

# Semi-supervised classification based on subspace sparse representation

Guoxian Yu · Guoji Zhang · Zili Zhang ·  
Zhiwen Yu · Lin Deng

Received: 31 July 2012 / Revised: 29 June 2013 / Accepted: 15 October 2013 /  
Published online: 29 October 2013  
© Springer-Verlag London 2013

**Abstract** Graph plays an important role in graph-based semi-supervised classification. However, due to noisy and redundant features in high-dimensional data, it is not a trivial job to construct a well-structured graph on high-dimensional samples. In this paper, we take advantage of sparse representation in random subspaces for graph construction and propose a method called *Semi-Supervised Classification based on Subspace Sparse Representation*, SSC-SSR in short. SSC-SSR first generates several random subspaces from the original space and then seeks sparse representation coefficients in these subspaces. Next, it trains semi-supervised linear classifiers on graphs that are constructed by these coefficients. Finally, it combines these classifiers into an ensemble classifier by minimizing a linear regression problem. Unlike traditional graph-based semi-supervised classification methods, the graphs of SSC-SSR are data-driven instead of man-made in advance. Empirical study on face images classification tasks demonstrates that SSC-SSR not only has superior recognition performance with respect to competitive methods, but also has wide ranges of effective input parameters.

**Keywords** Semi-supervised classification · High-dimensional data · Graph construction · Subspaces sparse representation

---

G. Yu (✉)  
College of Computer and Information Science, Southwest University,  
Chongqing 400715, China  
e-mail: guoxian85@gmail.com

G. Yu · Z. Yu  
School of Computer Science and Engineering, South China University of Technology,  
Guangzhou 510006, China

G. Zhang  
School of Sciences, South China University of Technology, Guangzhou 510640, China

Z. Zhang  
School of Information Technology, Deakin University, Geelong, VIC 3220, Australia

L. Deng  
Department of Computer Science, George Mason University, Fairfax, VA 22030, USA

## 1 Introduction

The ever-developing information technology produces large amount of data, such as images and web texts. These data often have a large number of features, and few of them are labeled. Applying traditional supervised classifiers on limited labeled samples often results in poor generalization capability. Furthermore, it is infeasible or too expensive to manually label samples, which often requires domain-specific experts. On the other hand, valuable information in labeled samples is wasted when applying unsupervised learning on all the samples, because these label information should be used as prior information, which can be utilized to boost the performance of a classifier [7]. Given these reasons, to achieve a good classifier, both labeled and unlabeled samples should be used.

Semi-supervised learning is a learning paradigm that can make use of labeled and unlabeled instances to train a learner with good generalization ability [39]. In this paper, we focus on semi-supervised classification (SSC), and more particularly on graph-based SSC (GSSC). In GSSC, labeled and unlabeled instances can be viewed as vertices (or nodes) of a weighted graph, and the edge's weights reflect the similarity between samples [39]. Various GSSC methods were studied in the past years. Zhu et al. used Gaussian random fields and harmonic functions (GFHF) to predict the unlabeled samples in a  $k$  nearest neighborhood ( $k$ NN) graph. Based on the consistency assumption, Zhou et al. [38] studied a local and global consistent (LGC) approach on a graph. Most GSSC methods are connected with consistency assumption, which means nearby samples are more likely to have similar labels (local consistency) and samples in the same structure are more likely to share the same label (global consistency) [38]. A neighborhood graph can be used to discretely approximate the local manifold structure of samples [1]. Belkin et al. [1] introduced a regularization term on a neighborhood graph and proposed a regularization framework-manifold regularization.

All the aforementioned techniques can be formulated as label propagation on a graph under different assumptions or schemes [39]. Maier et al. [18] reported that a graph-based clustering method has two different optimal solutions on two different graphs (i.e.,  $k$ NN graph and  $\varepsilon$  neighborhood graph). Yan et al. [33] unified most dimensionality reduction approaches into the framework of graph embedding and claimed that the main difference between these methods is the graph structure. Therefore, graph structure determines the quality of the graph embedding. Similarly, graph structure also determines the effectiveness of most GSSC methods [13, 17, 39]. However, it is a hard job to construct a graph that faithfully reflects the similarity between high-dimensional samples. The reason is that the distance between samples becomes meaningless as the number of features increases [22], and the similarity metric is often damaged by noisy or redundant features of high-dimensional data. For these reasons, the performance of GSSC in high-dimensional space is not as good as in low-dimensional space. Furthermore, most GSSC methods divide the graph construction into two steps, the definition of adjacent graph and the specification of edge's weight. This separation is usually man-made in advance, and the resulting graph cannot reflect the distribution of samples faithfully. To address these problems, based on sparse representation theory [6, 30, 35], Yan et al. [32] constructed a data-driven graph by simultaneously defining the connected edges and the weight of edges. More specifically, they sought the sparse representation coefficient vectors by optimizing an  $l_1$  norm regularized minimization problem and constructed an  $l_1$  graph using these vectors.

Nevertheless, seeking these sparse coefficient vectors in the original space of high-dimensional samples is not only costing huge time, but also requiring a large number of samples. On top of that, the resulting coefficients vectors are badly affected by redundant and noisy features of high-dimensional samples. Therefore, we make use of random sub-

space and solve the  $l_1$  norm regularized problem in several random subspaces. Next, we train a semi-supervised linear classifier on a graph defined by the coefficients sought in each subspace. Finally, we combine these classifiers into an ensemble classifier via minimizing a linear regression problem. We call our method as *Semi-Supervised Classification based on Subspace Sparse Representation* (hereinafter SSC-SSR). Experimental results on facial images classification demonstrate the base classifiers of SSC-SSR are not only diverse but also accurate. And therefore, these classifiers contribute to a competent ensemble classifier.

The rest of this paper is organized as follows. Related work on sparse representation and graph-based semi-supervised classification are reviewed, and our contributions are highlighted in Sect. 2. We introduce subspace sparse representation and SSC-SSR in Sect. 3 and discuss experimental results in Sect. 4. In Sect. 5, we give conclusions and along with direction for future work.

## 2 Related work

### 2.1 Sparse representation

Sparse representation (SR) has its origin in human cognitive system [21]. Olshausen et al. [21] simulated the sparse coding mechanism of human vision system by a Bayesian framework. Recently, SR was widely used in face recognition [23,30,32], signal classification [12] and subspace clustering [8].

Suppose  $\mathbf{B} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ ,  $\mathbf{B} \in \mathbb{R}^{D \times N}$  is an over-complete base,  $\mathbf{x} \in \mathbb{R}^D$  is a new sample,  $N$  is the number of basic samples, and  $D$  is the dimensionality of samples. The original SR is formed as follows:

$$\begin{aligned} \hat{\mathbf{a}}_0 &= \arg \min \|\mathbf{a}\|_0 \\ \text{s.t. } \mathbf{B}\mathbf{a} &= \mathbf{x}, \end{aligned} \quad (1)$$

where  $\|\mathbf{a}\|_0$  is the  $l_0$  norm, which equals to the number of nonzero entries of  $\mathbf{a}$ .  $\hat{\mathbf{a}}_0$  is the sparse reconstruction coefficient vector and satisfies:

$$\mathbf{x} = \hat{a}_{0,1}\mathbf{x}_1 + \hat{a}_{0,2}\mathbf{x}_2 + \dots + \hat{a}_{0,N}\mathbf{x}_N. \quad (2)$$

In pattern recognition domain [9,23,32],  $\hat{a}_{0,i}$  ( $1 \leq i \leq N$ ) is often used as the similarity between  $\mathbf{x}$  and  $\mathbf{x}_i$ .

However, Eq. (1) is NP hard and difficult to approximate. Recent advances in SR [6,30,35] proved that if  $\hat{\mathbf{a}}_0$  is sparse enough, Eq. (1) can be equivalently transformed into an  $l_1$  norm problem:

$$\begin{aligned} \hat{\mathbf{a}}_1 &= \arg \min \|\mathbf{a}\|_1 \\ \text{s.t. } \mathbf{B}\mathbf{a} &= \mathbf{x}, \end{aligned} \quad (3)$$

where  $\|\mathbf{a}\|_1$  is the  $l_1$  norm, which sums the absolute values of  $\mathbf{a}$ . Because of noise in samples, the equal constraint in Eq. (3) cannot always hold, Eq. (3) is relaxed as follows:

$$\begin{aligned} \hat{\mathbf{a}}_1 &= \arg \min \|\mathbf{a}\|_1 \\ \text{s.t. } \|\mathbf{B}\mathbf{a} - \mathbf{x}\| &\leq \epsilon, \end{aligned} \quad (4)$$

where  $\epsilon$  can be viewed as an error tolerance. The resulting reconstruction coefficient vector  $\hat{\mathbf{a}}_1$  is sparse and discriminative [23,30,35], which means the coefficients between  $\mathbf{x}$  and some of

samples with the same label as  $\mathbf{x}$  are nonzeros, while the coefficients between  $\mathbf{x}$  and samples that have different labels from  $\mathbf{x}$  are zeros or close to zeros.

