Semi-Supervised Sparse Coding

Jim Jing-Yan Wang and Xin Gao*

Abstract— **Sparse coding approximates the data sample as a sparse linear combination of some basic codewords and uses the sparse codes as new presentations. In this paper, we investigate learning discriminative sparse codes by sparse coding in a semi-supervised manner, 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 constraints provided by the labels of the 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 classifier parameters simultaneously in a unified objective function, we develop a semi-supervised sparse coding algorithm. Experiments on two real-world pattern recognition problems demonstrate the advantage of the proposed methods over supervised sparse coding methods on partially labeled data sets.**

I. Introduction

SPARSE Coding (SC) [1], [2], [3], [4], [5] has been a popular and effective data representation method for many applications, including pattern recognition [6], [7], [8], bioinformatics [9], [10], [11] and computer vision [12], [13], [14]. 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 are 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 combination coefficients of the data sample could be its new representation. Because they are sparse, the coefficient vector is often referred to as the sparse code. To solve the sparse code, one usually minimizes the approximation error with regard to the codebook and the sparse code, and at the same time seeks the sparsity of the sparse code.

Although SC has been used in many pattern recognition applications, such as palmprint recognition [15], dynamic texture recognition [16], human action recognition [17], speech recognition [7], digit recognition [18], and face recognition [19], in most cases, SC is used as an 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 the sparse coding procedure. However, in most pattern recognition problems, the class labels of the training samples are given. It is thus natural to improve the discriminative ability of the learned sparse codes for the classification purpose. To solve this problem, a few supervised SC methods were proposed to include the class labels during the coding of the samples. For example, Mairal et al. [20] proposed to learn the sparse codes of the samples and a classifier in the sparse code space simultaneously, by constructing and optimizing a unified objective function for the SC parameters and the classification parameters. Wang et al. [21] 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 use the class labels, they require that all the training samples are labeled. However, in some real-world applications, there are only very few training samples labeled, while the remaining training samples are unlabeled. Learning from such a training set is called semi-supervised learning [22], [23], [24], [25]. Semi-supervised learning, compared to the supervised learning, can explore both the labels of the labeled samples and the distribution of the overall 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 algorithm has been proposed to learn classifier from both labeled and unlabeled samples (inductive learning) [26], [27], or to learn the labels of the unlabeled samples from the labeled samples (transductive learning) [28], [29], [30]. However, surprisingly, no work has been done to learn discriminate sparse codes from partially labeled data set by utilizing both the labels and the feature vectors of the labeled samples, and the feature vectors of the unlabeled data samples. It is interesting to note that He et al. [31] 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, during the sparse graph learning procedure using SC, the class labels of the labeled samples were ignored. Thus in He et al.'s work [31], SC was also performed in an unsupervised way. Similarly, SC was also used to construct a sparse graph for the transductive learning problem in [32].

To fill this gap, we propose a semi-supervised SC method

Jim Jing-Yan Wang is with the University at Buffalo, The State University of New York, Buffalo, NY 14203, USA, and the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, 215006, China. (E-mail: jimjywang@gmail.com.)

Xin Gao is with the Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia. (Email: xin.gao@kaust.edu.sa.)

Correspondence should be addressed to Xin Gao.

This study was supported by a grant from King Abdullah University of Science and Technology (KAUST), Saudi Arabia, and Provincial Key Laboratory for Computer Information Processing Technology, China (Grant No. KJS1324).

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 label learning is 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 propose a unified objective function. 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 labels. 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 in two folds:

1) We propose 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 power.
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 classify the test samples.

The rest parts of this paper is organized as follows: in Section II, we introduce the proposed semi-supervised SC method; in Section III, the experiment results on two data sets are reported; and finally in Section IV the paper is concluded.

II. 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 optimize it.

A. 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 $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in R^{d \times n}$, 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 $\widehat{\mathbf{y}}_i \in \{1, 0\}^c$, with its ι-th element equal to one if it is labeled as the ι-th class, and the reminding elements equal to zero. The class label vector set of the labeled samples are denoted as $\{\widehat{\mathbf{y}}_1, \cdots, \widehat{\mathbf{y}}_l\} \in \mathbb{R}^c$, and they are further organized as a matrix $\widehat{Y}_l = [\widehat{\mathbf{y}}_1, \cdots, \widehat{\mathbf{y}}_l] \in \{1, 0\}^{c \times l}$, 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 $\mathbf{x}_i$, sparse coding tries to learn a codebook matrix $B = [\mathbf{b}_1, \cdots, \mathbf{b}_m] \in \mathbb{R}^{d \times m}$, where its columns are m codewords, and an m-dimensional coding vector $\mathbf{s}_i \in \mathbb{R}^m$, so that $\mathbf{x}_i$ could be approximated as the linear combination of the codewords,

