1 \) to \( d+c \) rows of matrix \( \tilde{B} \).

### 2.2.2. Optimizing \( S \)

To solve the sparse codes in \( S \), we fix \( \tilde{B} \), remove the terms irrelevant to \( S \), and the following problem is obtained,

\[
\min_{\tilde{B}} \| \bar{X} - \tilde{B}S \|_2^2 + \alpha \sum_{i=1}^{n} \| s_i \|_1 \quad (12)
\]

Similarly, this problem could be solved efficiently by the feature-sign search algorithm proposed in [32].

### 2.2.3. Optimizing \( Y \)

To solve the class label vectors in \( Y \), we fix \( B \), \( S \) and \( W \), remove the terms irrelevant to \( Y \), and get the following optimization problem,

\[
\min_{Y} \beta \sum_{i=1}^{n} \| y_i - Ws_i \|_2^2 + \gamma \sum_{i=1}^{n} \| y_i - \sum_{j \in N_i} A_{ij}y_j \|_2^2
\]
\[
= \beta \| Y - WS \|_2^2 + \gamma \| Y(I - A)^\top \|_2^2
\]
\[
s.t \ y_i = \hat{y}_i, i = 1, \ldots, l.
\]

We separate the class label matrix to to sub-matrices as \( Y = [Y_l \ Y_u] \), where \( Y_l \) contains the first \( l \) columns of \( Y \), which are the variable class label vectors of the labeled samples, while \( Y_u \) contains the remaining columns which are the variable class label vectors of the unlabeled samples. Similarly, we also separate \( S \) to two sub-matrices as \( S = [S_l \ S_u] \), where \( S_l \) contains the sparse codes of the labeled samples, while \( S_u \) contains the sparse codes of the labeled samples. Moreover, we define matrix \( Q = (I - A)^\top \) for convenience, and also separate it to two sub-matrices as \( Q = \begin{bmatrix} Q_l \\ Q_u \end{bmatrix} \) where \( Q_l \) contains its first \( l \) rows and \( Q_u \) contains its
remaining rows. With these definitions, we could rewrite the objective function in (13) as

\[
\beta \|Y - WS\|_2^2 + \gamma \|Y(I - A)^\top\|_2^2
\]

\[
= \beta \|Y_l - WS_l\|_2^2 + \beta \|Y_u - WS_u\|_2^2 + \gamma \|Y_l - W S_l\|_2^2 + \gamma \|W S_u\|_2^2 + \gamma \|Y_l Q_l + Y_u Q_u\|_2^2
\]  

(14)

Since it is constrained that \(y_i = \hat{y}_i\) for any \(i = 1, \cdots, l\), \(Y_l = \hat{Y}_l\) and it is actually not a variable. Thus we substitute \(Y_l = \hat{Y}_l\) to (14), only treat \(Y_u\) as variable to solve, and obtain the following optimization problem with regard to \(Y_u\),

\[
\min_{Y_u} \left\{ f(Y_u) = \beta \|Y_l - WS_l\|_2^2 + \beta \|Y_u - WS_u\|_2^2 + \gamma \|Y_l Q_l + Y_u Q_u\|_2^2 \right\}
\]  

(15)

To solve this problem, we simply set the derivative of the objective function \(f(Y_u)\) with regard to \(Y_u\) to zero, and obtain the solution for \(Y_u\),

\[
\frac{\partial f(Y_u)}{\partial Y_u} = 2\beta (Y_u - WS_u) + 2\gamma \left( Y_l Q_l + Y_u Q_u \right) Q_u^\top = 0
\]

\[
\Rightarrow Y_u = \left( \beta W S_u - \gamma Y_l Q_l Q_l^\top \right) \left( \beta I + \gamma Q_u Q_u^\top \right)^{-1}
\]  

(16)

2.3. Algorithm

We summarize the iterative learning algorithm for semi-supervised sparse coding (Semi-Sup Sc) in Algorithm 1. As we can see from the algorithm, we employ the original sparse coding algorithm to initialize the sparse code matrix, and employ the Linear Neighborhood Propagation (LNP) algorithm \[33\] to initialize the class label matrix. The iterations are repeated for \(T\) times and the updated solutions for \(B, S, W\) and \(Y_u\) are outputted.

2.4. Coding and Classifying New Sample

When a new test sample \(x\) comes, we first find its nearest neighbors from the training set \(\mathcal{N}\), and we assume that it could be reconstructed by these nearest
Algorithm 1 Semi-Sup Sc - Learning Algorithm.