Wright et al. [30] took advantage of these sparse coefficients and introduced a method coined as sparse representation-based classification (SRC), which achieved superior performance in face recognition. Qiao et al. [23, 24] used the sparse coefficients to construct a graph and incorporated this graph in sparsity preserving projections and in sparsity preserving discriminant analysis. Similarly, Yan et al. [32] constructed an  $l_1$  graph based on these coefficients and then applied this graph into GSSC and dimensionality reduction. As well as that, Fan et al. [9] studied a penalty term on the  $l_1$  graph and introduced an approach called sparsity regularized least square classification (S-RLSC).

SR is based on the knowledge that there are over-complete basic samples which can be used to reconstruct a new sample, and it often requires  $N > D$ . However, for high-dimensional samples, it is often  $D > N$ , which cannot meet the requirement of SR and causes the small sample size (SSS) problem [4]. Therefore, Eqs. (1–4) cannot directly apply on a small number of high-dimensional samples. To address this underdetermined problem, Yan et al. [32] extended  $\mathbf{B}$  to  $\bar{\mathbf{B}} = [\mathbf{B}, \mathbf{I}_D]$ ,  $\bar{\mathbf{B}} \in \mathbb{R}^{D \times (N+D)}$ , where  $\mathbf{I}_D$  is a  $D \times D$  identity matrix. However, this extension takes much more time to solve Eq. (4). For these reasons, we solve the sparse coefficients in random subspaces whose dimensionality is much smaller than  $N$ . Next, we make use of these coefficients to construct several graphs. And finally, we combine the classifiers trained on these graphs and propose SSC-SSR.

## 2.2 Graph optimization-based semi-supervised classification

Recently, researchers recognized the importance of graph in GSSC and introduced several graph construction methods. Wu et al. [31] incorporated a discriminative regularization term into manifold regularization [1] and proposed a method called semi-supervised discriminative regularization (SSDR). Similar to GHFH and LGC, SSDR divides the definition of connectivity and the edge's weights of a graph into two separate steps. Wang et al. [29] used the  $l_2$  graph [32] and studied a linear neighborhood propagation (LNP) approach. The  $l_2$  graph is constructed by the coefficients computed in a way similar to LLE [26]. Liu et al. [17] took advantage of iterative nonparametric matrix learning to construct a symmetric adjacent graph and proposed a method called robust multi-class graph transductive classification. Fan et al. [9] exploited sparse representation coefficients to construct an  $l_1$  graph and introduced S-RLSC. Instead of taking advantage of a single graph, Yu et al. [36] first executed semi-supervised dimensionality reduction in random subspaces and then constructed multiple  $k$ NN graphs in these dimensionality reduced subspaces and studied a method called semi-supervised classification based on random subspace dimensionality reduction (SSC-RSDR). Yu et al. [37] constructed multiple  $k$ NN graphs in randomly partitioned subspaces and proposed semi-supervised ensemble classification (SSEC). LNP and SSDR are derived from a  $k$ NN graph, which is dependent on a meaningful distance metric to define the neighborhood relationship and similarity among samples. However, it is rather difficult to get a meaningful distance metric among high-dimensional samples, and the suitable  $k$  and  $\sigma$  (Gaussian kernel width) are rather difficult to choose. Consequently, the performances of these methods are not prominent in high-dimensional data. In experimental comparison with LNP and SSDR, SSC-SSR not only has higher accuracy, but also is more robust to parameters selection. In addition, we also observe that SSC-SSR can achieve better performance than SSEC and two well-known ensemble classifiers, random forest [2] and rotation forest [25].

Several distinctions between SSC-SSR and other GSSC methods based on graph optimization should be highlighted:

1. We propose to solve SR in random subspaces and empirically show that solving several SR problems in random subspaces is more efficient and effective than solving one SR problem in the original space.
2. Different from LNP [29] that uses a single  $l_2$  graph, and the approaches [9, 23, 32] exploit a single  $l_1$  graph, SSC-SSR takes advantage of several  $l_1$  graphs.
3. SSC-RSDR [36] and SSEC [37] use several  $k$ NN graphs and majority vote to ensemble base classifiers, whereas SSC-SSR utilizes several  $l_1$  graphs and linear regression to ensemble base classifiers. Our empirical study shows the proposed linear regression approach is often more effective than the majority vote method.
4. SSC-SSR avoids the model parameters selection problem in most GSSC methods (i.e.,  $k$  and  $\sigma$ ) and can be effective in a wide range of input values of parameters.

### 3 Semi-supervised classification based on subspace sparse representation

SSC-SSR mainly consists of two steps: *Subspace Sparse Representation* (SSR) and *Semi-Supervised Classification based on SSR* (SSC-SSR). The details of each step will be discussed in the following two subsections.

#### 3.1 Subspace sparse representation

The computation load of Eq. (4) increases sharply as the dimensionality of  $\mathbf{x}$  increases. Wright et al. [30] reported that a test image took several seconds to calculate the sparse coefficients. In this paper, similar to Huang et al. [12], we modify Eq. (4) as:

$$\begin{aligned} \hat{\mathbf{a}}_1 &= \arg \min \|\mathbf{B}\mathbf{a} - \mathbf{x}\|^2 + \lambda \|\mathbf{a}\|_1 \\ \text{s.t. } \mathbf{a} &\geq \mathbf{0}, \end{aligned} \quad (5)$$

where  $\lambda$  balances the trade-off between the square reconstruction error and the  $l_1$  norm. Since we want to use  $\hat{\mathbf{a}}_1$  as the similarity between  $\mathbf{x}$  and the basic samples in  $\mathbf{B}$ , we constrain  $\mathbf{a} \geq \mathbf{0}$ . The motivation to minimize Eq. (5) is twofold: (i) we want  $\hat{\mathbf{a}}_1$  as sparse as possible, and (ii) the reconstruction error is minimized.

Solving Eq. (5) takes less time than solving Eq. (4), but it still costs more than 1 s to get the sparse representation coefficients for a new sample  $\mathbf{x}$  (with  $D = 1,260$ ) based on  $N = 1,300$  basic samples (the first 1,300 samples of AR face database [20]). We used the recently issued sparse learning with efficient projections (SLEP) [16] package<sup>1</sup> to solve Eq. (5) and the widely used  $l_1$ -magic package<sup>2</sup> to solve Eq. (4) on AR and CMU PIE [28] face images databases. For more efficient solvers to  $l_1$  norm minimization algorithms, one can refer to [34] and references therein. The recorded average time costs (on Windows 7 with Intel Core2 E8400 and 4 GB RAM) are listed in Table 1.

Obviously, reducing the dimensionality of samples can greatly speed up the solutions to Eqs. (4) and (5). For high-dimensional samples,  $D$  is often larger than  $N$ . As a result, the requirement of sufficient basic samples for SR cannot be fulfilled, and the sought coefficients of SR might not be sparse [32]. To address these issues, we seek the sparse coefficients in the random subspaces of samples, whose dimensionality is much lower than that of the original space. A random subspace method (RSM) was first introduced in decision forest [11], in which multiple decision trees were trained in random subspaces and then combined into an

<sup>1</sup> <http://www.public.asu.edu/~jye02/Software/SLEP/>.

<sup>2</sup> <http://users.ece.gatech.edu/~justin/l1magic/>.

**Table 1** Time costs on different size of data (s)

| Size ( $N \times D$ ) | SLEP ( $\lambda = 0.05$ ) | $l_1$ -magic ( $\epsilon = 0.05$ ) |
|-----------------------|---------------------------|------------------------------------|
| $1,300 \times 1,260$  | 1,802.9                   | 12,284.0                           |
| $1,300 \times 100$    | 24.81                     | 6,999.0                            |
| $1,000 \times 4,096$  | 56,551.4                  | 366,892.0                          |
| $1,000 \times 200$    | 36.7                      | 10,480.2                           |

ensemble classifier. Following this work, Breiman [2] studied a method called random forest and Rodriguez et al. [25] proposed an approach coined as rotation forest. Unlike decision forest, each tree of random forest makes use of a subset of the samples and features, instead of all the samples and features. Rotation forest first randomly divides the original features into  $K$  disjoint subsets with equal size and then bootstraps 75 % samples in each subset; next, it applies principal component analysis (PCA) [7] on each subset and rearranges the eigenvectors of PCA into a rotation matrix; after that, it trains a decision tree on the training samples projected by this rotation matrix. By repeating this process  $T$  times, rotation forest gets  $T$  decision trees and combines these trees for ensemble classification. As well as that, RSM was also applied in semi-supervised classification [36, 37].

Diversity and accuracy of the base classifiers are two crucial factors in ensemble systems [3, 15, 19]. The more accurate are the base classifiers, the more accurate is the ensemble classifier. The more diverse are the base classifiers, the more accurate is the final combined classifier. Unfortunately, it is difficult to acquire both highly diverse and accurate base classifiers [3, 15, 19]. The reason is that diversity means disagreement among the predictions of samples, whereas high accuracy indicates large agreement on these predictions. How to find a compromise between accuracy and diversity to achieve an accurate ensemble classifier? To reach this target, we make use of SSR to seek accurate base classifiers and random subspaces to achieve diversity among base classifiers. SSR can be stated in three steps:

1. Randomly generate  $T$  binary indicative vectors  $\mathbf{r}_t$  ( $1 \leq t \leq T$ ),  $\mathbf{r}_t$  is constrained as:

$$\sum_{j=1}^D \mathbf{r}_{tj} = p$$

$$\text{s.t. } \mathbf{r}_{tj} \in \{0, 1\}, \quad (6)$$

where  $\mathbf{r}_{tj} = 1$  denotes the  $j$ th feature is chosen in the  $t$ th random subspace, and  $\mathbf{r}_{tj} = 0$  indicates the  $j$ th feature is not selected in that subspace.  $p$  ( $p < D$ ) is used to control how many features are selected in each subspace, it also can restrict the dimensionality of each subspace.