$$\mathbf{x}_i \approx B\mathbf{s}_i \tag{1}$$

And at the same time, $\mathbf{s}_i$ should be as sparse as possible. Thus we also call $\mathbf{s}_i$ sparse code. The sparse code $\mathbf{s}_i$ is a new representation of $\mathbf{x}_i$. The sparse codes of the training samples are organized in a sparse code matrix $S = [\mathbf{s}_1, \cdots, \mathbf{s}_n] \in R^{m \times n}$, 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,

$$\begin{aligned} \min_{B,S} \ & \sum_{i=1}^{n} \left\{ \|\mathbf{x}_i - B\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 \right\}, \\ s.t \ & \|\mathbf{b}_k\|_2^2 \leq c, \end{aligned} \tag{2}$$

where the first term $\|\mathbf{x}_i - B\mathbf{s}_i\|_2^2$ is the approximation error term, the second term $\|\mathbf{s}_i\|_1$ is introduced to encourage the sparsity of each $\mathbf{x}_i$, and α is a trade-off parameter. Moreover, $\|\mathbf{b}_k\|_2^2 \leq c$ 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 $\mathbf{x}_i$ as $\mathbf{y}_i \in \mathbb{R}^c$. Please note that we relax it 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 $\{\mathbf{y}_1, \cdots, \mathbf{y}_n\} \in \mathbb{R}^c$, and further organized as a variable class label matrix, $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_n] \in \mathbb{R}^{c \times n}$. We assume that its class label vector could be approximated from its sparse code by a linear classifier,

$$\mathbf{y}_i \approx W\mathbf{s}_i, \tag{3}$$

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,

$$\begin{aligned} \min_{S,W,Y} \ & \sum_{i=1}^{n} \|\mathbf{y}_i - W\mathbf{s}_i\|_2^2 \\ s.t \ & \|\mathbf{w}_k\|_2^2 \leq e, k = 1, \cdots, m \\ & \mathbf{y}_i = \widehat{\mathbf{y}}_i, i = 1, \cdots, l. \end{aligned} \tag{4}$$

As we can see from the above objective function, we use the squared L_2 norm distance $\|\mathbf{y}_i - W\mathbf{s}_i\|_2^2$ as the approximation error for the i-th sample. Moreover,

$\|\mathbf{w}_k\|_2^2 \leq e$ constrain is introduced to reduce the complexity of the classifier, and $\mathbf{y}_i = \widehat{\mathbf{y}}_i, i = 1, \cdots, l$ 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 $\mathbf{x}_i$, its class label vector $\mathbf{y}_i$ could be reconstructed by the class labels of its nearest neighbors $\mathcal{N}_i$,

$$\mathbf{y}_i \approx \sum_{j \in \mathcal{N}_i} A_{ij} \mathbf{y}_j, \tag{5}$$

where A_{ij} is the reconstruction coefficient, which could be solved in the same way as Locally Linear Embedding (LLE) [33] by minimizing the reconstruction error in the original feature space,

$$\begin{aligned} \min_{A_{ij}|_{j=1}^n} & \left\| \mathbf{x}_i - \sum_{j \in \mathcal{N}_i} A_{ij} \mathbf{x}_j \right\|_2^2 \\ s.t\ & A_{ij} \geq 0, j \in \mathcal{N}_i, \ \sum_{j \in \mathcal{N}_i} A_{ij} = 1 \\ & A_{ij} = 0, j \notin \mathcal{N}_i \end{aligned} \tag{6}$$

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,

$$\begin{aligned} \min_{Y} & \sum_{i=1}^n \left\| \mathbf{y}_i - \sum_{j \in \mathcal{N}_i} A_{ij} \mathbf{y}_j \right\|_2^2 \\ s.t\ & \mathbf{y}_i = \widehat{\mathbf{y}}_i, i = 1, \cdots, l. \end{aligned} \tag{7}$$

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,

$$\begin{aligned} \min_{B,S,Y,W} & \sum_{i=1}^n \left\{ \|\mathbf{x}_i - B\mathbf{s}_i\|_2^2 + \alpha \|\mathbf{s}_i\|_1 + \beta \|\mathbf{y}_i - W\mathbf{s}_i\|_2^2 \right. \\ & \left. + \gamma \left\| \mathbf{y}_i - \sum_{j \in \mathcal{N}_i} A_{ij} \mathbf{y}_j \right\|_2^2 \right\} \\ s.t.\ & \|\mathbf{b}_k\|_2^2 \leq c, \|\mathbf{w}_k\|_2^2 \leq e, k = 1, \cdots, m, \\ & \mathbf{y}_i = \widehat{\mathbf{y}}_i, i = 1, \cdots, l. \end{aligned} \tag{8}$$

where β and γ are the tradeoff parameters, which are selected by cross-validation. 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 assign 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.

