Safety-Aware Semi-Supervised Classification

Yunyun Wang and Songcan Chen

***Abstract*— Though semi-supervised classification learning has attracted great attention over past decades, semi-supervised classification methods may show worse performance than their supervised counterparts in some cases, consequently reducing their confidence in real applications. Naturally, it is desired to develop a safe semi-supervised classification method that never performs worse than the supervised counterparts. However, to the best of our knowledge, few researches have been devoted to safe semi-supervised classification. To address this problem, in this paper, we invent a safety-control mechanism for safe semi-supervised classification by adaptive tradeoff between semi-supervised and supervised classification in terms of unlabeled data. In implementation, based on our recent semi-supervised classification method based on class memberships (SSCCM), we develop a safety-aware SSCCM (SA-SSCCM). SA-SSCCM, on the one hand, exploits the unlabeled data to help learning (as SSCCM does) under the assumption that unlabeled data can help learning, and on the other hand, restricts its prediction to approach that of its supervised counterpart least-square support vector machine (LS-SVM) under the assumption that unlabeled data can hurt learning. Therefore, prediction by SA-SSCCM becomes a tradeoff between those by semi-supervised SSCCM and supervised LS-SVM, respectively, in terms of the unlabeled data. As in SSCCM, the optimization problem in SA-SSCCM can be efficiently solved by the alternating iterative strategy, and the iteration convergence can theoretically be guaranteed. Experiments over several real datasets show the promising performance of SA-SSCCM compared with LS-SVM, SSCCM, and off-the-shelf safe semi-supervised classification methods.**

***Index Terms*— Alternating iterative strategy, least-square support vector machine (LS-SVM), semi-supervised classification, semi-supervised classification based on class memberships (SSCCM).**

I. INTRODUCTION

IN MANY real applications, the unlabeled data can be easily and cheaply collected, whereas the acquisition of labeled data is usually quite expensive and time-consuming, especially involving manual effort. For instance, in web page recommendation, huge amounts of web pages are available, but few users are willing to spend much time in marking which web pages they are interested in. In spam email detection, a large number of emails can be automatically collected, yet few of them are labeled spam or not by users. Therefore, semi-supervised learning, which exploits the huge amounts of unlabeled data jointly with the limited labeled data for learning, has attracted intensive attention during the past decades. In this paper, we focus on semi-supervised classification, and lots of semi-supervised classification methods were developed during the past decades [1]–[4].

Semi-supervised classification methods attempt to exploit the intrinsic data distribution information disclosed by the unlabeled data, and the information is usually considered to be helpful for learning. To exploit the unlabeled data, some assumption should be adopted for learning. Two common assumptions in semi-supervised classification are the cluster assumption and the manifold assumption [3]–[5]. The cluster assumption assumes similar instances are likely to share the same class label, thus guides the classification boundary passing through the low density region. The manifold assumption assumes data are distributed on some low dimensional manifold represented by a Laplacian graph, and similar instances should share similar classification outputs according to the graph. Almost all off-the-shelf semi-supervised classification methods adopt one or both of those assumptions explicitly or implicitly [1], [4]. For instance, the large margin semi-supervised classification methods, such as transductive support vector machine (TSVM) [6], semi-supervised SVM (S3VM) [7], and their variants [8], [9], adopt the cluster assumption. The graph-based semi-supervised classification methods, such as label propagation [10], [11], graph cuts [12], and Laplacian SVM (LapSVM) [13], adopt the manifold assumption. Those semi-supervised classification methods are applied in diverse applications, e.g., text classification [6], spam email detection [14], and bioinformatics [15], and so on.

It is, however, found that in some cases, semi-supervised classification methods may yield even worse performances than their supervised counterparts because of using the unlabeled data [16]–[18]. The unlabeled data can hurt the performance or be harmful for learning because of the failure of model assumption or data distribution assumption [4], [16], [19]. Considering that, Wang *et al.* [19] suggested a modified cluster assumption that similar instance should share similar class memberships rather than a crisp class label, and accordingly developed a new semi-supervised classification method based on class memberships (SSCCM). The modified cluster assumption can better capture real-data distribution than the cluster assumption, and therefore, SSCCM largely achieves better performance than semi-supervised classification methods based on the cluster assumption. Nevertheless, like other semi-supervised classification methods, SSCCM still performs worse than its supervised counterpart least square (LS)-SVM in some cases. It lowers the confidence of applying

Manuscript received August 8, 2012; accepted May 5, 2013. Date of publication June 17, 2013; date of current version October 15, 2013. This work was supported in part by the National Science Foundation of China under Grant 61035003, Grant 61170151, Grant 61272422, and Grant 61073114, and the Jiangsu Qinglan project.

Y. Wang is with the Department of Computer Science and Engineering, Nanjing University of Posts & Telecommunications, Nanjing 210046, China (e-mail: wangyunyun@njupt.edu.cn).

S. Chen is with the Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China (e-mail: s.chen@nuaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2263512

the semi-supervised methods to real applications. Therefore, it is desired to develop a safe semi-supervised classification method that never performs worse than its supervised counterparts, or degenerate performance because of using unlabeled data [17]. To the best of our knowledge, there were, however, only two recent researches [17], [18] on safe semi-supervised classification up to now. Li *et al.* [18] developed the S3VM-*us* through selecting unlabeled instances by hierarchical clustering such that only the unlabeled instances very likely to be helpful are exploited. More specifically, in S3VM-*us*, only the unlabeled instances with high confidence by hierarchical clustering are predicted by TSVM, and the rest are predicted by SVM. Finally, the chance of its performance degeneration is much smaller than that of TSVM. Simultaneously, Li *et al.* [17] also developed the safe S3VM (S4VM) method. Different from S3VM seeking an optimal low-density separator, S4VM exploits the candidate low-density separators simultaneously to reduce the risk of identifying a poor separator with unlabeled data. Specifically, S4VM finds diverse large-margin low-density separators, and then optimizes the label assignment for the unlabeled data in the worse case under the assumption that the ground-truth label assignment can be realized by one of the obtained low-density separators. Finally, the performance of S4VM is highly competitive to TSVM and never significantly inferior to (supervised) SVM. Both S3VM_*us* and S4VMs are actually improvements based on inductive S3VM, while belong to the transductive-style methods, which learn on both the given labeled and unlabeled data, and aim to seek only the labels for the given unlabeled data.

It is expected that a safe semi-supervised classification method can perform no worse than the supervised counterpart when the available unlabeled data are unreliable for learning, and no worse than the semi-supervised method when the available unlabeled data are reliable for learning. Simply, we expect the final prediction to approach to that of the supervised counterparts when given unreliable unlabeled data, and approach to that of semi-supervised method when given reliable unlabeled data. To this end, we invent a safety-control mechanism for safe semi-supervised classification in this paper by adaptively controlling the tradeoff between semi-supervised and supervised classification in terms of the available unlabeled data. In implementation, based on our recent SSCCM, we develop a safety-aware SSCCM (SA-SSCCM). SA-SSCCM, on the one hand, exploits the unlabeled data to help learning (as SSCCM does) under the assumption that the unlabeled data can help learning, on the other hand, restricts its prediction to approach to that of its supervised counterpart LS-SVM under assumption that the unlabeled data can hurt learning. Finally, the prediction by SA-SSCCM is a tradeoff between those by SSCCM and LS-SVM, respectively, in terms of the unlabeled data. Further, we provide a strategy for adjusting such tradeoff in terms of the given unlabeled data. As in SSCCM, the optimization problem in SA-SSCCM can be efficiently solved by the alternating iterative strategy, and the convergence of the iteration can be theoretically guaranteed. Empirical results on several real datasets show that the overall performance of SA-SSCCM is better than those of both SSCCM and LS-SVM, and in addition, the performance of SA-SSCCM is never significantly inferior to that of LS-SVM, and seldom significantly inferior to that of SSCCM.

Both S3VM_*us* and S4VMs are developed to perform no worse than SVM, whereas SA-SSCCM is developed to perform no worse than both the supervised and semi-supervised counterparts, which is reflected in the empirical results. In addition, different from transductive S3VM_*us* and S4VMs, SA-SSCCM learns in an inductive style, which learns on both the given labeled and unlabeled data, and aims to seek the discriminant function for predicting unseen instances. It is also worth pointing out that though we concern SSCCM in this paper, the proposed mechanism can be easily applied to other semi-supervised methods such as TSVM, S3VM, and LapSVM.

The rest of this paper is organized as follows. Section II briefly introduces SSCCM, Section III describes the proposed SA-SSCCM, Section IV describes the empirical results, and finally, conclusions are summarized in Section V.

II. Semi-Supervised Classification Method Based on Class Memberships

The cluster assumption assumes similar instances should share the same class label, thus implicitly assumes each instance should have a crisp label assignment. There are, however, instances difficult to be assigned to a single class in real applications, such as the boundary instances. In those cases, the cluster assumption cannot reflect the real-data distribution adequately, and would lead to poor predictions when adopted in semi-supervised classification. Therefore, Wang *et al.* [19] suggested a modified cluster assumption assuming similar instances should share similar class memberships rather than a crisp class label. For each instance, the class memberships are represented as a vector, the value of each element expresses the likelihood of the concerned instance belonging to the class. Through adopting the modified cluster assumption as well as the local learning principle (constraining each instance and its local weighted mean (LWM) share the same label membership vector), Wang *et al.* [19] developed a new semi-supervised classification method named SSCCM.