2. Generate  $T$  subspace sets as follows:

$$\mathbf{S}_t = \mathbf{X}(\mathbf{r}_t), \quad (7)$$

$\mathbf{S}_t = [\mathbf{s}_t^1, \mathbf{s}_t^2, \dots, \mathbf{s}_t^N]$ ,  $\mathbf{s}_t^i \in \mathbb{R}^p$  is the  $i$ th sample in  $\mathbf{S}_t$ ,  $\mathbf{s}_t^i$  includes the selected features of  $\mathbf{x}_i$  specified by  $\mathbf{r}_t$ .  $\mathbf{S}_t$  can be viewed as the  $t$ th random subspace set. Therefore, we get  $T$  random subspace sets  $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_T\}$ .

3. Compute the sparse representation coefficients on  $\mathbf{S}_t$ . Let  $\mathbf{Z}_t^i = [\mathbf{s}_t^1, \mathbf{s}_t^2, \dots, \mathbf{s}_t^{i-1}, \mathbf{s}_t^{i+1}, \dots, \mathbf{s}_t^N]$  and then solve the following minimization problem:

$$\hat{\mathbf{a}}_i = \arg \min \left\| \mathbf{Z}_t^i \mathbf{a} - \mathbf{s}_t^i \right\|^2 + \lambda \|\mathbf{a}\|_1$$

$$\text{s.t. } \mathbf{a} \geq \mathbf{0}. \quad (8)$$

Next, reform  $\hat{\mathbf{a}}_i$  as:

$$\hat{\mathbf{a}}_i = [\hat{a}_{i,1}, \hat{a}_{i,2}, \dots, \hat{a}_{i,i-1}, 0, \hat{a}_{i,i+1}, \dots, \hat{a}_{i,N}] \quad (9)$$

Finally, set  $\mathbf{W}_{ij}^t$  ( $\mathbf{W}^t \in \mathbb{R}^{N \times N}$ ) as follows:

$$\mathbf{W}_{ij}^t = \hat{a}_{i,j}. \quad (10)$$

$\mathbf{W}_{ij}^t$  can be viewed as the weight of the edge between  $s_i^j$  and  $s_t^j$ . By iterating on  $N$  projected samples in  $\mathbf{S}_t$ , we can construct a graph (specified by  $\mathbf{W}^t$ ) in the  $t$ th random subspace. Next, we train a graph-based semi-supervised classifier on this graph. By repeating these steps on other subspace sets,  $T$  classifiers can be trained on  $T$  subspace sets of  $\mathbf{S}$ . These classifiers will serve as base classifiers of SSC-SSR.

Since the graphs of SSC-SSR are defined in subspaces, they are less affected by redundant or noisy features, and more faithful to the distribution of samples projected in these subspaces. According to [8,30], if different objects exist in different subspaces or a union of linear subspaces, a new sample can be linearly reconstructed by other samples, which have the same label with this sample. In other words, the weights, between the new sample and the samples, whose labels are different from this sample, are zeros or nearly zeros. This fact indicates the graph, defined by sparse representation coefficients, is sparse and discriminative. Although these requirements of SR might not be fully met in random subspaces, we can still expect the graphs defined by these coefficients are discriminative in some extent. And this expectation will be corroborated in our experiments. In the meantime, semi-supervised classifiers trained on these graphs also hold the diversity across random subspaces. Inspired by stochastic discrimination theory [14], it is straightforward and promising to combine these base classifiers into a competent ensemble classifier. In this way, we not only avoid the huge time cost involving with sparse representation in the original space, but also enhance the performance of semi-supervised classification. Our empirical study shows the  $l_1$  graphs constructed in subspaces are more discriminative than the  $l_1$  graph constructed in the original space (c.f. Fig. 3 in Sect. 4).

SSR has some intrinsic differences from the methods based on SR in the original space. Qiao et al. [23,24], Yan et al. [32] and Fan et al. [9] used SR in the original space or PCA projected subspace to get a weight matrix. In contrast, SSR uses  $T$  random subspaces and gets  $T$  weight matrices more efficiently, since  $p \ll D$ . SRC [30] predicts the label of a test sample as the label of samples, which have least reconstruction error on this sample. Nevertheless, SRC only makes use of labeled samples. Wright et al. [30] proposed a variant of SRC by averaging the results of the random projections of original features. But this variant still cannot use unlabeled samples. On the contrary, the proposed SSR exploits both labeled and unlabeled samples.

One possible issue of SSR is that the discriminative features may not be included in the random subspaces. In fact, this issue also exists in RSM [11,36], but it seldom happens in applying SSC-SSR on high-dimensional data. There are four reasons:

First, suppose  $d$  is the number of discriminative features, then the probability that no discriminative features are selected in  $T$  random subspaces is bounded by  $((1 - \frac{d}{D})^p)^T$ . Suppose  $d = \frac{2}{100}D$ , if  $p = 50$  and  $T = 10$ , this probability is smaller than  $4.10 \times 10^{-5}$ ; if  $p = 100$  and  $T = 20$  (the setting of our experiments), this probability is smaller than  $2.83 \times 10^{-18}$ .

Second, it is proved that an ensemble classifier can achieve the accuracy no worse than the average accuracy of its base classifiers [3]. In this way, it avoids the risk of choosing a bad model or selecting less informative important features.

Third, if some of the discriminative features are excluded in random subspaces, diverse base classifiers still can be achieved. Although these classifiers are not as accurate as those trained on the discriminative features, according to stochastic discrimination theory [14], they still can be combined into a competitive ensemble classifier with good generalization ability.

Fourth, Wright et al. [30] stated and observed that SRC with random features could get similar performance as with other dimensionality reduction techniques, as long as the number of features is sufficient to get sparse coefficients, and this number is often no more than 100. This statement inspires us to solve sparse coefficients in the subspace, which has sufficient number of features to get these coefficients. A feature selection method can be used to generate a feature subset and be viewed as a dimensionality reduction approach. But the selected features may be biased by the selection criterion and only one  $l_1$  graph can be constructed on the selected feature subset.

For these reasons, we seek sparse representation coefficients in random subspaces and make use of these coefficients to construct several graphs for SSC-SSR.

### 3.2 Semi-supervised ensemble classification based on subspace sparse representation

In this subsection, instead of training a semi-supervised classifier on a single graph in the original space, we train several classifiers on several graphs constructed in random subspaces, these graphs are constructed with the weight matrices  $\mathbf{W}^t$  ( $1 \leq t \leq T$ ) specified in Eq. (10). Suppose there are  $N = l + u$  samples  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$ ,  $\mathbf{x}_i \in \mathbb{R}^D$ , where the first  $l$  samples are labeled and the left are unlabeled,  $y_i$  is the label of  $\mathbf{x}_i$  and  $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ . Here, we consider a *semi-supervised linear classifier* (SLC) as the base classifier of SSC-SSR. The objective function of SLC is to minimize:

$$\Psi(f) = \Omega(f, \mathbf{x}, \mathbf{y}) + \alpha \|f\|_f^2, \quad (11)$$

where the first term is the empirical loss on the labeled samples, and the second term is the regularization on the labeled and unlabeled samples,  $\alpha$  balances the importance of two terms.  $f(\mathbf{x})$  is specified as:

$$f(\mathbf{x}) = \mathbf{P}^T \mathbf{x} + \mathbf{b}, \quad (12)$$

where  $\mathbf{P} \in \mathbb{R}^{D \times C}$ ,  $\mathbf{b} \in \mathbb{R}^C$ . To compute the empirical loss on labeled samples, we extend the label vector  $\mathbf{y}$  into a label matrix  $\mathbf{Y} \in \mathbb{R}^{N \times C}$  as:

$$\mathbf{Y}_{ic} = \begin{cases} 1, & \text{if } y_i = c \\ 0, & \text{otherwise} \end{cases} \quad (13)$$