B. Optimization

It is difficult to find a closed-form solution for the problem in (8). Thus we use the alternate 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.

1) Optimizing B and W: We first discuss the optimization of B and W. As we show later, they could be solved 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,

$$\begin{aligned} \min_{B,W} & \sum_{i=1}^n \left\{ \|\mathbf{x}_i - B\mathbf{s}_i\|_2^2 + \beta \|\mathbf{y}_i - W\mathbf{s}_i\|_2^2 \right\} \\ & = \|X - BS\|_2^2 + \left\| \sqrt{\beta} Y - \sqrt{\beta} WS \right\|_2^2 \\ s.t.\ & \|\mathbf{b}_k\|_2^2 \leq c, \|\mathbf{w}_k\|_2^2 \leq e, k = 1, \cdots, m. \end{aligned} \tag{9}$$

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

$$\begin{aligned} \min_{\widetilde{B}} & \left\| \widetilde{X} - \widetilde{B} S \right\|_2^2 \\ s.t & \left\| \widetilde{\mathbf{b}}_k \right\|_2^2 \leq (c + \beta e), k = 1, \cdots, m. \end{aligned} \tag{10}$$

This problem could be solved using the Lagrange dual method proposed in [34]. After $\widetilde{B}$ is solved, B and W could be recovered from it as

$$\begin{aligned} B &= \widetilde{B}_{1, \cdots, d}, \\ W &= \frac{1}{\sqrt{\beta}} \widetilde{B}_{d+1, \cdots, d+c}, \end{aligned} \tag{11}$$

where $\widetilde{B}_{1, \cdots, d}$ is the frist d rows of the matrix $\widetilde{B}$, and $\widetilde{B}_{d+1, \cdots, d+c}$ is the $d+1$ to $d+c$ rows of matrix $\widetilde{B}$.

2) Optimizing S: To solve the sparse codes in S, we fix $\widetilde{B}$, remove the terms irrelevant to S, and the following problem is obtained,

$$\min_{\widetilde{B}} \left\| \widetilde{X} - \widetilde{B} S \right\|_2^2 + \alpha \sum_{i=1}^n \|\mathbf{s}_i\|_1 \tag{12}$$

Similarly, this problem could be solved efficiently by the feature-sign search algorithm proposed in [34].

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,

$$\begin{aligned}&\min_{Y}\ \beta\sum_{i=1}^{n}\|\mathbf{y}_i - W\mathbf{s}_i\|_2^2 + \gamma\sum_{i=1}^{n}\left\|\mathbf{y}_i - \sum_{j\in\mathcal{N}_i} A_{ij}\mathbf{y}_j\right\|_2^2\\ &\quad=\beta\|Y-WS\|_2^2+\gamma\left\|Y(I-A)^\top\right\|_2^2\\ &s.t\ \mathbf{y}_i=\widehat{\mathbf{y}}_i, i=1,\cdots,l.\end{aligned}\tag{13}$$

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

$$\begin{aligned}&\beta\|Y-WS\|_2^2+\gamma\left\|Y(I-A)^\top\right\|_2^2\\ &=\beta\|Y_l-WS_l\|_2^2+\beta\|Y_u-WS_u\|_2^2+\gamma\left\|[Y_l\ Y_u]\begin{bmatrix} Q_l \\ Q_u \end{bmatrix}\right\|_2^2\\ &=\beta\|Y_l-WS_l\|_2^2+\beta\|Y_u-WS_u\|_2^2+\gamma\|Y_lQ_l+Y_uQ_u\|_2^2\end{aligned}\tag{14}$$

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

$$\begin{aligned}\min_{Y_u}\ &\left\{f(Y_u)=\beta\left\|\widehat{Y}_l-WS_l\right\|_2^2+\beta\|Y_u-WS_u\|_2^2\right.\\ &\left.+\gamma\left\|\widehat{Y}_lQ_l+Y_uQ_u\right\|_2^2\right\}\end{aligned}\tag{15}$$

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,

$$\begin{aligned}&\frac{\partial f(Y_u)}{\partial Y_u}=2\beta\left(Y_u-WS_u\right)+2\gamma\left(\widehat{Y}_lQ_l+Y_uQ_u\right)Q_u^\top=0\\ &\Rightarrow Y_u=\left(\beta WS_u-\gamma\widehat{Y}_lQ_lQ_u^\top\right)\left(\beta I+\gamma Q_uQ_u^\top\right)^{-1}\end{aligned}\tag{16}$$

C. Algorithm

We summarize the iterative learning algorithm for Semi-Supervised Sparse Coding (SSSC) 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 [35] 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.