Now, given labeled data $X_l = \{x_i\}_{i=1}^{nl}$ with the corresponding labels $Y = \{y_i\}_{i=1}^{nl}$, and unlabeled data $X_u = \{x_j\}_{j=nl+1}^{n}$ where each $x_i \in R^d$, $y_i \in R^C$ for the C-class classification, and $n_u = n - n_l$. With a decision function $f(x)$ and a label membership function $\upsilon(x)$, SSCCM can be formulated as follows:

$$\begin{aligned}\min_{f,\upsilon_k(x_j)} \quad & \sum_{i=1}^{nl} ||f(x_i) - y_i||^2 + \lambda_s \sum_{i=1}^{nl} ||f(\hat{x}_i) - y_i||^2 \\ & + \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2 \\ & + \lambda_s \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(\hat{x}_j) - r_k||^2 + \lambda ||f||_{\mathcal{H}}^2 \\ & \text{s.t.} \sum_{k=1}^{C} \upsilon_k(x_j) = 1 \\ & 0 \le \upsilon_k(x_j) \le 1, k = 1 \ldots C, \quad j = n_l + 1 \ldots n \quad (1)\end{aligned}$$

where $||\cdot||_{\mathcal{H}}$ is a norm in the reproducing Kernel Hilbert space, $\{r_k\}_{k=1}^{C}$ are the encodings for the C classes, both $y_i \in R^C$ and $r_k \in R^C$ are encoded by the one-of-C rule, i.e., the kth element of y_i is one and the rest are zero if x_i belongs to the kth class, the kth element of r_k is one and the rest are zero, $f(x_i) \in R^C$ and $\upsilon(x_i) \in R^C$ for each x_i, and $\upsilon_k(x_i)$ expresses the likelihood of x_i belonging to the kth class, $\hat{x}_i$ is the LWM of x_i defined by

$$\hat{x}_i = \frac{\sum_{x_j \in \mathrm{Ne}(x_i)} S_{ij} x_j}{\sum_{x_j \in \mathrm{Ne}(x_i)} S_{ij}} \tag{2}$$

where $\mathrm{Ne}(x_i)$ is the k nearest-neighborhood of x_i measured by the Euclidean distance, and S_{ij} is a quantity inversely proportional to the distance between x_i and x_j, $\forall k = 1 \cdots C, i = 1 \cdots n$.

In SSCCM, each instance can belong to multiple classes with the corresponding class memberships, and in addition, each instance and its LWM share the same label membership vector. The optimization problem can be efficiently solved by the alternating iterative strategy, in which each step generates a closed-form solution. The convergence of the iterative solving process can be theoretically guaranteed. Finally, SSCCM achieves competitive performances compared with several state-of-the-art semi-supervised classification methods such as TSVM, LapSVM, and mean3svm.

Like other semi-supervised classification methods, SSCCM can yield a worse performance than its supervised counterpart LS-SVM. Thus, in this paper, we attempt to further develop a safety-aware semi-supervised classification method. In the computation of each LWM in SSCCM, there is, however, a latent risk [20] that instances from the opposite class may also be selected in the k nearest-neighborhood, and thus the obtained LWM may fall in the opposite class. To elude such a risk at the beginning of modeling SA-SSCCM, we remove the terms involving LWMs from (1) and simplify the optimization problem of SSCCM (we abuse the name SSCCM hereafter for convenience) as follows:

$$\begin{aligned} \min_{f,\upsilon_k(x_j)} \quad & ||f||_{\mathcal{H}}^2 + \lambda_1 \sum_{i=1}^{nl} ||f(x_i) - y_i||^2 \\ & + \lambda_2 \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2 \\ \text{s.t.} \quad & \sum_{k=1}^{C} \upsilon_k(x_j) = 1 \\ & 0 \le \upsilon_k(x_j) \le 1, k = 1 \cdots C, \quad j = n_l + 1 \cdots n. \end{aligned} \tag{3}$$

III. Safety-Aware SSCCM

Here, we describe the proposed SA-SSCCM method, including the formulation, optimization, algorithmic description, and value of parameter λ in separated sections.

A. Formulation

With (simplified) SSCCM, we develop a safety-aware semi-supervised classification method SA-SSCCM, whose final prediction approaches to that of semi-supervised SSCCM when the unlabeled data are helpful for learning, and approaches to that of supervised LS-SVM when the unlabeled data are harmful for learning. To use the prediction of LS-SVM in learning, we denote the decision function obtained from LS-SVM as $g(x)$, then with a decision function $f(x)$ and label membership function $\upsilon(x)$, we establish SA-SSCCM as follows:

$$\begin{aligned} \min_{f,\upsilon_k(x_j)} \quad & ||f||_{\mathcal{H}}^2 + \lambda_1 \sum_{i=1}^{n} ||f(x_i) - y_i||^2 \\ & + \lambda_2 \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2 \\ & + \left(\frac{1}{\lambda} - 1\right) \sum_{j=n_l+1}^{n} ||f(x_j) - g(x_j)||^2 \\ \text{s.t.} \quad & \sum_{k=1}^{C} \upsilon_k(x_j) = 1 \\ & 0 \le \upsilon_k(x_j) \le 1, k = 1 \ldots C, j = n_l + 1 \ldots n. \end{aligned} \tag{4}$$

The first three terms in the objective function of (4) seek the decision function $f(x)$ and label membership function $\upsilon(x)$ simultaneously through exploiting both labeled and unlabeled data as SSCCM does, while the last term controls the difference between the predictions (for the unlabeled instances) by SA-SSCCM and LS-SVM, respectively. Therefore, the prediction by SA-SSCCM becomes a tradedoff between those of SSCCM and LS-SVM, respectively, adjusted by λ. λ takes its value from [0, 1] according to the available unlabeled data, when λ approaches zero (let $1/\lambda$ be infinity when λ approaches zero), the prediction of SA-SSCCM degenerates to that of LS-SVM, and when λ approaches one, the prediction of SA-SSCCM degenerates to that of SSCCM. Therefore, λ plays an important role in SA-SSCCM controlling the tradeoff between SSCCM and LS-SVM, and its value will be discussed later in Section III-D.

B. Optimization

As in SSCCM, the optimization problem in SA-SSCCM is biconvex in (f, υ), for which we can resort to the alternating iterative solving strategy with convergence guarantee [21]. Each step in the optimization iteration yields a closed-form solution for both $f(x)$ and $\upsilon(x)$.

For fixed $\upsilon(x)$, the optimization problem of SA-SSCCM can be recast as follows:

$$\begin{aligned} \min_{f,\upsilon_k(x_j)} \quad & ||f||_{\mathcal{H}}^2 + \lambda_1 \sum_{i=1}^{n_l} ||f(x_i) - y_i||^2 \\ & + \lambda_2 \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2 \\ & + \left(\frac{1}{\lambda} - 1\right) \sum_{j=n_l+1}^{n} ||f(x_j) - g(x_j)||^2. \end{aligned} \tag{5}$$

The minimizer of (5) has the form $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$ based on the representer theorem [13], where each $\alpha_i \in R^{C \times 1}$.

Then, (5) can be reformulated as follows:

$$\min_{\alpha} M_1 = \mathrm{tr}(\alpha K \alpha^T) + \lambda_1 tr\left((\alpha K_l - Y)(\alpha K_l - Y)^T\right) + \lambda_2 \sum_{k=1}^{C} tr((\alpha K_u - L_k)\hat{V}_k(\alpha K_u - L_k)^T) + \left(\frac{1}{\lambda} - 1\right) tr((\alpha K_u - \alpha_0 K_u)(\alpha K_u - \alpha_0 K_u)^T) \quad (6)$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in R^{C\times n}$ is the Lagrange multiplier matrix, and α_0 is the Lagrange multiplier matrix for LS-SVM. $K = [K_l \; K_u] = \begin{bmatrix} K_u & K_{lu} \\ K_{ul} & K_{uu} \end{bmatrix}$ is a kernel matrix, where $K_u = \langle \phi(X_l), \phi(X_l)\rangle_{\mathcal{H}}$, $K_{lu} = \langle \phi(X_l), \phi(X_u)\rangle_{\mathcal{H}}$ and $K_{uu} = \langle \phi(X_u), \phi(X_u)\rangle_{\mathcal{H}}$. $\mathbf{L}_k$ is a $C \times n_u$ matrix with the kth row being an all-one vector and the rests being all-zero vectors. Let V_k be a label membership vector corresponding to the kth class, and $\hat{V}_k$ is defined as a diagonal matrix with the diagonal elements being the squared values of the entries in V_k.