Then, the defined loss function on  $l$  labeled samples is:

$$\begin{aligned} \Omega(f, \mathbf{X}, \mathbf{Y}) &= \sum_{i=1}^l \left\| \mathbf{P}^T \mathbf{x}_i + \mathbf{b} - \mathbf{Y}_i \right\|^2 \\ &= \sum_{i=1}^N \left( \left( \mathbf{P}^T \mathbf{x}_i + \mathbf{b} - \mathbf{Y}_i \right)^T \mathbf{H}_{ii} \left( \mathbf{P}^T \mathbf{x}_i + \mathbf{b} - \mathbf{Y}_i \right) \right) \\ &= \text{tr} \left( \left( \mathbf{P}^T \mathbf{X} + \mathbf{b} \mathbf{1}^T - \mathbf{Y}^T \right) \mathbf{H} \left( \mathbf{P}^T \mathbf{X} + \mathbf{b} \mathbf{1}^T - \mathbf{Y}^T \right)^T \right), \end{aligned} \quad (14)$$

where  $tr()$  means matrix trace operation,  $\mathbf{1} \in \mathbb{R}^N$  with all elements are ones, and  $\mathbf{H}$  is a diagonal matrix with  $\mathbf{H}_{ii} = 1$  if  $i \leq l$ ,  $\mathbf{H}_{ii} = 0$  otherwise.  $\|f\|_I^2$  can be computed as:

$$\begin{aligned}
 \|f\|_I^2 &= \frac{1}{2} \sum_{i,j=1}^N \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \mathbf{W}_{ij} \\
 &= \frac{1}{2} \sum_{i,j=1}^N \|\mathbf{P}^T \mathbf{x}_i - \mathbf{P}^T \mathbf{x}_j\|^2 \mathbf{W}_{ij} \\
 &= tr \left( \mathbf{P}^T \sum_{i=1}^N (\mathbf{x}_i \mathbf{W}_{ii} \mathbf{x}_i^T) \mathbf{P} - \mathbf{P}^T \sum_{i,j=1}^N (\mathbf{x}_i \mathbf{W}_{ij} \mathbf{x}_j^T) \mathbf{P} \right) \\
 &= tr \left( \mathbf{P}^T \mathbf{X} (\mathbf{A} - \mathbf{W}) \mathbf{X}^T \mathbf{P} \right) \\
 &= tr \left( \mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P} \right), \tag{15}
 \end{aligned}$$

where  $\mathbf{A}$  is a diagonal matrix with  $\mathbf{A}_{ii} = \sum_{j=1}^N \mathbf{W}_{ij}$ ,  $\mathbf{L}$  is a graph Laplacian matrix [5].

Based on Eqs. (14) and (15), the objective function of SLC is rewritten as:

$$\begin{aligned}
 \psi(\mathbf{X}, \mathbf{P}, \mathbf{b}) &= tr \left( \left( \mathbf{P}^T \mathbf{X} + \mathbf{b} \mathbf{1}^T - \mathbf{Y}^T \right) \mathbf{H} \left( \mathbf{P}^T \mathbf{X} + \mathbf{b} \mathbf{1}^T - \mathbf{Y}^T \right)^T \right) \\
 &\quad + \alpha tr \left( \mathbf{P}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P} \right). \tag{16}
 \end{aligned}$$

The target of minimizing the first term of Eq. (16) is to seek  $f(\mathbf{x})$ , whose outputs are similar to the original labels. The motivation to minimize the second term is that similar samples should have similar outputs. Put it in another way, if two samples  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are quite similar to each other, meaning  $\mathbf{W}_{ij}$  is big,  $f(\mathbf{x}_i)$  and  $f(\mathbf{x}_j)$  should have similar outputs; otherwise, there is a heavy penalty between them and causing a big loss.

Equation (16) can be solved by taking partial derivative of  $\psi(\mathbf{X}, \mathbf{P}, \mathbf{b})$  with respect to  $\mathbf{P}$  and  $\mathbf{b}$  as:

$$\frac{\partial \psi}{\partial \mathbf{P}} = 2 \mathbf{X} \mathbf{H} \left( \mathbf{X}^T \mathbf{P} + \mathbf{1} \mathbf{b}^T - \mathbf{Y} \right) + 2 \alpha \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{P} \tag{17}$$

$$\frac{\partial \psi}{\partial \mathbf{b}} = 2 \left( \mathbf{P}^T \mathbf{X} + \mathbf{b} \mathbf{1}^T - \mathbf{Y}^T \right) \mathbf{H} \mathbf{1}. \tag{18}$$

Let  $\frac{\partial \psi}{\partial \mathbf{P}} = \mathbf{0}$  and  $\frac{\partial \psi}{\partial \mathbf{b}} = \mathbf{0}$ , we get

$$\mathbf{P} = \left( \mathbf{X} \mathbf{H}_c \mathbf{X}^T + \alpha \mathbf{X} \mathbf{L} \mathbf{X}^T \right)^{-1} \mathbf{X} \mathbf{H}_c \mathbf{Y} \tag{19}$$

$$\mathbf{b} = \frac{(\mathbf{Y}^T - \mathbf{P}^T \mathbf{X}) \mathbf{H} \mathbf{1}}{N} \tag{20}$$

where  $\mathbf{H}_c$  is:

$$\mathbf{H}_c = \mathbf{H} - \frac{(\mathbf{H} \mathbf{1} \mathbf{1}^T \mathbf{H}^T)}{N}. \tag{21}$$

If  $N < D$ , a regularization term is added to avoid the singular problem as:

$$\mathbf{P} = \left( \mathbf{X} \mathbf{H}_c \mathbf{X}^T + \alpha \mathbf{X} \mathbf{L} \mathbf{X}^T + \beta \mathbf{I}_D \right)^{-1} \mathbf{X} \mathbf{H}_c \mathbf{Y}, \tag{22}$$

where  $\beta > 0$  is often set to a small value.

From Eqs. (19) and (20), we can observe  $\mathbf{P}$  and  $\mathbf{b}$  depend on  $\mathbf{L}$ .  $\mathbf{L}$  is derived from  $\mathbf{W}$ , which is often specified by Gaussian heat kernel similarity with respect to a  $k$ NN graph [1, 38, 40]. Fan et al. [9] and Yan et al. [32] used sparse representation coefficients in original space to specify  $\mathbf{W}$ . However, the noisy and redundant features of high-dimensional samples often distort the weight matrix  $\mathbf{W}$  and result in an unstable and inaccurate solution. Most GSSC methods perform well on low-dimensional samples, but they do not perform well on high-dimensional data. For these reasons, based on SSR in Sect. 3.1, we construct  $T$  graphs defined by the coefficients sought by SSR and then train a SLC on each graph. The next step is how to combine these  $T$  classifiers into an ensemble classifier.

Here we introduce a linear regression model to ensemble these  $T$  classifiers:

$$\begin{aligned} \arg \min_{\omega_c} \sum_{t=1}^T \sum_{i=1}^l (\omega_{ct} f_t(\mathbf{x}_i, c) - \mathbf{y}_{ic})^2 + \mu \|\omega_c\|^2 \\ = \arg \min_{\omega_c} \|\mathbf{F}_{lc} \omega_c - \mathbf{Y}_{lc}\|^2 + \mu \|\omega_c\|^2, \end{aligned} \quad (23)$$

where  $\omega_{ct}$  is the weight of the  $t$ th classifier with respect to the  $c$ th label,  $f_t(\mathbf{x}_i, c)$  is the  $c$ th entry of  $f_t(\mathbf{x}_i)$ .  $\mathbf{F}_c = [\mathbf{f}_{1c}, \mathbf{f}_{2c}, \dots, \mathbf{f}_{Tc}]^T \in \mathbb{R}^{N \times T}$ ,  $\mathbf{f}_{lc}$  is the predicted likelihood vector of  $\mathbf{f}_l$  with respect to the  $c$ th label on  $N$  samples, and  $\mathbf{F}_{lc}$  is the first  $l$  rows of  $\mathbf{F}_c$ .  $\mathbf{Y}_{lc}$  is the first  $l$  rows with respect to the  $c$ th column of  $\mathbf{Y}$ .  $\mu$  is set to balance the importance of the first term and the second term. Our motivation in using  $\omega_{ct}$  is that different classifiers may have different weights on the  $c$ th label. The analytical solution of Eq. (23) is:

$$\omega_c = \left( \mathbf{F}_{lc}^T \mathbf{F}_{lc} + \mu \mathbf{I}_T \right)^{-1} \mathbf{F}_{lc}^T \mathbf{Y}_{lc}, \quad (24)$$

where  $\mathbf{I}_T$  is a  $T \times T$  identity matrix. Let  $\mathbf{M} = [\omega_1, \omega_2, \dots, \omega_C]^T \in \mathbb{R}^{T \times C}$ , then the ensemble predicted likelihood vector is:

$$f(\mathbf{x}) = \sum_{t=1}^T \mathbf{M}_t \circ f_t(\mathbf{x}), \quad (25)$$

where  $\circ$  is the element-wise multiplication operator. We transform the predicted likelihood vector  $f(\mathbf{x})$  into a label as:

$$l(\mathbf{x}) = \arg \max_{1 \leq c \leq C} f(\mathbf{x}, c), \quad (26)$$

where  $f(\mathbf{x}, c)$  indicates the  $c$ th element of  $f(\mathbf{x})$ . The procedure of SSC-SSR is described in Algorithm 1.

Although the requirements of SR [8, 30] cannot be fully met in the random subspaces and the coefficients of SSR maybe not as discriminative as in the original space, it is still able to achieve accurate classifiers in these subspaces. These base classifiers might not be as accurate as the classifier trained in the original space, but they hold diversity across random subspaces. From stochastic discrimination theory [14], these  $T$  base classifiers  $f_t(\mathbf{x})$  can be combined into a more accurate ensemble classifier. This advantage will be confirmed in our following experiments. From the procedure of SSC-SSR, we can find our SSC-SSR does not need to specify  $k$  and  $\sigma$  in advance. So SSC-SSR, in some extent, mitigates the parameters selection problem associated with most GSSC methods.