Algorithm 1 Learning Algorithm of SSSC.

```
Input: Training data matrix $X$;
Input: Training data label matrix for labeled samples $\widehat{Y}_l$;
Input: Tradeoff parameters $\alpha$, $\beta$ and $\gamma$.;
Input: Iteration number $T$.
Initialize the sparse code matrix $S^0$ by performing original sparse coding to $X$;
Initialize the class label matrix $Y^0$;
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 $Y_u^T$.
```

D. Coding and Classifying New Samples

When a new test sample $\mathbf{x}$ comes, we first find its nearest neighbors $\mathcal{N}$ from the training set, and we assume that it could be reconstructed by these nearest neighbors. The reconstruction coefficients $a_i|_{i\in\mathcal{N}}$ are computed by solving a problem in (6). To solve its sparse code vector $\mathbf{s}$, and its class label vector $\mathbf{y}$, 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

$$\begin{aligned}\min_{\mathbf{s},\mathbf{y}}\ &\left\{\|\mathbf{x}-B\mathbf{s}\|_2^2+\alpha\|\mathbf{s}_i\|_1+\beta\|\mathbf{y}-W\mathbf{s}\|_2^2\right.\\ &\left.+\gamma\left\|\mathbf{y}-\sum_{i\in\mathcal{N}}a_i\mathbf{y}_i\right\|_2^2\right\},\end{aligned}\tag{17}$$

where $\mathbf{y}_i$ is the class label vector of the i-th training sample. To solve this problem, we also adopt the alternate optimization strategy. In an iterative algorithm, we optimize $\mathbf{s}$ and $\mathbf{y}$ in turn.

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

$$\begin{aligned}\min_{\mathbf{s}}\ &\left\{\|\mathbf{x}-B\mathbf{s}\|_2^2+\alpha\|\mathbf{s}_i\|_1+\beta\|\mathbf{y}-W\mathbf{s}\|_2^2\right.\\ &\left.=\|\widetilde{\mathbf{x}}-\widetilde{B}\mathbf{s}\|_2^2+\alpha\|\mathbf{s}_i\|_1\right\},\end{aligned}\tag{18}$$

where $\widetilde{\mathbf{x}} = \begin{bmatrix}\mathbf{x}\\ \sqrt{\beta}\mathbf{y}\end{bmatrix}$. This problem could be solved using the feature-sign search algorithm proposed in [34].

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

$$\min_{\mathbf{y}} \left\{ \beta \|\mathbf{y} - W\mathbf{s}\|_2^2 + \gamma \left\| \mathbf{y} - \sum_{i \in \mathcal{N}} a_i \mathbf{y}_i \right\|_2^2 \right\}. \quad (19)$$

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

$$\mathbf{y} = \frac{1}{\beta + \gamma} \left(\beta W \mathbf{s} + \gamma \sum_{i \in \mathcal{N}} a_i \mathbf{y}_i \right) \quad (20)$$

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 ι^*-th class with the largest value in the class label vector **y**,

$$\iota^* = \arg\max_{\iota \in \{1, \cdots, c\}} \mathbf{y}(\iota), \quad (21)$$

where $\mathbf{y}(\iota)$ is the ι-th element of **y**.

III. Experiments

In this section, we evaluate the performance of the proposed semi-supervised sparse coding algorithm on two real-world data sets.

A. Cytochromes P450 Inhibition Prediction

The cytochromes P450 is a family of enzymes which are involved in the metabolism of most modern drugs [36], [37], [38]. There are five major isoforms of cytochromes P450, which are 1A2, 2C9, 2C19, 2D6, and 3A4 [39]. It is very important to model the interactions of the cytochromes P450 with the drug-like compounds in drug-drug interaction studies. In this case, predicting if a given compound can inhibit these isoforms plays an important role in the drug design [40]. Here, we evaluated the proposed algorithm in the problem of cytochromes P450 inhibition prediction.

1) Data Set and Protocol: We collected a data set of compounds for each isoform, and each compound is an inhibitor or a non-inhibitor of the isoform. The numbers of inhibitors and non-inhibitors of each isoform are given in Figure 1. As we can see from the figure, the data sets are not balanced. For each isoform, non-inhibitors are usually more than inhibitors. To represent each compound, we extracted the molecular signatures as features, which were computed from the atomic signatures of circular atomic fragments [41], [42], [43]. The problem of cytochromes P450 inhibition prediction is to learn a predictor from the given data set to predict whether a candidate compound is an inhibitor or a non-inhibitor. Thus it is a binary classification problem.