Now, by zeroing the derivative of M_1 with respect to α, we have the following:

$$\frac{\partial M_1}{\partial \alpha} = \alpha K + \lambda_1(\alpha K_l - Y)K_l^T + \lambda_2 \sum_{k=1}^{C}(\alpha K_u - L_k)\hat{V}_k K_u^T + \left(\frac{1}{\lambda} - 1\right)((\alpha K_u - \alpha_0 K_u)K_u^T) = 0 \quad (7)$$

leading to the following solution:

$$\alpha = \left(\lambda_1 K_l^T Y + \lambda_2 \sum_{k=1}^{C} K_u \hat{V}_k L_k + \left(\frac{1}{\lambda} - 1\right)\alpha_0 K_u K_u^T\right) \times \left(\lambda_1 K_l K_l^T + \lambda_2 \sum_{k=1}^{C} K_u \hat{V}_k K_u^T + K + \left(\frac{1}{\lambda} - 1\right) K_u K_u^T\right)^{-1}. \quad (8)$$

For fixed $f(x)$, the optimization problem of SA-SSCCM becomes

$$\min_{\upsilon_k(x_j)} \sum_{k=1}^{C}\sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2$$
$$\text{s.t.} \sum_{k=1}^{C} \upsilon_k(x_j) = 1$$
$$0 \le \upsilon_k(x_j) \le 1, k = 1 \ldots C, j = n_l + 1 \ldots n. \quad (9)$$

Using the Lagrange multiplier method, we define the following:

$$M_2 = \sum_{k=1}^{C}\sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 ||f(x_j) - r_k||^2 - \lambda_j \left(\sum_{k=1}^{C} \upsilon_k(x_j) - 1\right). \quad (10)$$

Likewise, zeroing the derivative of M_2 with respect to each $\upsilon_k(x_j)$, $\forall k = 1 \ldots C, j = 1 \ldots n_u$, we obtain the following:

$$\frac{\partial M_2}{\partial \upsilon_k(x_j)} = 2||f(x_j) - r_k||^2 - \lambda_j = 0 \quad (11)$$

thus

$$\upsilon_k(x_j) = \lambda_j/2||f(x_j) - r_k||^2. \quad (12)$$

Further, integrating the constraint $\sum_{k=1}^{C} \upsilon_k(x_j) = 1$, we have the following:

$$\upsilon_k(x_j) = \frac{1/||f(x_j) - r_k||^2}{\sum_{k=1}^{C} 1/||f(x_j) - r_k||^2} \quad (13)$$

where $k \in \{1 \ldots C\}, i \in \{1 \ldots n\}$.

Therefore, for an arbitrary instance x

$$\upsilon_k(x) = \frac{1/||f(x) - r_k||^2}{\sum_{k=1}^{C} 1/||f(x) - r_k||^2} \quad (14)$$

where $k \in \{1 \ldots C\}$.

As in SSCCM, prediction for each given instance in SA-SSCCM can be implemented by not only the decision function $f(x)$, but also the class membership function $\upsilon(x)$, which reflects the likelihoods of the instance to individual classes. As shown in the following Proposition 1, different from the likely invoked inconsistent predictions by such two functions in SSCCM, the two predictions in SA-SSCCM are always consistent.

Proposition 1: Predictions for each given instance by the decision function and class membership function are always consistent.

Proof: For arbitrary instance x_i, its class label predicted by the decision function is $\hat{y}_i = \max_{k=1\ldots C} f_k(x_i)$, thus $x_i \in X_k$ implies that $f_k(x_i) > f_j(x_i), \forall j = 1 \ldots C, j \ne k$, where $\mathbf{X}_k$ is the set of instances belonging to the kth cluster. While its class label predicted by the class membership function is $\tilde{y}_i = \max_{k=1\ldots C} \upsilon_k(x_i)$, thus from (14), $x_i \in X_k$ implies that $||f(x_i) - r_k||^2 < ||f(x_i) - r_j||^2$, then $f(x_i)^T r_k > f(x_i)^T r_j$, or equivalently, $f_k(x_i) > f_j(x_i), \forall j = 1 \ldots C, j \ne k$. Therefore, the prediction conditions for $x_i \in X_k$ by both $f(x)$ and $\upsilon(x)$ are equivalent, thus the two predictions are consistent. ■

C. Algorithmic Description

The optimization of SA-SSCCM follows an alternating iterative strategy. As in SSCCM, the initial values for the label memberships of unlabeled instances in SA-SSCCM can be obtained by several strategies, such as randomization, some fuzzy clustering technique such as FCM, or simply setting them to all zeros. The iteration terminates when $|M_k - M_{k-1}| < \varepsilon M_{k-1}$, where M_k is the objective function value at the kth iteration and ε is a predefined threshold. The concrete algorithm description of SA-SSCCM is shown in Table I.

As in SSCCM, the alternating iterative process in the optimization of SA-SSCCM can be theoretically guaranteed to be convergent, and the detailed proof is similar to that in [19].

D. Value for λ

The λ in SA-SSCCM controls the prediction between those by SSCCM and LS-SVM, respectively, and in what follows, we attempt to select its value from [0, 1], which is actually a data-dependent problem. When the labeled instances are sufficient, we can adopt typical parameter-selection strategies such as cross validation. When the labeled instances are insufficient, the cross validation is invalid [3], [17]–[19], and

TABLE I

ALGORITHM DESCRIPTION OF SA-SSCCM

Input X_l: labeled data;
Y_l: labels of X_l;
X_u: unlabeled data;
λ, λ_1, λ_2: regularization parameters;
ε: iterative stop parameter;
σ: kernel parameter; Maxiter: maximum number for iteration.

Output $f(x)$: decision function;
$\upsilon(x)$: label membership function.

Procedure

Initialize the label memberships for the unlabeled data;
Set the initial objective function value to infinity, i.e., M_0= INF;
For k= 1…Maxiter
Update α by (8), and $f(x)$ by the represent theorem with obtained α;
Update $\upsilon(x)$ by (14);
Update the objective function value M_k;
If $|M_k - J_{k-1}| < \varepsilon M_{k-1}$
Break, return $f(x)$ and $\upsilon(x)$;
Endif
Endfor

thus we adopt the ensemble strategy as in [22]. Specifically, the problem of choosing λ can be simplified to choose its best value from $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$, where $\lambda_1 = 0$ and $\lambda_m = 1$, or equivalently, choosing the best decision function f from $\{f_1, f_2, \ldots, f_m\}$, where f_1 and f_m are the decision functions obtained, respectively, from LS-SVM and SSCCM. From the ensemble strategy for semi-supervised learning [22], [23], the optimal f can be represented as a linear combination of the base functions $\{f_i\}_{i=1}^m$, i.e., $f = \sum_{i=1}^m \omega_i f_i, \sum_{i=1}^m \omega_i = 1$, $\omega_i \geq 0, i = 1 \ldots m$. Finally, given $\{\lambda_i\}_{i=1}^m$, we can obtain the corresponding $\{f_i\}_{i=1}^m$, then the problem is converted to seek a set of combination weights $\{\omega_1, \omega_2, \ldots, \omega_m\}$, in terms of the following:

$$\min_{\omega_k} \frac{1}{2}\eta(FR\omega)^T L(FR\omega) - \vec{1}_C^T(Y_l F_l R\omega)$$
$$\text{s.t.} \quad \omega^T e = 1, \omega_k \geq 0 \tag{15}$$

where $\omega = [\omega_1, \omega_2, \cdots \omega_m]^T$, $e \in R^{m\times 1}$ is a vector with all elements equaling one. $F = [f^1(\mathbf{X})^T, f^2(\mathbf{X})^T \ldots f^m(\mathbf{X})^T] \in R^{n\times(C\times m)}$ and $F_l = [f^1(\mathbf{X}_l)^T, f^2(\mathbf{X}_l)^T \ldots f^m(\mathbf{X}_l)^T] \in R^{l\times(C\times m)}$ are the prediction matrix for the whole data set and the labeled data set, respectively. $\mathbf{Y}_l \in R^{C\times l}$ is the label matrix for the labeled data. $R = [\mathbf{R}^1\mathbf{R}^2 \ldots \mathbf{R}^m]^T \in R^{(C\times m)\times m}$, each $\mathbf{R}^k \in R^{m\times C}$ where the kth row being an all-one vector and the rests being all-zero vectors. $\vec{1}_C \in R^{C\times 1}$ is a vector with all elements equaling one. L is a graph Laplacian formulated as $L = D - W$, where $W = [w_{\text{ij}}]_{n\times n}$ is the weight matrix over the graph, and D is a diagonal matrix with the ith diagonal entry being $D_{ii} = \sum_{j=1}^n W_{\text{ij}}$. η is a regularization parameter.

The first term in the objective function of (15) guarantees the smoothness over the graph Laplacian [13], and the second term guarantees the correct predictions for labeled data. The optimization of (15) is a quadratic programming (QP) problem, which can be efficiently solved by any QP solvers.

Through adopting the ensemble strategy, the problem of setting λ is converted to the optimization of linearly combined coefficients $\{w_1, w_2, \ldots, w_m\}$ for a collection of discriminant functions $\{f_i\}_{i=1}^m$ with respect to $\{\lambda_i\}_{i=1}^m$, which is somewhat analog to multiple kernel learning [24] where the selection for the optimal parameters of a single kernel is translated to the selection of the linearly combined coefficients of multiple kernels.