---

**Algorithm 1** SSC-SSR: Semi-Supervised Classification based on Subspace Sparse Representation
 

---

**Input:** $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N], \mathbf{y} = [y_1, y_2, \dots, y_T], \mathbf{x}, C, p, T$ **Output:** $l(\mathbf{x})$ **Training:**

- 1: **for**  $t = 1$  to  $T$  **do**
- 2: Randomly generate an indicative vector  $\mathbf{r}_t$  using Eq. (6) and project  $\mathbf{X}$  into  $\mathbf{S}_t$  using Eq. (7).
- 3: Get the sparse representation weight matrix  $\mathbf{W}^t$  on  $\mathbf{S}^t$  using Eqs. (8-10).
- 4: Construct the graph Laplacian matrix  $\mathbf{L}^t$  on  $\mathbf{S}^t$  using  $\mathbf{W}^t$ .
- 5: Solve  $f_t(\mathbf{s}_t) = \mathbf{P}_t^T \mathbf{s}_t + \mathbf{b}_t$  on  $\mathbf{S}^t$  using Eqs. (17-18) with  $\mathbf{L}$  replaced by  $\mathbf{L}^t$ .
- 6: **end for**
- 7: Compute  $\mathbf{M}$  using Eq. (24).

**Testing:**

- 8: **for**  $t = 1$  to  $T$  **do**
  - 9: Project  $\mathbf{x}$  into the  $t$ -th random subspace using indicative vector  $\mathbf{r}_t$ ,  $\mathbf{x}_t = \mathbf{x}(\mathbf{r}_t)$
  - 10: Compute  $f_t(\mathbf{x}_t)$  using  $\mathbf{P}_t$  and  $\mathbf{b}_t$ .
  - 11: **end for**
  - 12: Ensemble  $[f_t(\mathbf{x}_t)]_{t=1}^T$  with  $\mathbf{M}$  using Eq. (25) and get the final prediction using Eq. (26).
- 

### 3.3 Complexity analysis

Besides the sought of sparse representation coefficients, the time complexity of SLC and SSC-SSR are dominated by matrix multiplication and matrix inverse in Eq. (19). The complexity of matrix multiplication is  $O(ND^2 + N^2D)$  and the complexity of matrix inverse is  $O(D^3)$ , so the time complexity of SLC is  $O(N^2D + ND^2 + D^3)$ . SSC-SSR is based on SLC in  $T$  random subspaces, its time complexity is  $O(T(N^2p + Np^2 + p^3))$ . Since  $Tp \approx D$  and the computation of sparse representation coefficients in  $T$  subspaces often takes much less time than that in one original space (see Tables 1 and 4), SSC-SSR has less time complexity than SLC.

## 4 Experiments

### 4.1 Experiments setup

In this section, we conduct experiments on four public available face databases to experimentally compare the effectiveness of SSC-SSR with other methods. The comparing methods are LNP [29], SSDR [31], S-RLSC [9], SSEC [37] and two supervised ensemble classifiers random forest [2] and rotation forest [25]. The first three are semi-supervised classifiers, and the last three are ensemble approaches. We also report the results of the proposed SLC in the original space. SSC-RSDR constructs several  $k$ NN graphs in the subspaces projected by semi-supervised dimensionality reduction and integrates semi-supervised nonlinear classifiers trained on these graphs. SSC-SSR and SSEC directly construct graphs in the random subspaces and ensemble semi-supervised linear classifiers trained on these graphs. Thus, we do not compare SSC-SSR and SSEC with SSC-RSDR.

The face databases are AR [20], extended Yale B (yaleB) [10], CMU PIE [28] and ORL [27]. AR face database contains over 4,000 color face images from 126 persons (70 men and 56 women). In the experiments, 2,600 images of 100 persons (50 men and 50 women) were used. Before the experiments, we aligned and resized these images into  $42 \times 30$  pixels and then normalized them into 8 bit grayscale; thus, each image is a vector in 1,260-dimensional

**Table 2** Notations used in the experiments

| Notation | Means                               |
|----------|-------------------------------------|
| $N$      | Number of training samples          |
| $Nt$     | Number of testing samples           |
| $C$      | Number of classes                   |
| $D$      | Dimensionality of original data     |
| $T$      | Number of base classifiers          |
| $p$      | Dimensionality of random subspace   |
| $n$      | Number of labeled images per person |
| $k$      | Neighborhood size                   |
| $\sigma$ | Gaussian kernel width               |

space. Next, we randomly selected 13 images per person as the training set and the rest as testing set. YaleB contains 21,888 single light source images of 38 individuals each captured under 576 viewing conditions ( $9 \times 64$  illumination conditions). We chose the near frontal subset for our experiments and resized these images into  $32 \times 32$  pixels, with 256 gray level per pixel, so each image is a 1,024-dimensional vector. We then randomly selected half number of images per person as the training set and the rest as testing set. CMU PIE face database contains 41,368 face images from 68 individuals. The face images were captured under varying pose, illumination and expression conditions. Before the experiments, we resized the images to  $64 \times 64$  pixels, and each is an 8 bit gray scale image, so each image is a point in the 4,096-dimensional space. In the experiments, we selected the Pose27 subset and then randomly chose half number of images of the same individual as training set and the remaining as testing set. ORL is consisted of 40 individuals with each person 10 images. Before the experiments, we aligned and resized these images into  $32 \times 32$  pixels in grayscale. And then we selected 6 images per person as training set and the left as testing set. The last three processed face databases can be downloaded from Deng Cai's homepage.<sup>3</sup> Note, in the first two datasets,  $N > D$ , and in the last two datasets,  $N < D$ .

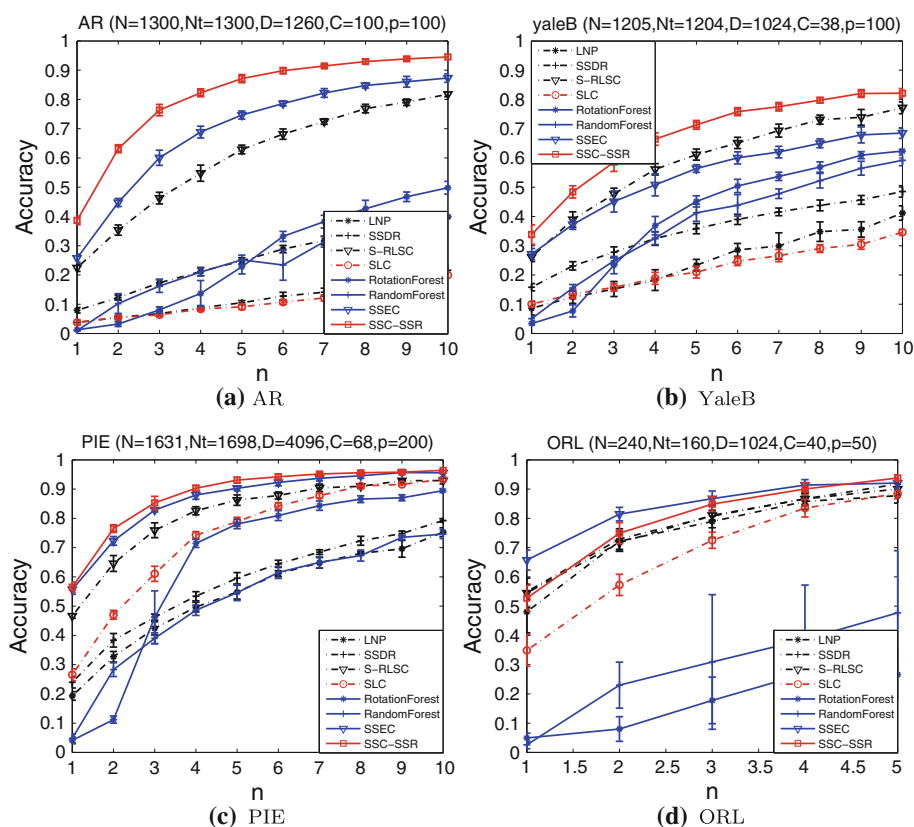
In the following experiments, we use the notations described in Table 2. In these experiments, unless extra specified,  $T$  is set to 20,  $n$  is set to 10,  $p$  is set to 100,  $k$  is set to 5, and  $\sigma$  is set as the average of Euclidean distance of the training samples. This specification of  $k$  and  $\sigma$  is commonly used in other references [17, 38].  $\lambda$ ,  $\alpha$  and  $\beta$  are all set to 0.05,  $\mu$  is set as 0.01.  $\gamma_D$  in SSDR [31] is set to 0.9, we searched the best value of  $\gamma_D$  between 0 and 1 with step size 0.1 and observed  $\gamma_D = 0.9$  gave the best results. The kernel Gram matrix  $\mathcal{K}$  used in S-RLSC and SSDR is specified by a Gaussian heat kernel. The number of disjoint subsets in rotation forest is equal to 20, and the number of decision trees in rotation forest and random forest is also equal to  $T$ . We use the recently issued SLEP<sup>4</sup> to get the sparse representation coefficients. To reduce the random impact, all the reported results (along with standard deviations) are the average of 20 independent runs. In each run, the training set and testing set are randomly partitioned and the labeled samples in the training set are randomly selected.

#### 4.2 Recognition accuracy in different data sizes

In order to investigate the performance of SSC-SSR on different values of  $n$ , four experiments are conducted on AR, yaleB, PIE with  $n$  increasing from 1 to 10 and on ORL with  $n$  increasing

<sup>3</sup> <http://www.cad.zju.edu.cn/home/dengcai/Data/data.html>.