To conduct the experiment, for each isoform, we performed the 10-fold cross-validation [44] to the data set. Each data set of an isoform was split into ten folds, and each fold was used as the test set in turn, while the remaining nine folds were used as the training set. For each taining set, we only randomly labeled a small part (about 20%)of the compounds with the class labels (inhibitors or non-inhibitors), while leaving the remaining part as unlabeled compounds. The proposed learning algorithm was performed to the molecular signatures of the training compounds to learn the codebook, the classifier and the labels of the unlabeled compounds. Then the compounds in the test set were used as test sample one by one. The learned codebook and the classifier were used to code and classify the test compound.

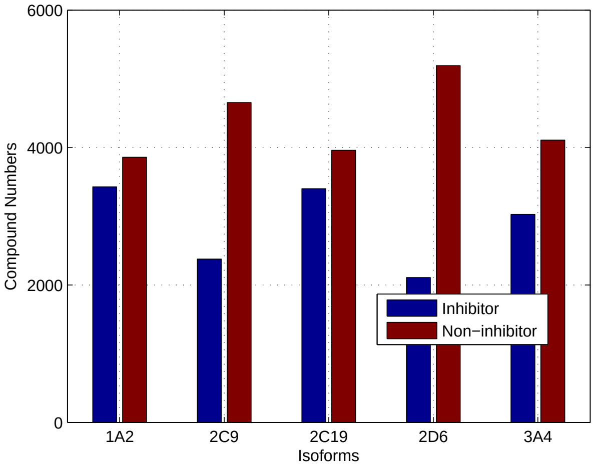

Fig. 1. The numbers of inhibitors and non-inhibitors of each isoform in the cytochromes P450 inhibition prediction data set.

To evaluate the prediction performance, we used the following performance measures as prediction performance metrics: Sensitivity (Sen), Specificity (Spc), Accuracy (Acc), and F1 score (F1). To calculate these metrics, we first calculate the following values for each test set: True Positive (TP) which is the number of inhibitor compounds that were correctly predicted, True Negative (TN) which is the number of non-inhibitor compounds that were correctly predicted, False Positive (FP) which is the number of non-inhibitor compounds wrongly predicted as inhibitor compounds, and False Negative (FN) which is the number of inhibitor compounds wrongly predicted as non-inhibitor compounds. With these values computed from the test set, the performance measures are defined as,

$$\begin{aligned} Sen &= \frac{TP}{TP + FN}, Spc = \frac{TN}{FP + TN}, \\ Acc &= \frac{TP + TN}{TP + TN + FP + FN}, \\ F1 &= \frac{2 \times TP}{2 \times TP + FP + FN}. \end{aligned} \quad (22)$$

Please note that the ranges of Sen, Spc, Acc and F1 values are all from 0 to 1, and a larger value indicates a better prediction performance.

2) Results: Since the proposed algorithm is the first semi-supervised sparse coding algorithm, we compared it to some unsupervised and supervised sparse coding algorithms. For the unsupervised sparse coding algorithms, we compared the proposed SSSC against the original sparse coding (SC) algorithm proposed in [2], and the popular manifold regularized

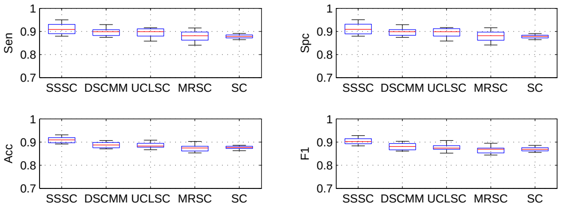

Fig. 2. Experimental results on the 1A2 inhibitor data set.

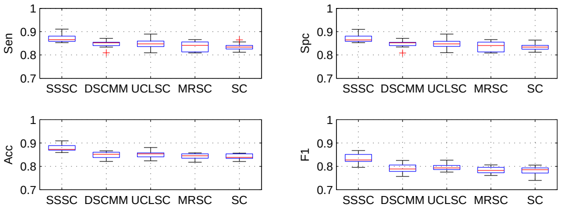

Fig. 3. Experimental results on the 2C9 inhibitor data set.