IV. EXPERIMENT

Here, we first present simple illustrations to provide some intuition of how our new SA-SSCCM addresses the safe semi-supervised classification problem. Then, we evaluate the performance of SA-SSCCM on several real datasets by comparing with supervised LS-SVM, semi-supervised SSCCM, and off-the-shelf safe semi-supervised classification methods S3VM_*us* [18] and S4VMs [17].

A. Illustrative Experiments

SSCCM can be expected to perform better than supervised LS-SVM through exploiting the unlabeled data. However, the unlabeled data may hurt the performance because of the misspecification of the model or data distribution [16]–[18], thus similar to other semi-supervised classification methods, SSCCM also yields a worse performance than LS-SVM in some cases. In what follows, we will give intuitive illustrations of how SA-SSCCM works in cases that the unlabeled data hurt and benefit the classification, respectively.

We consider the bupa and ionosphere datasets as representations here. For each data set, we randomly select half instances for training, and the rests for testing, and for the training data set, we randomly select ten labeled instances, and leave the rests unlabeled. The linear kernel is adopted here, the regularization parameters λ_1, λ_2, and η are fixed to 100, 1, and 1, respectively, and ε is set to 10^{-3}. Each edge-weight in the Laplacian graph is given by $w_{\text{ij}} = \exp^{-||x_i - x_j||^2/2\sigma^2}$. The initial values for the label memberships of unlabeled instances in SA-SSCCM are simply set to all zeros following SSCCM, then SA-SSCCM is actually started from the labeled instances alone.

We first consider the bupa data set in the case that the unlabeled data hurt the classification. The regularization parameter λ in SA-SSCCM controls the tradeoff between semi-supervised and supervised learning. Fig. 1(a) shows the performances of SA-SSCCM with respect to different values of λ from $[0, 0.05, 0.1, \ldots, 1]$ with a space of 0.05. In Fig. 1(a), SSCCM performs worse than LS-SVM, indicating that the unlabeled data hurt the performance in this case. The performance of SA-SSCCM, degenerates to those of LS-SVM and SSCCM when λ approaches to zero and one, respectively, and decreases with the increase of λ. When λ is well-tuned, the performance of S3CCM can reach the better one between SSCCM and LS-SVM, i.e., the performance of LS-SVM. Therefore, SA-SSCCM with well-tuned λ can perform comparable with LS-SVM in case that the unlabeled data hurt the classification.

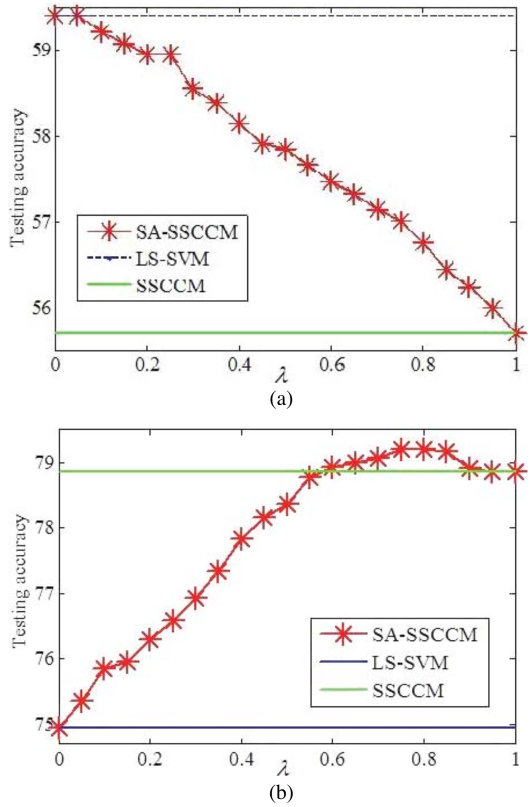

Fig. 1. Accuracies of LS-SVM, SSCCM, and SA-SSCCM with λ varying in range [0, 0.05, 0.1, …, 1] over (a) bupa and (b) ionosphere.

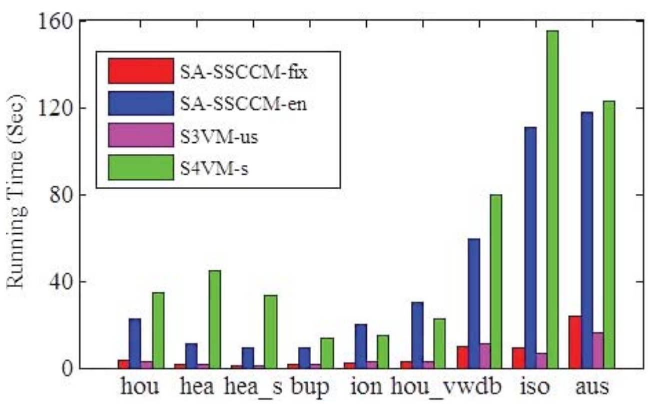

Fig. 2. Running time (s) of SA-SSCCM_*fix*, SA-SSCCM_*en*, S3VM_*us*, and S4VM_*s*.

Further, we conduct illustration on the ionosphere data set in the case that the unlabeled data benefit the classification. Fig. 1(b) shows the performance of SA-SSCCM with respect to different values of λ from $[0, 0.05, 0.1, \ldots, 1]$ over ionosphere. In Fig. 1(b), SSCCM performs better than LS-SVM, indicating that the unlabeled data can benefit the classification in this case. The performance of SA-SSCCM increases with the increase of λ, and outperforms those of both SSCCM and LS-SVM when λ is well tuned. Therefore, SA-SSCCM with well-tuned λ can outperform both SSCCM and LS-SVM when the unlabeled data benefit the classification.

From such an illustration, we can observe that the prediction of SA-SSCCM indeed tends to be the balance between those of SSCCM and LS-SVM, which exactly verifies our analysis in Section III-A. In addition, SA-SSCCM with well-tuned λ can be expected to perform comparable with the better one between LS-SVM and SSCCM, or even better than both LS-SVM and SSCCM. When the labeled data are sufficient, we can select the value for λ by cross validation as in supervised scenario, and when the labeled data are scare, we turn to the ensemble strategy in [22] here. The selection of regularization parameters is, however, still an open problem for both unsupervised and semi-supervised classification, and the value for λ is still a worth-studying problem in the future.

TABLE II
ATTRIBUTES OF THE REAL DATASETS

ID	Data	#Ins	#Fea	ID	Data	#Inst	#Fea
1	House	232	16	11	Austra	690	15
2	Heart	270	9	12	Diabetes	768	8
3	Heart-statlog	270	13	13	German	1000	20
4	Bupa	345	6	14	Optdigits	1143	42
5	Ionosphere	351	34	15	BCI	400	117
6	Vehicle	435	16	16	G241c	1500	241
7	House-votes	435	16	17	G241d	1500	241
8	Arrhythmia	452	279	18	COIL2	1500	241
9	WDBC	569	14	19	Digit1	1500	241
10	Isolet	600	51	20	USPS	1500	241

B. Performance Evaluation

Here, we evaluate the performance of SA-SSCCM over a set of real datasets by comparison with supervised LS-SVM, semi-supervised SSCCM, and off-the-shelf safe semi-supervised classification methods S3VM_*us* [18] and S4VM [17]. For SSCCM, we adopt the formulation in (3) for more direct comparison, and for S4VM, we adopt the version with representative sampling for more efficient computation [17]. Inductive semi-supervised methods including SA-SSCCM can predict the given unlabeled data as well by the decision function. Thus, in our experiments, for comparing with transductive S3VM_*us* and S4VM, we implement all methods in transductive style, i.e., learning on both the labeled and unlabeled data, and predicting the performances over the given unlabeled data.

1) Experimental Setups: We carry out performance comparison over both UCI and benchmark datasets, and their descriptions are shown in Table II. Each UCI data set is randomly split into two halves, one half for training and the other for testing, and the training set contains only ten labeled instances and the rests are unlabeled. This process along with the classifier learning is repeated 30 times and the average testing accuracies are reported. Both linear and Gaussian kernels are adopted here. The regularization parameters C_1 and C_2 are fixed to 1 and 0.1, respectively, λ_1, λ_2, and η are fixed to 100, 1, and 1, respectively, ε is set to 10^{-3}, and the width parameter σ in Gaussian kernel is set to the average distance between all instance pairs.