<sup>4</sup> <http://www.public.asu.edu/~jye02/Software/SLEP/>.



**Fig. 1** Accuracy versus  $n$ . **a** AR. **b** YaleB. **c** PIE. **d** ORL

from 1 to 5. Note, the other images in the training set are used as unlabeled samples for semi-supervised classifiers. The recorded results are plotted in Fig. 1.

In Fig. 1, all the methods have increasing accuracies with  $n$  rising, and SSC-SSR achieves the best performance in most times. S-RLSC takes advantage of SR in the original space and gets better performance than LNP and SSDR. This observation indicates that the  $l_1$  graph can reflect the similarity relationship between samples much better than the  $l_2$  graph or  $k$ NN graph. However, S-RLSC still loses to SSC-SSR. SSC-SSR is also based on SR, but it seeks the coefficients more efficiently in random subspaces than in the original space. SLC is trained on the single  $l_1$  graph in the original space, it always cannot outperform SSC-SSR. This is because the sparse coefficients in the original space are badly distorted by the noisy features. These observations show that SSR is more effective than SR in the original space. The gap between SSC-SSR and S-RLSC on AR is more obvious than that on other facial databases. The reason is that there is more noise (or large occlusions) in the images of AR than that of other facial databases. This fact demonstrates our SSC-SSR is more robust to noisy features than S-RLSC. Rotation forest and random forest only make use of labeled samples and lose to SSC-SSR and SSEC in most cases, this fact verifies that unlabeled data can be used to boost a classifier performance. Both SSC-SSR and SSEC utilize the same number of graphs, SSC-SSR performs better than SSEC on AR, yaleB and PIE. This observation shows the  $l_1$  graphs sought in subspaces are also discriminative. SSC-SSR loses to SSEC on ORL, one

**Table 3** Regression versus majority vote

| Dataset | Method        | $n = 2$                 | $n = 4$          | $n = 6$                 | $n = 8$                 | $n = 10$                |
|---------|---------------|-------------------------|------------------|-------------------------|-------------------------|-------------------------|
| AR      | Regression    | 63.15 $\pm$ 1.36        | 82.28 $\pm$ 1.35 | <b>89.84</b> $\pm$ 1.06 | <b>92.95</b> $\pm$ 0.76 | <b>94.54</b> $\pm$ 0.88 |
|         | Majority vote | 64.15 $\pm$ 2.14        | 81.51 $\pm$ 1.54 | 88.70 $\pm$ 0.74        | 91.81 $\pm$ 0.94        | 93.26 $\pm$ 0.85        |
| YaleB   | Regression    | 48.39 $\pm$ 2.10        | 66.45 $\pm$ 2.11 | <b>75.83</b> $\pm$ 1.37 | <b>79.74</b> $\pm$ 0.88 | <b>82.13</b> $\pm$ 1.61 |
|         | Majority vote | <b>51.08</b> $\pm$ 2.25 | 65.78 $\pm$ 1.83 | 74.15 $\pm$ 1.21        | 77.78 $\pm$ 1.07        | 79.86 $\pm$ 2.15        |

The better performance are in bolded (Statistical significance examined via pairwise  $t$  test at 95 %)

possible reason is that SR often requires a large number of basic samples and ORL has a relative small number of training samples. The performance of SLC on PIE and ORL is much better than that on AR and yaleB. The reason is that,  $N < D$  on PIE and ORL, SLC is based on Eq. (22) and  $\beta \mathbf{I}_D$  has larger effect than  $\alpha \mathbf{X} \mathbf{L} \mathbf{X}^T$ . This fact also indicates our motivation to seek sparse representation in random subspaces is rational. These experimental results demonstrate that SSC-SSR can make use of random subspaces and labeled samples much better than other comparing methods.

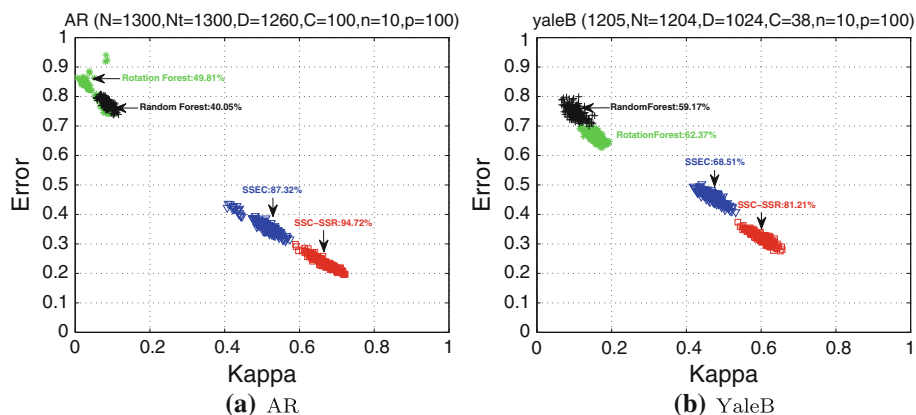
LNP optimizes an objective function (similar to LLE [26]) to capture the local linear structure among samples, but it loses to other approaches on AR, yaleB and PIE. This indicates that optimizing a  $k$ NN graph might not help to get a competent classifier. SSDR [31] incorporates a discriminate term to enhance its discriminate ability. Nevertheless, SSDR needs to set the trade-off between the discriminative term and other terms in advance. We had tried to get an optimal solution of SSDR by optimizing this tradeoff, but the experimental results of SSDR, in most cases, still could not outperform other methods. The reason is that SSDR is mainly determined by the local smoothness term, this term is defined on a  $k$ NN graph, which is distorted by noisy and redundant features of high-dimensional facial images.

To investigate the difference between regression-based classifier ensemble in Eq. (23) and majority vote-based classifier ensemble, the latter one is used in SSEC [see Eq. (25) in [37]], we conduct another two experiments on AR and yaleB. In these experiments, 20 base classifier SLCs are combined by regression and majority vote, respectively. The recorded results are reported in Table 3. From this table, we can observe that SSC-SSR based on regression performs better than the SSC-SSR based on majority vote as the number of labeled samples increases. The reason is that the regression-based model depends on the labeled samples to get  $\mathbf{M}$  in Eq. (25), and more labeled samples help to result in a better estimation of  $\mathbf{M}$ . These results demonstrate that the regression-based ensemble approach used in SSC-SSR is effective and more competent than the majority vote technique.

#### 4.3 Diversity versus accuracy analysis

In this subsection, we conduct experiments to explore the diversity and accuracy of the base classifiers of SSC-SSR and compare them with the base classifiers of random forest, rotation forest and SSEC. Here, we adopt the widely used Kappa-error diagram [19, 36, 37] to visually analyze the pairwise diversity and accuracy of the base classifiers of an ensemble classifier. In this diagram, the  $x$ -axis denotes the diversity between two base classifiers and the  $y$ -axis represents the average error of these two classifiers.

Let  $\mathbf{A} \in \mathbb{R}^{C \times C}$  be a coincidence matrix of two base classifiers  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$ ,  $\mathbf{A}(c_1, c_2)$  is the proportion of testing samples that  $f_s(\mathbf{x})$  predict as  $c_1$  and  $f_t(\mathbf{x})$  predict as  $c_2$ . So  $\sum_{c=1}^C \mathbf{A}(c, c)$  is the observed agreements between  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$ , and the diversity between  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$  can be computed as:



**Fig. 2** Kappa-error graph. **a** AR. **b** YaleB

$$\kappa_{st} = \frac{\sum_{c=1}^C A(c, c) - ABC}{1 - ABC}, \quad (27)$$

where ABC means ‘agreement-by-chance’ between  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$ . ABC can be calculated as:

$$ABC = \sum_{c_1=1}^C \left( \sum_{c_2=1}^C A(c_1, c_2) \right) \left( \sum_{c_2=1}^C A(c_2, c_1) \right) \quad (28)$$

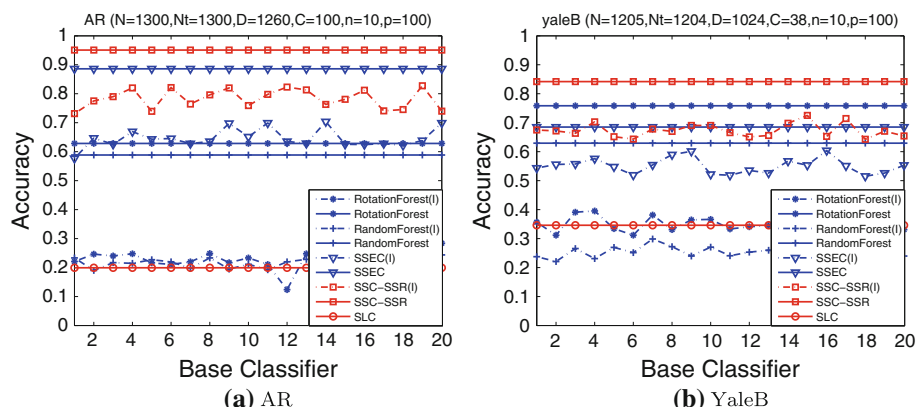
The average error  $e_{st}$  between  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$  is:

$$e_{st} = \frac{e_s + e_t}{2}, \quad (29)$$

where  $e_s$  and  $e_t$  are the error rates of  $f_s(\mathbf{x})$  and  $f_t(\mathbf{x})$  on the testing samples. Thus, the pairwise accuracy can be viewed as  $1 - e_{st}$ . From the definition of  $\kappa$  and  $e$ , we can observe that small values of  $\kappa$  indicate high diversity between base classifiers, and small values of  $e$  mean high accuracy between base classifiers. With the above preparations, we set  $T = 20$  and  $n = 10$  on AR and yaleB and then record the Kappa values and error rates of these base classifiers. The results are plotted in Fig. 2.