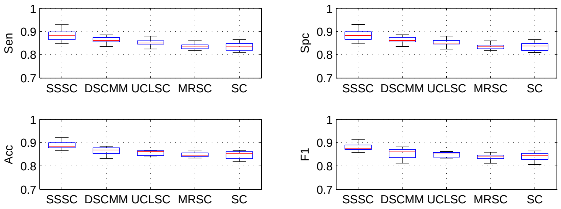

Fig. 4. Experimental results on the 2C19 inhibitor data set.

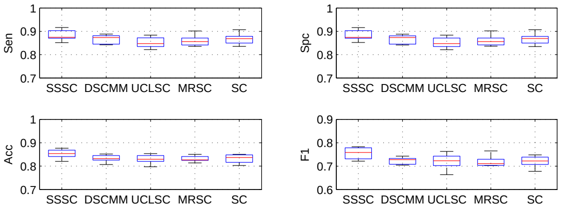

Fig. 5. Experimental results on the 2D6 inhibitor data set.

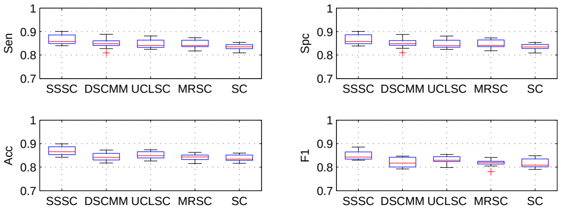

Fig. 6. Experimental results on the 3A4 inhibitor data set.

sparse coding (MRSC) algorithm proposed in [45]. For the supervised sparse coding algorithm, we compared it against the unified classifier learning and sparse coding (UCLSC) algorithm proposed in [20], and the discriminative sparse coding on multi-manifold (DSCMM) algorithm proposed in [21]. Please note that for the supervised sparse coding algorithms, it is required that all the training samples are labeled. In this case, we only used the labeled samples in the training set, while the unlabeled samples were ignored. The experiment results of four different performance measures on the five data sets are given in Fig. 2 - 6. It is clear that our SSSC algorithm consistently outperforms all other supervised and unsupervised sparse coding algorithms, namely DSCMM, UCLSC, MRSC and SC, in terms of the Sen, Spc, Acc and F1 measures. This implies that SSSC is able to learn more discriminative sparse codes to distinguish inhibitors from non-inhibitors by learning discriminative codebooks and classifiers. The performance of supervised methods, DSCMM and UCLSC, is comparable to that of unsupervised methods, MRSC and SC. We should note that only labels are used by the supervised sparse coding methods, while unsupervised methods can explore all samples. However, supervised methods include class labels to improve the discriminative ability of the sparse codes during learning, but unsupervised methods simply ignore them. Only the proposed semi-supervised method, SSSC, can use both the labels and all samples. Thus it is not surprising that it archives the best performance.

B. Wireless Sensor Fault Diagnosis

In this experiment, we evaluate the proposed algorithm on the problem of wireless sensor fault diagnosis for wireless networks [46], [47], [48], [49], [50], [51].

1) Data Set and Setup: We collected a data set of 300 samples of wireless sensors. The samples were classified to four fault types, including shock, biasing, short circuit, and shifting. We also included the normal type, making it five types in total. For each type, there are 60 samples. For each sample, we used the output signal of wireless sensors as the feature to predict its state type.

To conduct the experiment, we also employed the 10-fold cross validation. The entire data set was split to 10 folds randomly. Each fold was used as the test set in turn, and the remaining nine folds were combined and used as the training set to train the diagnosis model. Most of the training samples were unlabeled while only a small portion of the training samples was labeled. We performed the proposed algorithm to learn the codebook, classifier, sparse codes and class labels of the unlabeled training samples. The learned codebook and classifier are used to represent and classify the test samples. The classification performance is measured by the classification accuracy (Acc) for multi-class problem, which is defined as follows,

$$Acc = \frac{Number\ of\ correctly\ classified\ test\ samples}{Number\ of\ test\ samples} \tag{23}$$

The value of Acc also varies from 0 to 1, and a larger Acc indicates better classification performance.