For the benchmark datasets, we follow the experimental setups in [4] and [19]. Specifically, for each data set, there are

TABLE III

PERFORMANCE COMPARISON WITH TEN LABELED INSTANCES

Lin.	SVM	LS-SVM	TSVM	SSCCM	S3VM_us	S4VM_s	SA-SSCCM_fix	SA-SSCCM_en
1	92.0 ± 1.4	89.1 ± 1.2	93.6 ± 1.1^{W}	95.3 ± 3.9^{W}	$92.3 \pm 1.5_{L}$	$93.3 \pm 1.8_{L}$	$91.1 \pm 3.3^{W}_{L}$	$\mathbf{95.2 \pm 3.5}^{W}$
2	58.6 ± 11.2	58.4 ± 7.8	58.2 ± 10.3	57.3 ± 9.7^{L}	58.4 ± 10.5	58.8 ± 11.4	57.1 ± 8.7^{L}	$\mathbf{59.1 \pm 9.8}^{W}_{W}$
3	63.6 ± 8.0	65.7 ± 7.3	63.9 ± 7.5	$\mathbf{68.2 \pm 11.6}^{W}$	63.8 ± 8.2	63.9 ± 7.8	67.8 ± 10.3^{W}	68.0 ± 10.2^{W}
4	56.6 ± 7.7	$\mathbf{59.4 \pm 6.2}$	55.3 ± 7.2^{L}	55.7 ± 5.3^{L}	55.8 ± 7.6	55.8 ± 7.8	$57.9 \pm 5.6^{L}_{W}$	$59.2 \pm 5.9_{W}$
5	76.9 ± 5.6	74.9 ± 4.2	74.9 ± 2.5^{L}	$\mathbf{78.9 \pm 4.6}^{W}$	$76.9 \pm 5.6_{W}$	$76.6 \pm 4.5_{W}$	78.4 ± 4.1^{W}	$\mathbf{78.9 \pm 4.5}^{W}$
6	79.1 ± 3.2	77.3 ± 3.1	76.0 ± 5.9^{L}	$\mathbf{81.6 \pm 7.6}^{W}$	$78.8 \pm 5.4_{W}$	$79.5 \pm 5.2_{W}$	$80.1 \pm 5.8^{W}_{L}$	81.1 ± 7.2^{W}
7	66.5 ± 11.7	64.1 ± 14.2	$\mathbf{77.6 \pm 10.9}^{W}$	74.2 ± 13.7^{W}	$68.2 \pm 11.8^{W}_{L}$	$70.6 \pm 9.2^{W}_{L}$	74.5 ± 7.6^{W}	74.5 ± 7.7^{W}
8	62.5 ± 13.8	58.0 ± 16.4	64.6 ± 13.4^{W}	$\mathbf{66.5 \pm 15.2}^{W}$	$62.9 \pm 14.3_{L}$	$63.2 \pm 13.2_{L}$	$64.6 \pm 6.8^{W}_{L}$	$64.8 \pm 6.7^{W}_{L}$
9	84.3 ± 2.9	84.9 ± 2.4	83.5 ± 1.7	86.6 ± 2.7^{W}	84.3 ± 2.9	$84.8 \pm 2.6_{W}$	$84.5 \pm 3.6_{L}$	$\mathbf{86.7 \pm 2.9}^{W}$
10	92.6 ± 6.8	92.6 ± 5.5	87.9 ± 9.4^{L}	91.4 ± 2.7^{L}	$93.4 \pm 7.4_{W}$	$\mathbf{94.6 \pm 6.8}^{W}_{W}$	$92.6 \pm 6.6_{W}$	$94.0 \pm 8.7^{W}_{W}$
11	76.7 ± 7.1	76.3 ± 12.0	80.8 ± 9.4^{W}	$\mathbf{83.2 \pm 5.6}^{W}$	$79.4 \pm 7.8^{W}_{L}$	$79.8 \pm 7.6^{W}_{L}$	$81.3 \pm 5.1^{W}_{L}$	$82.2 \pm 5.3^{W}_{L}$
12	58.8 ± 2.8	59.4 ± 6.9	57.7 ± 1.0^{L}	58.2 ± 2.6^{L}	$58.8 \pm 2.8_{W}$	58.2 ± 3.4	$59.2 \pm 2.1_{W}$	$\mathbf{59.6 \pm 2.6}_{W}$
13	60.1 ± 6.0	61.5 ± 4.8	56.2 ± 9.2^{L}	$\mathbf{62.6 \pm 6.9}^{W}$	$61.6 \pm 6.2^{W}_{W}$	$60.2 \pm 5.6_{W}$	$60.8 \pm 5.4_{L}$	62.2 ± 6.8
14	95.9 ± 2.3	97.4 ± 1.3	99.7 ± 0.4^{W}	$\mathbf{99.6 \pm 0.2}^{W}$	$96.4 \pm 2.6_{L}$	98.8 ± 2.8^{W}	98.7 ± 0.9^{W}	98.9 ± 0.7^{W}
15	52.8 ± 1.9	52.6 ± 3.1	51.3 ± 3.4^{L}	53.6 ± 2.6^{W}	$52.6 \pm 2.3_{W}$	52.0 ± 2.3	$\mathbf{55.6 \pm 2.4}^{W}_{W}$	$54.8 \pm 3.0^{W}_{W}$
16	54.6 ± 4.4	61.3 ± 3.5	$\mathbf{78.4 \pm 5.8}^{W}$	63.2 ± 3.4^{W}	$54.7 \pm 2.9_{L}$	$54.6 \pm 4.2_{L}$	$62.1 \pm 3.5_{L}$	63.8 ± 3.6^{W}
17	56.4 ± 5.2	59.5 ± 4.1	53.4 ± 9.6^{L}	55.4 ± 7.4^{L}	$56.4 \pm 5.4_{W}$	$56.3 \pm 5.1_{W}$	$59.0 \pm 3.5_{W}$	$\mathbf{59.7 \pm 4.4}_{W}$
18	56.3 ± 5.1	58.5 ± 4.3	56.5 ± 5.9	$\mathbf{58.9 \pm 4.9}$	56.3 ± 5.9	56.4 ± 4.8	58.5 ± 4.4	58.8 ± 4.9
19	77.0 ± 5.2	65.4 ± 10.8	$\mathbf{79.7 \pm 3.3}^{W}$	65.6 ± 9.1	$77.4 \pm 5.4_{L}$	$76.9 \pm 5.1_{L}$	65.6 ± 4.6	66.5 ± 8.4
20	77.9 ± 3.0	79.9 ± 0.9	72.4 ± 3.7^{L}	80.1 ± 0.4	$78.0 \pm 2.7_{W}$	$78.2 \pm 2.9_{W}$	80.2 ± 0.9	$\mathbf{80.3 \pm 0.7}$
Avg. acc	69.9	69.8	71.1	71.8	70.3	70.6	71.5	**72.4**
Semisuperv. versus superv. W/T/L			7/4/9	12/3/5	3/17/**0**	4/16/**0**	9/9/2	12/8/**0**
Safe semisuperv. versus semisuperv. W/T/L					8/5/7	7/7/6	5/9/8	6/12/**2**
Gau.	SVM	LS-SVM	TSVM	SSCCM	S3VM_us	S4VM_s	SA-SSCCM_fix	SA-SSCCM_en
1	91.0 ± 1.1	89.6 ± 0.6	92.8 ± 1.0^{W}	$\mathbf{95.6 \pm 1.6}^{W}$	$91.5 \pm 1.0_{L}$	92.4 ± 1.1^{W}	$92.8 \pm 0.9^{W}_{L}$	95.3 ± 0.8^{W}
2	57.2 ± 8.6	57.4 ± 7.2	58.8 ± 7.9^{W}	58.1 ± 6.4	$56.7 \pm 8.5_{L}$	$\mathbf{59.8 \pm 7.8}^{W}$	57.6 ± 6.4	58.1 ± 6.4
3	61.9 ± 3.9	61.8 ± 5.1	61.4 ± 4.8	64.1 ± 4.9^{W}	62.1 ± 3.9	62.2 ± 4.1	$\mathbf{64.9 \pm 4.8}^{W}$	64.1 ± 4.9^{W}
4	$\mathbf{61.9 \pm 6.3}$	60.9 ± 2.4	59.1 ± 5.9^{L}	60.6 ± 6.4	59.4 ± 5.4^{L}	$61.6 \pm 4.6_{W}$	61.4 ± 5.7	60.9 ± 6.4
5	77.9 ± 4.9	77.8 ± 6.9	73.7 ± 3.3^{L}	$\mathbf{79.3 \pm 5.0}^{W}$	$77.8 \pm 4.9_{W}$	$77.4 \pm 4.5_{W}$	79.7 ± 4.7^{W}	$\mathbf{79.3 \pm 5.0}^{W}$
6	78.1 ± 5.4	77.8 ± 3.7	73.5 ± 7.0^{L}	77.2 ± 6.1^{L}	$77.5 \pm 6.1_{W}$	$78.8 \pm 5.6_{W}$	77.8 ± 6.1	$\mathbf{80.5 \pm 4.8}^{W}_{W}$
7	71.4 ± 12.4	69.6 ± 9.8	74.9 ± 13.5^{W}	$\mathbf{76.7 \pm 8.7}^{W}$	$70.5 \pm 11.6_{L}$	$72.6 \pm 10.8_{L}$	$74.3 \pm 5.0^{W}_{L}$	76.3 ± 4.7^{W}
8	73.4 ± 11.9	72.6 ± 10.2	$\mathbf{78.8 \pm 7.5}^{W}$	75.0 ± 9.3^{W}	$72.6 \pm 11.0_{L}$	$74.5 \pm 11.6_{L}$	$76.2 \pm 9.8^{W}_{W}$	75.2 ± 10.2^{W}
9	$\mathbf{85.4 \pm 2.3}$	82.7 ± 3.9	84.2 ± 1.5	82.4 ± 5.0	$85.3 \pm 2.3_{W}$	84.8 ± 1.8	82.0 ± 4.1	82.5 ± 4.9
10	89.1 ± 7.3	88.3 ± 8.4	87.8 ± 9.4	87.0 ± 7.1^{L}	$89.8 \pm 7.3_{W}$	88.6 ± 6.8	$\mathbf{89.4 \pm 6.9}^{W}_{W}$	87.1 ± 14.6
11	77.1 ± 5.9	78.0 ± 6.4	81.9 ± 7.3^{W}	83.0 ± 5.1^{W}	$78.7 \pm 7.6^{W}_{L}$	$77.9 \pm 6.2_{L}$	$79.2 \pm 5.2^{W}_{L}$	$\mathbf{83.2 \pm 4.8}^{W}$
12	59.1 ± 3.1	59.8 ± 5.2	51.7 ± 1.0^{L}	59.6 ± 2.6	$58.8 \pm 2.8_{W}$	$57.6 \pm 2.6_{W}$	59.2 ± 2.1	$\mathbf{59.7 \pm 2.6}$
13	60.6 ± 5.6	59.6 ± 8.8	53.3 ± 6.3^{L}	$\mathbf{62.8 \pm 6.9}^{W}$	$61.1 \pm 5.1_{W}$	$60.8 \pm 5.4_{W}$	$61.2 \pm 5.3^{W}_{L}$	62.5 ± 6.9^{W}
14	91.6 ± 3.8	92.6 ± 2.3	96.8 ± 1.6^{W}	97.2 ± 1.4^{W}	$92.3 \pm 3.1_{L}$	$94.8 \pm 3.1^{W}_{L}$	97.7 ± 1.1^{W}	$\mathbf{98.4 \pm 1.5}^{W}_{W}$
15	51.1 ± 2.9	50.9 ± 0.8	51.4 ± 2.7	53.0 ± 2.4^{W}	51.1 ± 2.7	51.2 ± 2.7	$\mathbf{55.1 \pm 3.6}^{W}_{W}$	53.0 ± 2.4^{W}
16	53.6 ± 4.4	52.4 ± 4.6	$\mathbf{59.0 \pm 4.8}^{W}$	49.9 ± 0.1^{L}	$53.6 \pm 4.6_{L}$	$53.8 \pm 4.4_{L}$	$55.8 \pm 2.2^{W}_{W}$	$57.5 \pm 2.8^{W}_{W}$
17	52.8 ± 5.2	52.2 ± 4.6	53.1 ± 6.7	49.9 ± 0.2^{L}	52.9 ± 5.0	53.0 ± 5.4	$53.2 \pm 2.0^{W}_{W}$	$\mathbf{58.3 \pm 3.0}^{W}_{W}$
18	56.8 ± 4.0	53.0 ± 3.0	57.3 ± 4.2	55.7 ± 2.9^{W}	56.8 ± 4.1	57.0 ± 3.8	$\mathbf{60.7 \pm 3.8}^{W}_{W}$	55.8 ± 2.8^{W}
19	57.5 ± 6.5	56.1 ± 12.1	$\mathbf{80.3 \pm 3.1}^{W}$	66.3 ± 10.7^{W}	$60.2 \pm 6.7^{W}_{L}$	$76.8 \pm 6.4^{W}_{L}$	$60.2 \pm 2.5^{W}_{L}$	66.6 ± 10.6^{W}
20	79.9 ± 2.2	80.0 ± 0.1	71.3 ± 2.6^{L}	81.1 ± 0.9^{W}	$79.5 \pm 1.9_{W}$	$79.3 \pm 2.3_{W}$	$\mathbf{81.9 \pm 1.5}^{W}$	81.4 ± 1.1
Avg. acc	69.1	68.4	70.1	70.7	69.4	70.7	71.0	**71.8**
Semisuperv. versus superv. (W/T/L)			7/7/6	11/5/4	2/17/1	4/18/**0**	15/7/**0**	14/6/**0**
Safe semisuperv. versus semisuperv. (W/T/L)					7/5/8	7/9/6	6/9/5	4/16/**0**