In Fig. 2, SSC-SSR achieves the highest accuracies, 94.72 % on AR and 81.21 % on yaleB. The base classifiers of SSC-SSR are not as diverse as those of other ensemble classifiers, but they are more accurate and hold the diversity across different subspaces, they contribute to a competent ensemble classifier. The base classifiers of rotation forest have lower accuracies than the base classifiers of random forest on AR, they have higher diversities than the latter ones, so rotation forest gain higher accuracy than random forest on AR. For the experiment on yaleB, the base classifiers of rotation forest and random forest have some overlaps on the Kappa values, since the base classifiers of rotation forest get higher accuracies than those of random forest, rotation forest gets higher accuracy than random forest. These facts show that both the diversity and accuracy of base classifiers should be considered in the design of an ensemble classifier.

To further investigate SSC-SSR, we record the accuracies of 20 base classifiers of SSC-SSR and the accuracy of SSC-SSR. The experimental results on AR and yaleB are reported in Fig. 3. In the figure, the broken lines with a suffix (I) are individual base classifiers of the



**Fig. 3** Accuracy of base classifiers versus ensemble classifier. **a** AR. **b** YaleB

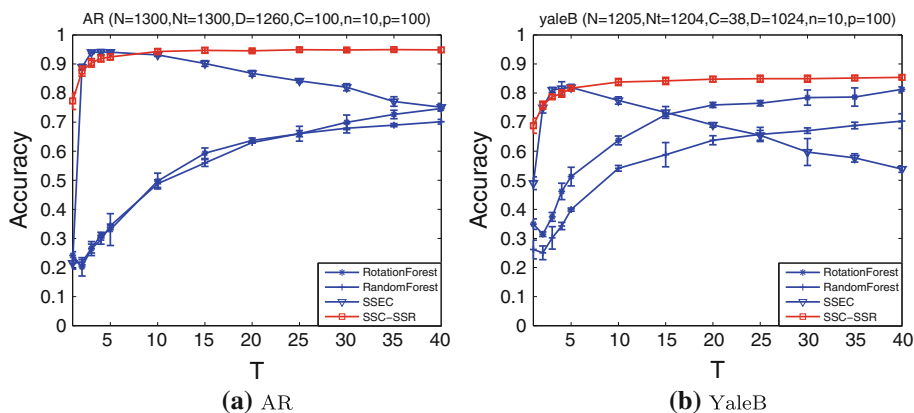
corresponding ensemble classifiers (solid lines without such a suffix). We also include the results of SLC to study the difference between SR in the subspace and SR in the original space.

In Fig. 3, we can observe SSC-SSR always performs better than any of its base classifiers, this observation indicates that the regression-based ensemble in Eq. (23) is effective and these base classifiers hold the diversity across subspaces. The base classifiers of SSEC have lower accuracies than those of SSC-SSR, so SSEC always loses to SSC-SSR. This phenomenon shows that the  $l_1$  graphs, defined by coefficients sought by SSR, are more discriminative than  $k$ NN graphs. Another conclusion can be made is that with the same diversity across random subspaces, more accurate base classifiers can result in a more competent ensemble classifier. The base classifiers of SSC-SSR always have higher accuracies than SLC in the original space. This fact shows that sparse representation coefficients sought in the random subspaces are more discriminative than those in the original space. One possible reason is that these coefficients sought in subspaces are less affected by noisy and redundant features. This fact also justifies our motivation in using SSR for diverse and accurate base classifiers. From these results, we can argue that SSC-SSR can find a better tradeoff between diversity and accuracy than other comparing ensemble classifiers.

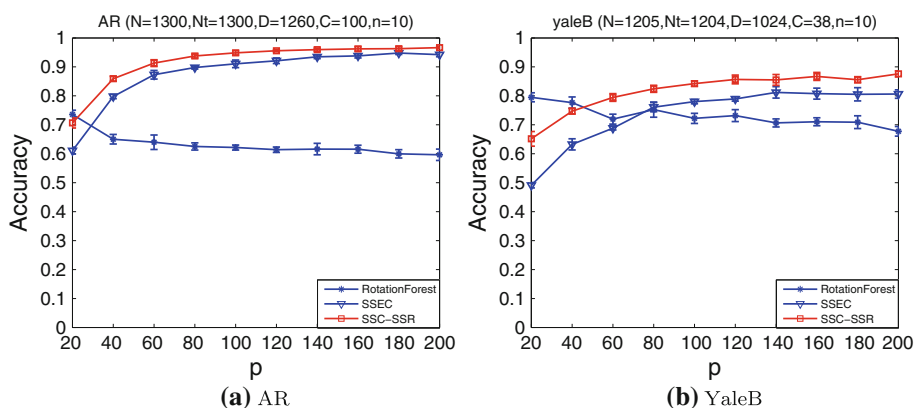
#### 4.4 Sensitivity on number of base classifiers

The number of base classifiers is an important parameter of an ensemble classifier. The required time and space increase as the number of base classifiers rises. To explore the performance of SSC-SSR under different number of base classifiers, another two experiments were conducted on AR and yaleB with  $T$  rising from 1 to 40. Figure 4 plots the recorded results.

From Fig. 4, we can observe that SSC-SSR can achieve good and stable performance when  $T \geq 10$ . Whereas the performance of SSEC downgrades as  $T$  increases. The reason is that the subspaces used in SSEC are randomly partitioned and the dimensionality of random subspaces is  $D/T$ . More base classifiers comes with more subspaces, but the dimensionality of these subspaces is too small to get accurate base classifiers, so the performance of SSEC downgrades as  $T$  increases. The sensitivity analysis show that SSC-SSR can select an effective  $T$  in a wider range than rotation forest, random forest and SSEC.



**Fig. 4** Accuracy versus number of base classifiers. **a** AR. **b** YaleB



**Fig. 5** Accuracy versus dimensionality of subspace. **a** AR. **b** YaleB

#### 4.5 Sensitivity on dimensionality of subspace

The dimensionality of random subspace is an important parameter for random subspace methods. To study the effect of  $p$ , two experiments are executed on AR and yaleB with  $p$  varying from 20 to 200. The corresponding experimental results are shown in Fig. 5. Random forest does not need to specify  $p$  in advance, its results are not reported.

It can be found that all the comparing methods rely on the choice of  $p$ . Note, for SSEC,  $p = D/T$ . At first,  $p$  is too small to learn good coefficient matrices or construct good  $k$ NN graphs, so SSC-SSR and SSEC lose to rotation forest. As  $p$  increases, there is sufficient number of features to get good coefficient matrices, and SSC-SSR gradually outperforms rotation forest. The performance of rotation forest on AR and yaleB downgrades as the dimensionality of subspaces increases. The possible reason is that AR and yaleB have many noisy features and the rotation matrices of rotation forest are badly distorted when the dimensionality of subspaces is rather large. No matter what  $p$  is chosen, SSC-SSR always has higher accuracies than SSEC. This fact shows that SSC-SSR can select a suitable  $p$  easier than SSEC. In Fig. 5, we can observe that the accuracy of SSC-SSR keeps relative stable as  $p \geq 100$ , for there

**Table 4** Time cost of SR in  $T = 20$  subspaces versus in *one* original space (s)

| Dataset( $N \times D$ )        | $p = 50$ | $p = 100$ | $p = 200$ | $D$       |
|--------------------------------|----------|-----------|-----------|-----------|
| AR ( $1,300 \times 1,260$ )    | 293.9    | 496.2     | 1,089.1   | 1,802.9   |
| YaleB ( $1,205 \times 1,024$ ) | 328.5    | 491.7     | 935.8     | 1,041.3   |
| PIE ( $1,631 \times 4,096$ )   | 528.9    | 795.6     | 1,672.5   | 118,304.4 |
| ORL ( $240 \times 1,024$ )     | 11.9     | 12.7      | 24.2      | 210.5     |

is sufficient number of features to get the sparse representation coefficients when  $p \geq 100$ . This observation coincides with the observation of Wright et al. [30].

We also record the average run time (on Windows 7 with Intel Core2 E8400 and 4GB RAM) of solving SR in 20 subspaces and in the original space, and report them in Table 4. From this table, we can observe that computing sparse representation coefficients in  $T = 20$  subspaces takes much less time than those in one original space, especially for the datasets with  $D > N$  (i.e., PIE and ORL). Together with the complexity analysis in Sect. 3.3, we can claim that our motivation in reducing the computation load of sparse representation and boosting the performance of semi-supervised classification on high-dimensional is effective.