2) Results: The boxplots of the accuracy of 10-fold cross validations are given in Fig. 7. From this figure, we can see that the proposed semi-supervised sparse coding and classification method SSSC significantly outperforms the

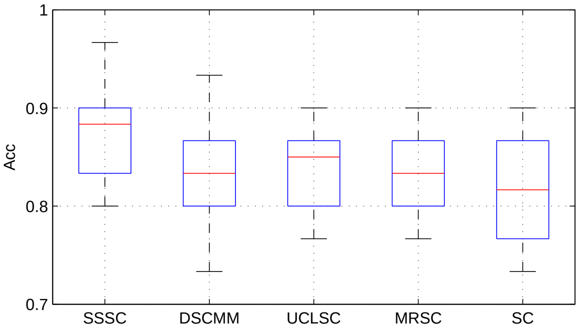

Fig. 7. Experimental results on the wireless sensor fault diagnosis data set.

other sparse coding methods on the wireless sensor fault diagnosis task. This is because our method utilizes both the labeled and unlabeled samples in learning the sparse code, while others do not effectively use such information. Again, the supervised methods DSCMM and UCLSC do not show much better improvement over the unsupervised methods MRSC and SC. It is clear that the proposed SSSC combines the advantages of both supervised and unsupervised methods. The codebook and the class labels of unlabeled samples are directly learned from training samples. Thus it is better adaptive to the data and higher classification accuracy can be achieved.

IV. Conclusion

We have proposed a sparse coding method for the semi-supervised data representation and classification task. To the best of our knowledge, this paper is the first attempt to learn sparse code on partially labeled data sets. Experimental results have shown that our proposed method SSSC are not only significantly better than state-of-the-art unsupervised sparse coding methods, but also outperforms supervised sparse coding methods. How to explore more discriminative information from both labeled and unlabeled, and combine them with our proposed semi-supervised sparse coding algorithm to further improve the learning performance appears to be an interesting direction in machine learning and pattern recognition communities.

References

[1] B. Olshausen and D. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[2] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems*, 2007, pp. 801–808.

[3] M. Zheng, J. Bu, and C. Chen, "Hessian sparse coding," *Neurocomputing*, vol. 123, pp. 247–254, 2014.

[4] L. Liu, M. Esmalifalak, Q. Ding, V. Emesih, and Z. Han, "Detecting false data injection attacks on power grid by sparse optimization," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612–621, March 2014.

[5] J. J.-Y. Wang, H. Bensmail, and X. Gao, "Feature selection and multi-kernel learning for sparse representation on a manifold," *Neural Networks*, vol. 51, pp. 9–16, 2014.

[6] S. Liu and M. Liu, "Fingerprint orientation modeling by sparse coding," in *Proceedings - 2012 5th IAPR International Conference on Biometrics, ICB 2012*, 2012, pp. 176–181.

[7] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2010, pp. 4346–4349.

[8] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2011.

[9] S. Chen, C.-Y. Zhang, and K. Song, "Recognizing short coding sequences of prokaryotic genome using a novel iteratively adaptive sparse partial least squares algorithm," *Biology Direct*, vol. 8, no. 1, 2013.

[10] K. Zhang, J. Han, T. Groesser, G. Fontenay, and B. Parvin, "Inference of causal networks from time-varying transcriptome data via sparse coding," *PLoS ONE*, vol. 7, no. 8, 2012.

[11] Z. Lei, K. Chen, H. Li, H. Liu, and A. Guo, "The gaba system regulates the sparse coding of odors in the mushroom bodies of drosophila," *Biochemical and Biophysical Research Communications*, vol. 436, no. 1, pp. 35–40, 2013.

[12] C. Wang, S. Yan, L. Zhang, and H.-J. Zhang, "Multi-label sparse coding for automatic image annotation," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, pp. 1643–1650.

[13] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. I, 2005, pp. 50–57.

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

[15] L. Shang, W. Huai, G. Dai, J. Chen, and 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)*, vol. 6064 LNCS, no. PART 2, pp. 112–119, 2010.

[16] B. Ghanem and N. Ahuja, "Sparse coding of linear dynamical systems with an application to dynamic texture recognition," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 987–990.

[17] Y. Liu and Y. Li, "Human action recognition in videos using distance image volumes and sparse coding," *Journal of Computational Information Systems*, vol. 8, no. 9, pp. 3557–3564, 2012.

[18] K. Labusch, E. Barth, and T. Martinetz, "Simple method for high-performance digit recognition based on sparse coding," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1985–1989, 2008.

[19] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 625–632.

[20] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," in *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, 2009, pp. 1033–1040.

[21] J. J.-Y. Wang, H. Bensmail, N. Yao, and X. Gao, "Discriminative sparse coding on multi-manifolds," *Knowledge-Based Systems*, vol. 54, no. 0, pp. 199 – 206, 2013.

[22] M. Bilenko, S. Basu, and R. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004*, 2004, pp. 81–88.

[23] L. Käll, J. Canterbury, J. Weston, W. Noble, and M. MacCoss, "Semi-supervised learning for peptide identification from shotgun proteomics datasets," *Nature Methods*, vol. 4, no. 11, pp. 923–925, 2007.

[24] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine Learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

[25] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings, Twentieth International Conference on Machine Learning*, vol. 2, 2003, pp. 912–919.