two settings, one including ten labeled instances and the other including 100 instances. Further, for each data set and each setting, there are 12 subsets of labeled data and finally, the average performances on the unlabeled data are reported. The regularization parameters C_1 and C_2 in the compared methods are set to 100 and 0.1, respectively, λ_1, λ_2, and η are fixed

TABLE IV
PERFORMANCE COMPARISON WITH 100 LABELED INSTANCES

Dataset	SVM	LS-SVM	TSVM	SSCCM	S3VM_*us*	S4VM_*s*	SA-SSCCM_*cv*	SA-SSCCM_*en*
1	96.7 ± 0.4	96.7 ± 0.4	96.6 ± 0.5	96.7 ± 0.4	95.6 ± 0.5^{L}	96.5 ± 0.4	96.7 ± 0.4	$\mathbf{96.9 \pm 0.4}$
2	85.2 ± 1.9	85.8 ± 2.0	85.1 ± 1.9	$\mathbf{86.2 \pm 2.0}$	85.1 ± 1.8	85.5 ± 1.6	86.1 ± 2.1	86.2 ± 2.1
3	82.3 ± 2.2	83.8 ± 1.5	81.9 ± 1.3	84.1 ± 1.5	82.9 ± 2.1	82.0 ± 1.8	84.5 ± 1.7	$\mathbf{84.7 \pm 1.6}$
4	59.5 ± 3.7	58.5 ± 2.8	57.7 ± 3.6^{L}	59.2 ± 3.0	$59.1 \pm 3.7_{W}$	$59.3 \pm 3.1_{W}$	$\mathbf{59.7 \pm 2.9}$	58.4 ± 2.8
5	89.6 ± 0.9	88.9 ± 0.9	86.7 ± 3.1^{L}	89.6 ± 0.6^{W}	$89.6 \pm 0.9_{W}$	$\mathbf{90.6 \pm 0.8}_{W}$	90.1 ± 0.5^{W}	89.5 ± 0.7^{W}
6	92.4 ± 1.9	90.9 ± 2.1	88.5 ± 1.6^{L}	91.8 ± 1.9^{W}	$92.4 \pm 1.9_{W}$	$\mathbf{92.8 \pm 2.1}_{W}$	$92.7 \pm 2.1^{W}_{W}$	91.8 ± 1.9^{W}
7	91.6 ± 1.8	91.4 ± 1.9	87.3 ± 2.6^{L}	93.6 ± 1.9^{W}	$91.7 \pm 2.7_{W}$	$91.4 \pm 1.5_{W}$	$\mathbf{93.9 \pm 1.9}^{W}$	93.3 ± 2.1^{W}
8	91.8 ± 1.8	91.9 ± 18	90.3 ± 2.4^{L}	94.0 ± 1.3^{W}	$91.8 \pm 2.2_{W}$	$92.2 \pm 2.1_{W}$	$\mathbf{94.1 \pm 1.3}^{W}$	93.9 ± 1.4^{W}
9	95.6 ± 0.9	95.7 ± 0.6	83.6 ± 0.9^{L}	96.0 ± 0.5^{W}	$95.6 \pm 0.9_{W}$	$94.9 \pm 0.8_{W}$	$\mathbf{96.8 \pm 0.5}^{W}_{W}$	96.0 ± 0.5^{W}
10	98.0 ± 1.2	97.5 ± 1.0	99.6 ± 0.3^{W}	99.9 ± 0.1^{W}	99.1 ± 1.3^{W}	$98.2 \pm 0.9_{L}$	$\mathbf{100 \pm 0}^{W}_{W}$	$\mathbf{100 \pm 0}^{W}_{W}$
11	96.4 ± 4.6	95.8 ± 4.3	97.6 ± 5.1^{W}	$\mathbf{98.0 \pm 5.1}^{W}$	97.5 ± 4.6^{W}	96.7 ± 3.9	$\mathbf{98.0 \pm 5.0}^{W}$	97.7 ± 5.1^{W}
12	73.4 ± 2.2	73.8 ± 1.3	70.6 ± 6.9^{L}	76.1 ± 1.6^{W}	$73.6 \pm 2.1_{W}$	$74.8 \pm 1.8^{W}_{W}$	$76.8 \pm 0.9^{W}_{W}$	$\mathbf{77.1 \pm 0.9}^{W}$
13	69.9 ± 2.2	63.5 ± 1.9	58.2 ± 18.7^{L}	72.2 ± 0.1^{W}	$69.5 \pm 1.7_{W}$	$70.1 \pm 1.5_{W}$	$\mathbf{72.2 \pm 0.1}^{W}$	$70.3 \pm 0.5^{W}_{L}$
14	99.1 ± 0.4	98.9 ± 0.5	$\mathbf{99.8 \pm 0.2}$	$\mathbf{99.8 {\pm} 0}^{W}$	99.4 ± 0.4	99.6 ± 0.2	$\mathbf{99.8 \pm 0}^{W}$	99.6 ± 0.3^{W}
15	71.8 ± 3.5	73.1 ± 2.9	71.5 ± 4.1	75.3 ± 2.4^{W}	71.6 ± 3.2	71.5 ± 3.4	$76.2 \pm 2.2^{W}_{W}$	$\mathbf{76.8 \pm 2.9}^{W}_{W}$
16	75.0 ± 1.9	76.3 ± 1.5	$\mathbf{80.0 \pm 1.7}^{W}$	73.9 ± 2.2^{L}	$75.0 \pm 1.9_{L}$	$75.2 \pm 2.5_{L}$	$76.5 \pm 1.5_{W}$	$75.6 \pm 1.8_{W}$
17	72.4 ± 2.8	73.1 ± 3.1	$\mathbf{76.3 \pm 2.4}^{W}$	73.8 ± 2.7	$72.8 \pm 1.8_{L}$	$74.9 \pm 2.3^{W}_{L}$	74.1 ± 2.5	72.9 ± 2.1
18	79.6 ± 2.1	80.2 ± 2.5	80.4 ± 1.7	80.8 ± 2.2	79.2 ± 2.3	$\mathbf{81.1 \pm 2.9}^{W}$	80.8 ± 3.1	80.0 ± 2.7
19	92.4 ± 1.4	92.3 ± 1.5	92.5 ± 1.9	91.6 ± 1.8^{L}	92.4 ± 1.4	$\mathbf{92.6 \pm 2.1}$	$92.3 \pm 1.5_{W}$	$92.5 \pm 1.5_{W}$
20	88.2 ± 0.9	88.0 ± 0.8	86.7 ± 1.4^{L}	86.6 ± 1.1^{L}	$88.1 \pm 0.9_{W}$	$\mathbf{88.5 \pm 1.3}_{W}$	$88.2 \pm 0.8_{W}$	$87.8 \pm 1.1_{W}$
Avg. acc	85.0	84.8	83.5	85.9	85.1	85.4	**86.5**	86.1
Semisuperv. versus superv. (W/T/L)			4/7/9	11/6/3	2/17/1	3/17/**0**	11/9/**0**	11/9/**0**
Safe semisuperv. versus semisuperv. (W/T/L)					9/9/2	9/8/3	8/12/**0**	5/14/1