## 5 Conclusions and future work

In this paper, to address the drawback of graph-based semi-supervised classification in high-dimensional data, we put forward an ensemble method called Semi-Supervised Classification based on Subspace Sparse Representation (SSC-SSR in short). The base classifiers of SSC-SSR are trained on graphs defined by sparse representation coefficients in random subspaces, they are diverse and accurate, and contribute to a competent ensemble classifier. Our experimental results show that SSC-SSR not only has better performance but also wider ranges in choice of effective parameters than other comparing methods.

Although sparse representation coefficients can be solved much more efficiently in the subspace, there are still several issues for further investigation. Devising efficient  $l_1$  norm regularized solver is still promising and has the potential of wide applications. Our proposed approach relaxes the required data distribution for sparse representation, but it is still lacking of theoretic justification. In our future work, we will try to exploit the label information in solving the sparse representation coefficients.

**Acknowledgments** The authors are grateful to the discussion with Dr. Jieping Ye and appreciated to the valuable comments from anonymous reviewers and editors. This work is supported by Natural Science Foundation of China (Nos. 61003174, 61101234 and 61372138), Natural Science Foundation of Guangdong Province (No. S2012010009961), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20110172120027), Cooperation Project in Industry, Education and Academy of Guangdong Province and Ministry of Education of China (No. 2011B090400032), Fundamental Research Funds for the Central Universities (Nos. 2012ZZ0064, XDJK2010B002, XDJK2013C123), Doctoral Fund of Southwest University (Nos. SWU110063 and SWU113034), Open Project from Key Laboratory of Electronic Commerce Market Application Technology (No. 2011GDECOF01) and China Scholarship Council (CSC).

## References

1. Belkin M, Niyogi P, Sindhvani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Learn Res* 7:2399–2434

2. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
3. Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. *Inf Fusion* 6(1):5–20
4. Chen L, Liao H, Ko M, Lin J, Yu G (2000) A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recogn* 33(10):1713–1726
5. Chung F (1997) Spectral graph theory. Regional conference series in mathematics, No 92
6. Donoho D (2006) Compressed sensing. *IEEE Trans Inf Theory* 52(4):1289–1306
7. Duda R, Hart P, Stork D (2001) Pattern classification. Wiley, New York
8. Elhamifar E, Vidal R (2009) Sparse subspace clustering. *Proc IEEE Conf Comput Vis Pattern Recogn* 2009:2790–2797
9. Fan M, Gu N, Qiao H, Zhang B (2011) Sparse regularization for semi-supervised classification. *Pattern Recogn* 44(8):1777–1784
10. Georgiades A, Belhumeur P, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans Pattern Anal Mach Intell* 23(6):643–660
11. Ho T (1998) The random subspace method for constructing decision forests. *IEEE Trans Pattern Anal Mach Intell* 20(8):832–844
12. Huang K, Aviyente S (2006) Sparse representation for signal classification. In: *Proceedings of advances in neural information processing systems 2006*, pp 609–616
13. Jebara T, Wang J, Chang S (2009) Graph construction and b-matching for semi-supervised learning. In: *Proceedings of international conference on machine learning*, pp 441–448
14. Kleinberg E (2000) On the algorithmic implementation of stochastic discrimination. *IEEE Trans Pattern Anal Mach Intell* 22(5):473–490
15. Kuncheva L, Whitaker C (2003) Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach Learn* 51(2):181–207
16. Liu J, Ye J (2009) Efficient Euclidean projections in linear time. In: *Proceedings of international conference on machine learning*, pp 657–664
17. Liu W, Chang S (2009) Robust multi-class transductive learning with graphs. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 381–388
18. Maier M, Von Luxburg U, Hein M (2008) Influence of graph construction on graph-based clustering measures. In: *Proceedings of advances in neural information processing system*, pp 1025–1032
19. Margineantu D, Dietterich T (1997) Pruning adaptive boosting. In: *Proceedings of international conference on machine learning*, pp 211–218
20. Martinez A (1998) The AR face database. Technical Report 24, CVC
21. Olshausen B, Field D (1997) Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vis Res* 37(23):3311–3325
22. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. *ACM SIGKDD Explor Newsl* 6(1):90–105
23. Qiao L, Chen S, Tan X (2010) Sparsity preserving projections with applications to face recognition. *Pattern Recogn* 43(1):331–341
24. Qiao L, Chen S, Tan X (2010) Sparsity preserving discriminant analysis for single training image face recognition. *Pattern Recogn Lett* 31(5):422–429
25. Rodríguez J, Kuncheva L, Alonso C (2006) Rotation forest: a new classifier ensemble method. *IEEE Trans Pattern Anal Mach Intell* 28(10):1619–1630
26. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326
27. Samaria F, Harter A (1994) Parameterisation of a stochastic model for human face identification. In: *Proceedings of the second IEEE workshop on applications of computer vision*, pp 138–142
28. Sim T, Baker S, Bsat M (2003) The CMU pose, illumination, and expression (PIE) database. *IEEE Trans Pattern Anal Mach Intell* 25(12):1615–1618
29. Wang J, Wang F, Zhang C, Shen H, Quan L (2009) Linear neighborhood propagation and its applications. *IEEE Trans Pattern Anal Mach Intell* 31(9):1600–1615
30. Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2009) Robust face recognition via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 31(2):210–227
31. Wu F, Wang W, Yang Y, Zhuang Y, Nie F (2010) Classification by semi-supervised discriminative regularization. *Neurocomputing* 73(10):1641–1651
32. Yan S, Wang H (2009) Semi-supervised learning with sparse representation. In: *Proceedings of SIAM conference on data mining*, pp 792–801
33. Yan S, Xu D, Zhang B, Zhang HJ, Yang Q, Lin S (2007) Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell* 29(1):40–51

34. Yang A, Ganesh G, Sastry S, Ma Y (2010) Fast  $\ell_1$ -minimization algorithms and an application in robust face recognition: a review. Technical Report 2010–13, Department of Electrical Engineering and Computer Science, University of California at Berkeley
35. Yang J, Zhang L, Xu Y, Yang J (2012) Beyond sparsity: the role of L1-optimizer in pattern classification. *Pattern Recogn* 45(3):1104–1118
36. Yu G, Zhang G, Domeniconi C, Yu Z, You J (2012) Semi-supervised classification based on random subspace dimensionality reduction. *Pattern Recogn* 45(3):1119–1135
37. Yu G, Zhang G, Yu Z, Domeniconi C, You J, Han G (2012) Semi-supervised ensemble classification in subspaces. *Appl Soft Comput* 12(5):1511–1522
38. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: *Proceedings of advances in neural information processing systems*, pp 321–328
39. Zhu X (2008) Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin-Madison
40. Zhu X, Ghahramani Z, Lafferty J (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: *Proceedings of international conference on machine learning* 2003, pp 912–919

## Author Biographies



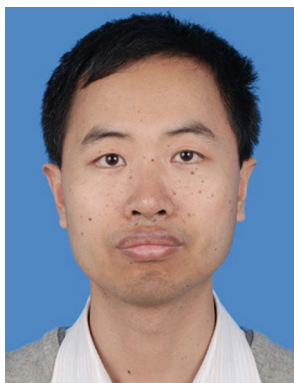
**Guoxian Yu** is an Associate Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. He received B.Sc. degree in Software Engineering from Xi'an University of Technology, Xi'an, China, in 2007, and Ph.D. in Computer Science from South China University of Technology, Guangzhou, China, in 2013. He visited the Data Mining Lab in the George Mason University, VA, USA, from 2011 to 2013. He is a recipient of Best Poster Award of SDM12 Doctoral Forum and Best Student Paper Award of 10th IEEE International Conference on Machine Learning and Cybernetics (ICMLC). His current research interests include machine learning, data mining and bioinformatics, particularly in semi-supervised learning, multi-label learning and ensemble learning.



**Guoji Zhang** is a Professor at the School of Sciences, South China University of Technology, Guangzhou, China. He received his B.Sc. degree in Computer Application and Ph.D. degree in Circuit and System from South China University of Technology, in 1977 and 1999, respectively. His research interests include computational intelligence, computational electromagnetic and cryptology, and he has published over 50 research papers.



**Zili Zhang** is a Professor at the College of Computer and Information Science, Southwest University, Chongqing, China, and a Senior Lecturer at Deakin University, Australia. He received his B.Sc. degree from Sichuan University, M.Eng. from Harbin Institute of Technology, and Ph.D. from Deakin University, all in Computing. He authored or co-authored more than 100 refereed papers in international journals or conference proceedings, 1 monograph, and 5 textbooks. His research interests include bio-inspired artificial intelligence, agent-based computing, big data analysis, and agent-data mining interaction and integration.



**Zhiwen Yu** is a Professor in the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He received the B.Sc. and M.Phil. degrees from the Sun Yat-Sen University in China in 2001 and 2004 respectively, and the Ph.D. degree in Computer Science from the City University of Hong Kong, in 2008. His research interests include bioinformatics, machine learning, pattern recognition, intelligent computing and data mining. He has published more than 70 technical articles in referred journals and conference proceedings in the areas of bioinformatics, artificial intelligence, pattern recognition and multimedia.



**Lin Deng** is a Ph.D. student in the Department of Computer Science, George Mason University, VA, USA. He received B.Sc. degree in Computer Science from Renmin University of China, Beijing, China, in 2005, and M.S. degree in Computer and Information Science from Gannon University, PA, USA, in 2011. His current research interests include software engineering, information security and data mining.