[26] J. Y. Ching, A. K. Wong, and K. C. Chan, "Class-dependent discretization for inductive learning from continuous and mixed-mode data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 641–651, 1995.

[27] R. Michalski, "A theory and methodology of inductive learning," *Artificial Intelligence*, vol. 20, no. 2, pp. 111–161, 1983.

[28] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning*, vol. 2006, 2006, pp. 1081–1088.

[29] Y. Chen, G. Wang, and S. Dong, "Learning with progressive transductive support vector machine," *Pattern Recognition Letters*, vol. 24, no. 12, pp. 1845–1855, 2003.

[30] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings, Twentieth International Conference on Machine Learning*, vol. 1, 2003, pp. 290–297.

[31] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2849–2856.

[32] S. Yang, X. Wang, L. Yang, Y. Han, and L. Jiao, "Semi-supervised action recognition in video via labeled kernel sparse coding and sparse l1 graph," *Pattern Recognition Letters*, vol. 33, no. 14, pp. 1951 – 6, 2012/10/15.

[33] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[34] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[35] J. Wang, F. Wang, C. Zhang, H. C. Shen, and L. Quan, "Linear neighborhood propagation and its applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 9, pp. 1600–1615, 2009.

[36] E. Simpson, M. Mahendroo, G. Means, M. Kilgore, M. Hinshelwood, S. Graham-Lorence, B. Amarneh, Y. Ito, C. Fisher, M. Michael, C. Mendelson, and S. Bulun, "Aromatase cytochrome p450, the enzyme responsible for estrogen biosynthesis," *Endocrine Reviews*, vol. 15, no. 3, pp. 342–355, 1994.

[37] C. Baj-Rossi, T. Rezzonico Jost, A. Cavallini, F. Grassi, G. De Micheli, and S. Carrara, "Continuous monitoring of naproxen by a cytochrome p450-based electrochemical sensor," *Biosensors and Bioelectronics*, vol. 53, pp. 283–287, 2014.

[38] M. Rasmussen, C. Klausen, and B. Ekstrand, "Regulation of cytochrome p450 mrna expression in primary porcine hepatocytes by selected secondary plant metabolites from chicory (cichorium intybus l.)," *Food Chemistry*, vol. 146, pp. 255–263, 2014.

[39] F. Guengerich, "Cytochrome p450s and other enzymes in drug metabolism and toxicity," *AAPS Journal*, vol. 8, no. 1, pp. E105–E111, 2006.

[40] M. Rostkowski, O. Spjuth, and P. Rydberg, "Whichcyp: Prediction of cytochromes p450 inhibition," *Bioinformatics*, vol. 29, no. 16, pp. 2051–2052, 2013.

[41] J.-L. Faulon, M. Misra, S. Martin, K. Sale, and R. Sapra, "Genome scale enzyme - metabolite and drug - target interaction predictions using the signature molecular descriptor," *Bioinformatics*, vol. 24, no. 2, pp. 225–233, 2008.

[42] J.-L. Faulon, C. Churchwell, and D. Visco Jr., "The signature molecular descriptor. 2. enumerating molecules from their extended valence sequences," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 3, pp. 721–734, 2003.

[43] J.-L. Faulon, D. Visco Jr., and R. Pophale, "The signature molecular descriptor. 1. using extended valence sequences in qsar and qspr studies," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 3, pp. 707–720, 2003.

[44] D. Rojatkar, K. Chinchkhede, and G. Sarate, "Handwritten devnagari consonants recognition using mlpnn with five fold cross validation," in *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2013*, 2013, pp. 1222–1226.

[45] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely–laplacian sparse coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3555–3561.

[46] J. Yang and Z. Fei, "HDAR: Hole detection and adaptive geographic routing for ad hoc networks," in *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*. IEEE, 2010, pp. 1–6.

[47] A. De Paola, G. Lo Re, F. Milazzo, and M. Ortolani, "Qos-aware fault detection in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[48] D.-R. Duh, S.-P. Li, and V. Cheng, "Distributed fault-tolerant event region detection of wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[49] S. Chessa and P. Santi, "Crash faults identification in wireless sensor networks," *Computer Communications*, vol. 25, no. 14, pp. 1273–1282, 2002.

[50] X. Luo, M. Dong, and Y. Huang, "On distributed fault-tolerant detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 55, no. 1, pp. 58–70, 2006.

[51] J. Yang and Z. Fei, "Bipartite graph based dynamic spectrum allocation for wireless mesh networks," in *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*. IEEE, 2008, pp. 96–101.