Dataset	SVM	LS-SVM	TSVM	SSCCM	S3VM_*us*	S4VM_*s*	SA-SSCCM_*cv*	SA-SSCCM_*en*
1	96.7 ± 0.4	96.7 ± 0	96.6 ± 0.4	96.7 ± 0	96.4 ± 0.5	96.6 ± 0.4	$\mathbf{97.0 \pm 0}$	$\mathbf{97.0 \pm 0}$
2	86.0 ± 1.4	87.8 ± 0.5	81.4 ± 1.8^{L}	87.4 ± 0.7^{L}	85.7 ± 1.8^{W}	$85.9 \pm 1.2_{W}$	$\mathbf{87.9 \pm 0.7}_{W}$	87.6 ± 0.7
3	79.6 ± 0	84.7 ± 0	80.0 ± 0^{W}	86.5 ± 0^{W}	$79.4 \pm 0^{L}_{L}$	$80.5 \pm 0^{W}_{W}$	86.5 ± 0^{W}	$\mathbf{86.9 \pm 0}^{W}$
4	66.3 ± 2.9	65.3 ± 3.4	66.5 ± 2.7	65.7 ± 3.2	66.0 ± 3.7	$\mathbf{66.7 \pm 3.5}$	$66.5 \pm 3.7^{W}_{W}$	65.9 ± 3.2
5	91.5 ± 1.7	91.9 ± 0.9	81.6 ± 2.0^{L}	92.2 ± 0.7^{W}	$89.6 \pm 0.9^{L}_{W}$	$90.7 \pm 0.8_{W}$	$\mathbf{92.3 \pm 0.6}^{W}$	92.2 ± 0.8^{W}
6	92.2 ± 3.4	95.0 ± 1.1	92.0 ± 1.2	$\mathbf{96.1 \pm 1.0}^{W}$	92.4 ± 1.1	$94.3 \pm 0.9^{W}_{W}$	$\mathbf{96.1 \pm 1.0}^{W}$	96.0 ± 1.0^{W}
7	92.4 ± 1.7	94.1 ± 2.3	88.2 ± 3.2^{L}	$\mathbf{94.7 \pm 1.5}$	$92.5 \pm 2.3_{W}$	$91.9 \pm 1.9_{W}$	$\mathbf{94.7 \pm 2.2}$	94.5 ± 2.3
8	89.6 ± 2.6	92.4 ± 3.3	85.6 ± 2.6^{L}	94.5 ± 2.8^{W}	$89.8 \pm 3.1_{W}$	$89.3 \pm 3.4_{W}$	94.5 ± 2.2^{W}	94.5 ± 2.8^{W}
9	94.9 ± 0.7	95.1 ± 0.4	89.8 ± 1.6^{L}	96.9 ± 0.9^{W}	$94.9 \pm 1.1_{W}$	$94.8 \pm 0.9_{W}$	97.1 ± 0.4^{W}	96.5 ± 0.8^{W}
10	97.2 ± 1.2	98.6 ± 0.8	99.8 ± 0.2^{W}	$\mathbf{100 \pm 0}^{W}$	98.4 ± 0.6^{W}	99.8 ± 0.9^{W}	$\mathbf{100 \pm 0}^{W}$	$\mathbf{100 \pm 0}^{W}$
11	95.3 ± 0.2	95.8 ± 0.4	95.0 ± 0.2	96.0 ± 0.4	95.5 ± 0.2	95.3 ± 0.2	$\mathbf{96.3 \pm 0.5}^{W}$	95.9 ± 0.6
12	73.2 ± 1.8	75.1 ± 1.0	71.2 ± 1.3^{L}	74.2 ± 1.0^{L}	$72.6 \pm 1.0_{W}$	$73.4 \pm 1.0_{W}$	$\mathbf{75.4 \pm 1.7}_{W}$	$75.2 \pm 1.9_{W}$
13	71.6 ± 3.4	67.6 ± 2.9	53.1 ± 3.2^{L}	72.7 ± 3.7^{W}	$71.0 \pm 2.3_{W}$	$71.4 \pm 2.9_{W}$	72.9 ± 2.4^{W}	71.8 ± 3.1^{W}
14	99.6 ± 0.1	99.7 ± 0.2	99.7 ± 0.1^{W}	99.7 ± 0.1	$99.4 \pm 0.2^{L}_{L}$	$99.6 \pm 0.1_{L}$	$\mathbf{99.8 \pm 0}^{W}_{W}$	99.7 ± 0.1^{W}
15	67.4 ± 3.2	68.6 ± 2.3	65.8 ± 2.8^{L}	70.2 ± 3.1^{W}	$67.2 \pm 3.4_{W}$	$67.2 \pm 2.8_{W}$	$\mathbf{70.8 \pm 2.6}^{W}$	69.8 ± 3.8^{W}
16	69.6 ± 6.8	68.4 ± 5.1	$\mathbf{78.1 \pm 2.2}^{W}$	77.3 ± 4.1^{W}	$69.8 \pm 3.2_{L}$	$75.2 \pm 2.9^{W}_{L}$	77.9 ± 2.1^{W}	$75.2 \pm 3.3^{W}_{L}$
17	61.3 ± 8.6	64.2 ± 7.8	$\mathbf{66.3 \pm 5.8}^{W}$	62.8 ± 6.9^{L}	$61.9 \pm 6.2_{L}$	$62.3 \pm 3.3^{W}_{L}$	$65.8 \pm 4.2^{W}_{W}$	$64.1 \pm 3.6_{W}$
18	87.6 ± 2.8	87.2 ± 2.5	87.5 ± 2.5	88.3 ± 2.2^{W}	87.5 ± 2.9	87.2 ± 3.1	88.8 ± 2.1^{W}	$\mathbf{89.2 \pm 3.0}^{W}_{W}$
19	95.1 ± 1.6	94.8 ± 1.8	95.5 ± 1.5	95.5 ± 1.6	95.1 ± 1.4	$\mathbf{96.2 \pm 1.3}^{W}$	95.8 ± 0.9^{W}	95.2 ± 1.8
20	84.6 ± 2.1	85.7 ± 1.8	$\mathbf{90.8 \pm 1.5}^{W}$	87.9 ± 2.6^{W}	$86.8 \pm 2.8^{W}_{L}$	89.4 ± 1.5^{W}	88.2 ± 1.5^{W}	87.8 ± 2.1^{W}
Avg. acc	84.6	85.3	83.2	86.7	84.3	85.4	**87.2**	86.8
Semisuperv. versus superv. (W/T/L)			6/6/8	11/6/3	2/15/3	7/13/**0**	16/4/**0**	12/8/**0**
Safe semisuperv. versus semisuperv. (W/T/L)					8/7/5	10/7/3	3/17/**0**	3/16/1

to 100, 0.1, and 1, respectively, and ε is set to 10^{-3}. Both linear and Gaussian kernels are used, the width parameter σ in the Gaussian kernel is set to the average distance between instances denoted by δ when ten instances are labeled, and selected through five-fold cross validation from $\{0.25, 0.5, 1, 2, 4\}\delta$ over labeled training data in the case of 100 labeled instances.