Input: Training data matrix $X$;
Input: Training data label matrix for labeled samples $\hat{Y}_l$;
Input: Tradeoff parameters $\alpha$, $\beta$ and $\gamma$.

Initialize the sparse code matrix $S^0$ by performing original sparse coding to $X$;
Initialize the class label matrix $Y^0$ by performing (LNP) algorithm;

for $t = 1, \cdots, T$ do
    Update codebook matrix $B^t$ and the classifier parameter matrix $W^t$ as in (10) by fixing $S^{t-1}$, and $Y^{t-1}$;
    Update sparse code matrix $S^t$ as in (12) by fixing $B^t$, and $Y^{t-1}$;
    Update the variable class label matrix $Y^t$ as in (16) by fixing $B^t$, and $S^t$;
end for

Output: The codebook matrix $B^T$, the sparse code matrix $S^T$, the classifier parameter matrix $W^T$, and the class label matrix for the unlabeled samples $Y_u^T$.

neighbors. The reconstruction coefficients $a_{i| i \in \mathcal{N}}$ are computed by solving a problem in (1). To solve its sparse code vector $s$, and its class label vector $y$, we use the codebook $B$, classifier parameter matrix $W$, and the class label matrix $Y$ learned from the training set. The optimization problem is formulated as

$$
\min_{s, y} \left\{ \|x - Bs\|_2^2 + \alpha \|s\|_1 + \beta \|y - Ws\|_2^2 + \gamma \left\| y - \sum_{i \in \mathcal{N}} a_i y_i \right\|_2^2 \right\}
$$

(17)

To solve this problem, we also adopt the alternative optimization strategy. In an iterative algorithm, we optimize $s$ and $y$ in turn.

- **Solving s**: When $s$ is optimized, $y$ is fixed, and the following problem is solved,

---

11
\[
\min_s \left\{ \|x - Bs\|^2_2 + \alpha\|s_i\|_1 + \beta\|y - Ws\|^2_2 = \|\tilde{x} - \tilde{B}s\|^2_2 + \alpha\|s_i\|_1 \right\}
\]

(18)

where \(\tilde{x} = \begin{bmatrix} x \\ \sqrt{\beta}y \end{bmatrix} \). This problem could be solved using the feature-sign search algorithm proposed in [32].

- **Solving \(y\):** When \(s\) is fixed and \(y\) is optimized, we have the following problem,

\[
\min_y \left\{ \beta\|y - Ws\|^2_2 + \gamma \left\| y - \sum_{i \in \mathcal{N}} a_i y_i \right\|^2_2 \right\}
\]

(19)

It could be solved easily by setting the derivative with regard to \(y\) to zero, and the solution is obtained as

\[
y = \frac{1}{\beta + \gamma} \left( \beta Ws + \gamma \sum_{i \in \mathcal{N}} a_i y_i \right)
\]

(20)

By repeating the above two procedures for \(T\) times, we could obtain the optimal sparse code \(s\) and the class label vector \(y\) for the test sample \(x\). It will be further classifier to the \(\iota^*\)-th class with the largest value in the class label vector \(y\),

\[
\iota^* = \arg \max_{\iota \in \{1, \ldots, c\}} y(\iota)
\]

(21)

where \(y(\iota)\) is the \(\iota\)-th element of \(y\).

3. Conclusion

In this paper, we proposed a semi-supervised sparse coding method. We assume that the class labels of the unlabeled samples could be predicted from the sparse codes, and by minimizing the prediction error, we use the class label information to regularize the learning of the sparse codes. Moreover, we also regularize the learning of class labels with the manifold structure and the learn
sparse code and classifier. Thus the class labels, sparse codes and the classifier are updated alternatively in an iterative algorithm. This work is the first effort toward learning discriminative sparse codes from partially labeled data set.

References

[1] B. Olshausen, D. Field, Sparse coding of sensory inputs, Current Opinion in Neurobiology 14 (4) (2004) 481–487.

[2] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, 2007, pp. 801–808.

[3] M. Zheng, J. Bu, C. Chen, Hessian sparse coding, Neurocomputing 123 (2014) 247–254.

[4] S. Liu, M. Liu, Fingerprint orientation modeling by sparse coding, 2012, pp. 176–181.

[5] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, H. Hermansky, Sparse coding for speech recognition, 2010, pp. 4346–4349.