In the case of ten labeled instances, we set the value of λ in SA-SSCCM, respectively, by fixing it to 0.5 and the ensemble strategy, corresponding to SA-SSCCM_*fix* and SA-SSCCM_*en*, respectively. In the case of 100 labeled instances, we set its value, respectively, by the ensemble and cross-validation strategy, corresponding to SA-SSCCM_*en* and SA-SSCCM_*cv*, respectively. In addition, in SA-SSCCM_*en*, the candidate values for λ is uniformly selected from the interval [0, 1] with a space of 0.2, i.e., [0, 0.2, 0.4, 0.6, 0.8, 1].

2) Performance Comparison: The performances with 10 and 100 labeled instances are shown in Tables III and IV, respectively, in which the upper-half and lower-half parts correspond to performances adopting the linear and Gaussian kernels, respectively. In each part, each row (except the last three ones) gives the performances (classification accuracy and standard deviation) of individual methods over each data set. In each row, the bold value represents the best performance over each data set, the value with a superscript W/L represent that the semi-supervised or safe semi-supervised method performs better/worse than the corresponding supervised counterpart with significant difference by t-test, and the value with a subscript W/L represents that the safe semi-supervised method performs better/worse than the original semi-supervised method with significant difference by t-test. The third-to-last row provides the average accuracies of individual methods, respectively, over all datasets. The second-to-last row gives the numbers of cases in which the semi-supervised or safe semi-supervised method performs better/comparable/worse (W/T/L) than the corresponding supervised counterpart, and the last row gives the numbers of cases in which the safe semi-supervised method performs better/comparable/worse (W/T/L) than the original semi-supervised method.

From these tables, we can make several observations as follows.

a) The overall performances of SVM and LS-SVM are comparable. The overall performances of SSCCM (based on LS-SVM) are, however, better than those of TSVM (based on SVM), as the modified clustering assumption can better capture the real-data distribution than the clustering assumption. It is exactly the reason why we develop a safety-aware semi-supervised classification method based on SSCCM, though such a mechanism can also be similarly applied to other semi-supervised classification methods such as TSVM.
b) The chance of performance degeneration in S3VM_*us* is much smaller than that of TSVM. Specifically, TSVM performs worse than SVM totally in 32 out of the 80 cases, and S3VM_*us* performs worse than SVM in only five cases. Simultaneously, S4VM never performs worse than SVM, and its overall performance is highly competitive with that of TSVM. Therefore, S3VM_*us* and S4VM can be adopted for safe semi-supervised classification.
c) SSCCM performs worse than LS-SVM totally in 15 cases, whereas SA-SSCCM performs better than LS-SVM in all cases. Therefore, SA-SSCCM can be adopted for safe semi-supervised classification.
d) It is expected that the safe semi-supervised methods can perform no worse than the supervised counterparts, and simultaneously, perform no worse than the original semi-supervised methods. S3VM_*us* performs worse than TSVM totally in 22 cases, and S4VM performs worse than TSVM totally in 18 cases, whereas SA-SSCCM perform worse than SSCCM in only two cases. Therefore, SA-SSCCM can be expected to achieve the new goal of safe semi-supervised classification.
e) The overall performance of SA-SSCCM is better than those of both S3VM_*us* and S4VM, indicating the high competitiveness of SA-SSCCM for safe semi-supervised classification.

3) Computation Efficiency: To illustrate the computation efficiency of SA-SSCCM, we provide the running time comparison of SA-SSCCM with the off-the-shelf safe semi-supervised classification methods S3VM_*us* and S4VM. For S4VM, we select S4VM_*s* based on representative sampling as it is much faster in learning than S4VM_*a* based on global simulated annealing search [17].

Fig. 2 shows the training times of SA-SSCM_*fix*, SA-SSCCM_*en*, S3VM_*us,* and S4VM with the linear kernel over nine representative datasets in the case of ten labeled instances. From Fig. 2, SA-SSCCM_*fix* has comparable time cost with S3VM_*us*, and much lower time cost than S4VM_*s*. SA-SSCCM_*en* scales worse than S3VM_*us*, but better than S4VM_*s* in most cases. Therefore, the efficiency of SA-SSCCM is competitive compared with the off-the-shelf safe semi-supervised classification methods.

V. Conclusion

Semi-supervised classification method may yield worse performance than their supervised counterpart because of using the unlabeled data in some cases. Therefore, it naturally reduced the confidence for applying semi-supervised classification methods to real applications. To address this problem without sacrificing the semi-supervised classification performance, we invented a safety-control mechanism by adaptively controlling the tradeoff between semi-supervised and supervised learning in terms of the available unlabeled data. In implementation, we developed a safety-aware semi-supervised classification method named safety-aware SSCCM based on our recent SSCCM. The prediction by SA-SSCCM accomplished a tradeoff between those by SSCCM and LS-SVM in terms of the unlabeled data. Finally, empirical results over several real datasets showed that the overall performance of SA-SSCCM was better than those of both SSCCM and LS-SVM, and in addition, the performance of SA-SSCCM was never significantly inferior to that of LS-SVM, and seldom significantly inferior to that of SSCCM. Simultaneously, the computation efficiency of SA-SSCCM was competitive compared with the off-the-shelf safe semi-supervised classification methods.

The value of parameter λ was a critical issue in SA-SSCCM controlling the tradeoff between SSCCM and LS-SVM. In this paper, it could be selected by cross validation when the labeled data were sufficient and by the ensemble strategy, otherwise. Nevertheless, seeking the optimal values for parameters in semi-supervised learning is still an open problem worth studying, and further establishing a solid theory regarding with it is absolutely necessary, it naturally becomes an important future work for us.

REFERENCES

[1] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowl. Inf. Syst.*, vol. 24, no. 3, pp. 415–439, 2010.

[2] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. San Mateo, CA, USA: Morgan Kaufmann, 2009.

[3] X. Zhu, "Semi-supervised learning literature survey," Ph.D. dissertation, Dept. Comput. Sci., Wisconsin-Madison, Univ., Madison, WI, USA, Jul. 2008.

[4] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[5] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "Semi-boost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2000–2014, Nov. 2009.

[6] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn.*, 1999, pp. 200–209.

[7] G. Fung and O. L. Mangasarian, "Semi-supervised support vector machine for unlabeled data classification," *Optim. Methods Softw.*, vol. 15, no. 1, pp. 99–105, 2001.

[8] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, Jan. 2006.

[9] Y.-F. Li, J. Kwok, and Z.-H. Zhou, "Semi-supervised learning using label mean," in *Proc. 26th Int. Conf. Mach. Learn.*, 2009, pp. 633–640.

[10] Y. Bengio, O. B. Alleau, and N. L. Roux, "Label propagation and quadratic criterion," in *Semi-Supervised Learning*, Cambridge, MA, USA: MIT Press, 2006, pp. 193–216.

[11] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Dept. Lang. Technol., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CALD-02-107, 2002.

[12] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 19–26.

[13] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 2399–2434, 2006.

[14] B. Pfahringer, "A semi-supervised spam mail detector," in *Proc. 17th Eur. Conf. Mach. Learn. 10th Eur. Conf. Principles Pract. Knowl. Discovery Databases*, 2006, pp. 1–5.

[15] N. Kasabov and S. Pang, "Transductive support vector machines and applications in bioinformatics for promoter recognition," *Neural Inf. Process. Lett. Rev.*, vol. 3, pp. 31–38, May 2004.

[16] T. Yang and C. E. Priebe, "The effect of model misspecification on semi-supervised classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2093–2103, Oct. 2011.

[17] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1081–1088.

[18] Y.-F. Li and Z.-H. Zhou, "Improving semi-supervised support vector machines through unlabeled instances selection," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 500–505.

[19] Y. Wang, S. Chen, and Z.-H. Zhou, "New semi-supervised classification method based on modified cluster assumption," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 689–702, May 2012.

[20] B. Yang and S. Chen, "Disguised discrimination of locality-based unsupervised dimensionality reduction," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 24, no. 7, pp. 1011–1025, 2010.

[21] J. Gorski and F. Pfeuffer, "Biconvex sets and optimization with biconvex functions: A survey and extensions," *Math. Methods Oper. Res.*, vol. 66, no. 3, pp. 373–407, 2007.

[22] Z. Xu, R. Jin, I. King, M. R. Lyu, and Z. Yang, "Adaptive regularization for transductive support vector machine," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2125–2133.

[23] K. Chen and S. Wang, "Regularized boost for semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 281–288.

[24] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, Jan. 2011.

Yunyun Wang received the B.S. degree in computer science and technology from Anhui Normal University, Wuhu, China, in 2006, and the Ph.D. degree in computer science and engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2012.

She is currently an Instructor with the Department of Computer Science and Engineering, Nanjing University of Posts & Telecommunications, Nanjing. Her current research interests include pattern recognition, machine learning, and neural computing.

Songcan Chen received the B.S. degree in mathematics from Hangzhou University (now Zhejiang University), Hangzhou, China, in 1983, the M.S. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, in 1985, and the Ph.D. degree in communication and information systems in 1997.

He was with the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1986. Since 1998, he has been a Full-Time Professor with the Department of Computer Science & Engineering, NUAA. He has authored or co-authored over 130 scientific peer-reviewed papers. His current research interests include pattern recognition, machine learning, and neural computing.