[6] R. Jenatton, J. Mairal, G. Obozinski, F. Bach, Proximal methods for hierarchical sparse coding, Journal of Machine Learning Research 12 (2011) 2297–2334.

[7] S. Chen, C.-Y. Zhang, K. Song, Recognizing short coding sequences of prokaryotic genome using a novel iteratively adaptive sparse partial least squares algorithm, Biology Direct 8 (1).

[8] K. Zhang, J. Han, T. Groesser, G. Fontenay, B. Parvin, Inference of causal networks from time-varying transcriptome data via sparse coding, PLoS ONE 7 (8).

[9] Z. Lei, K. Chen, H. Li, H. Liu, A. Guo, The gaba system regulates the sparse coding of odors in the mushroom bodies of drosophila, Biochemical and Biophysical Research Communications 436 (1) (2013) 35–40.
[10] C. Wang, S. Yan, L. Zhang, H.-J. Zhang, Multi-label sparse coding for automatic image annotation, 2009, pp. 1643–1650.

[11] T. Hazan, S. Polak, A. Shashua, Sparse image coding using a 3d non-negative tensor factorization, Vol. I, 2005, pp. 50–57.

[12] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, 2009, pp. 1794–1801.

[13] L. Shang, W. Huai, G. Dai, J. Chen, J. Du, Palmprint recognition using 2d-gabor wavelet based sparse coding and rbpnn classifier, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6064 LNCS (PART 2) (2010) 112–119.

[14] B. Ghanem, N. Ahuja, Sparse coding of linear dynamical systems with an application to dynamic texture recognition, 2010, pp. 987–990.

[15] Y. Liu, Y. Li, Human action recognition in videos using distance image volumes and sparse coding, Journal of Computational Information Systems 8 (9) (2012) 3557–3564.

[16] K. Labusch, E. Barth, T. Martinetz, Simple method for high-performance digit recognition based on sparse coding, IEEE Transactions on Neural Networks 19 (11) (2008) 1985–1989.

[17] M. Yang, L. Zhang, J. Yang, D. Zhang, Robust sparse coding for face recognition, 2011, pp. 625–632.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, 2009, pp. 1033–1040.

[19] J. J.-Y. Wang, H. Bensmail, N. Yao, X. Gao, Discriminative sparse coding on multi-manifolds, Knowledge-Based Systems 54 (0) (2013) 199 – 206.

[20] M. Bilenko, S. Basu, R. Mooney, Integrating constraints and metric learning in semi-supervised clustering, 2004, pp. 81–88.
[21] L. Käll, J. Canterbury, J. Weston, W. Noble, M. MacCoss, Semi-supervised learning for peptide identification from shotgun proteomics datasets, Nature Methods 4 (11) (2007) 923–925.

[22] M. Belkin, P. Niyogi, Semi-supervised learning on riemannian manifolds, Machine Learning 56 (1-3) (2004) 209–239.

[23] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, Vol. 2, 2003, pp. 912–919.

[24] J. Y. Ching, A. K. Wong, K. C. Chan, Class-dependent discretization for inductive learning from continuous and mixed-mode data, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (7) (1995) 641–651.

[25] R. Michalski, A theory and methodology of inductive learning, Artificial Intelligence 20 (2) (1983) 111–161.

[26] K. Yu, J. Bi, V. Tresp, Active learning via transductive experimental design, Vol. 2006, 2006, pp. 1081–1088.

[27] Y. Chen, G. Wang, S. Dong, Learning with progressive transductive support vector machine, Pattern Recognition Letters 24 (12) (2003) 1845–1855.

[28] T. Joachims, Transductive learning via spectral graph partitioning, Vol. 1, 2003, pp. 290–297.

[29] R. He, W.-S. Zheng, B.-G. Hu, X.-W. Kong, Nonnegative sparse coding for discriminative semi-supervised learning, 2011, pp. 2849–2856.

[30] S. Yang, X. Wang, L. Yang, Y. Han, L. Jiao, Semi-supervised action recognition in video via labeled kernel sparse coding and sparse l1 graph, Pattern Recognition Letters 33 (14) (2012/10/15) 1951 – 6.

[31] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[32] H. Lee, A. Battle, R. Raina, A. Ng, Efficient sparse coding algorithms, in: Advances in neural information processing systems, 2006, pp. 801–808.

[33] J. Wang, F. Wang, C. Zhang, H. C. Shen, L. Quan, Linear neighborhood propagation and its applications, Pattern Analysis and Machine Intelligence, IEEE Transactions on 31 (9) (2009) 1600–1615.