New Semi-Supervised Classification Method Based on Modified Cluster Assumption

Yunyun Wang, Songcan Chen, and Zhi-Hua Zhou, *Senior Member, IEEE*

***Abstract*— The *cluster assumption*, which assumes that "similar instances should share the same label," is a basic assumption in semi-supervised classification learning, and has been found very useful in many successful semi-supervised classification methods. It is rarely noticed that when the cluster assumption is adopted, there is an implicit assumption that every instance should have a crisp class label assignment. In real applications, however, there are cases where it is difficult to tell that an instance definitely belongs to one class and does not belong to other neighboring classes. In such cases, it is more adequate to assume that "similar instances should share similar label memberships" rather than sharing a crisp label assignment. Here "label memberships" can be represented as a vector, where each element corresponds to a class, and the value at the element expresses the likelihood of the concerned instance belonging to the class. By adopting this modified cluster assumption, in this paper we propose a new semi-supervised classification method, that is, semi-supervised classification based on class membership (SSCCM). Specifically, we try to solve the decision function and adequate label memberships for instances simultaneously, and constrain that an instance and its "local weighted mean" (LWM) share the same label membership vector, where the LWM is a robust image of the instance, constructed by calculating the weighted mean of its neighboring instances. We formulate the problem in a unified objective function for the labeled, unlabeled data and their LWMs based on the square loss function, and take an alternating iterative strategy to solve it, in which each step generates a closed-form solution, and the convergence is guaranteed. The solution will provide both the decision function and the label membership function for classification, their classification results can verify each other, and the reliability of semi-supervised classification learning might be enhanced by checking the consistency between those two predictions. Experiments show that SSCCM obtains encouraging results compared to state-of-the-art semi-supervised classification methods.**

***Index Terms*— Cluster assumption, iteration, label membership function, local weighted mean, semi-supervised classification.**

I. Introduction

IN MANY real applications such as image analysis, drug discovery and web page analysis, the acquisition of labeled data is usually expensive and time-consuming, while the collection of unlabeled data is relatively much easier [1], [2]. Consequently, semi-supervised learning, and more specifically, semi-supervised classification, which learns from a combination of labeled and unlabeled data for better performance than using the labeled data alone, has attracted considerable attention. During the past decades, lots of semi-supervised classification methods have been developed, and comprehensive reviews can be found in [3]–[5].

Roughly speaking, semi-supervised classification approaches can be categorized into four paradigms, that is, generative approaches, semi-supervised large margin approaches, graph-based approaches, and disagreement-based approaches [5], [6], while this paper focuses on the second one. Generally, semi-supervised classification methods attempt to exploit the intrinsic data distribution disclosed by the unlabeled data, and the data distribution information is generally helpful to construct a better prediction model. To exploit unlabeled data, some assumptions need to be adopted. One of the most common assumptions is the cluster assumption, which assumes that "similar instances should share the same label" [3], [4], and [7]. This assumption has been adopted by many semi-supervised classification methods and has been found useful. It is rarely mentioned that when adopting this assumption, there is an implicit assumption, that is, every instance should have a crisp class label assignment. In real applications, however, there are cases where it is difficult to tell that an instance definitely belongs to one class and does not belong to other neighboring classes. For example, in image segmentation, the boundary pixels can belong to either class, in book classification, the classic book "statistical learning theory" of Vapnik [8] can be classified into either statistic category or machine learning category.

Researchers have also found that the utilization of unlabeled data is not always helpful, sometimes it may even hurt the performance [9]–[11]. Usually this hurting is attributed to the failure of the presumed model assumption or data distribution assumption [12]–[14]. Recently, some efforts have been devoted to the safe utilization of unlabeled data [15], [16]. In this paper, we propose to consider a modified cluster assumption, that is, "similar instances should share similar label memberships," which is able to capture the real data distribution better in many cases, thus provide more choices for data distribution assumption. Here "label memberships" can be represented as a vector, where each element corresponds to a class, and the value at the element expresses the likelihood of the concerned instance belonging to the class. By adopting this modified cluster assumption, we further develop a new

Manuscript received December 7, 2010; revised January 20, 2012; accepted January 21, 2012. Date of publication February 16, 2012; date of current version May 2, 2012. This work was supported in part by the National Science Foundation of China under Grant 61035003, Grant 61073097, and Grant 60973097, the National Fundamental Research Program of China under Grant 2010CB327903, and a Jiangsu 333 High-Level Talent Cultivation Program.

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

Z.-H. Zhou is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210046, China (e-mail: zhouzh@nju.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.2012.2186825

semi-supervised classification method named semi-supervised classification based on class membership (SSCCM), which seeks both the decision function and label memberships for instances to all classes simultaneously. During the learning process, we further constrain that an instance and its "local weighted mean" (LWM) share the same label membership vector according to the local learning principle [17], [18], where the LWM is a robust image of the instance, constructed by calculating the weighted mean of its neighboring instances [19]. We choose the square loss function here due to its simplicity and formulate the problem into a unified objective function for the labeled, unlabeled data and their LWMs. We take an alternating iterative strategy to solve it, in which each step generates a closed-form solution, and the convergence is guaranteed. The solution will provide both the decision function and label membership function. Notice that classification can be made by the decision function as well as the label membership function. This offers another advantage of SSCCM, that is, the two classification results can verify each other, and the reliability of semi-supervised classification might be enhanced by checking the consistency between those two predictions. Though we adopt the square loss function here, other loss functions for classification can also be used to develop different semi-supervised classification methods based on the modified cluster assumption. Finally, experiments on both toy and real datasets show that SSCCM has competitive results compared to the state-of-the-art semi-supervised classification methods.

Besides, it is worth noting that though the modified clustering assumption resembles the fuzzy assignment in clustering learning, SSCCM differs from those fuzzy semi-supervised clustering methods [20]–[22]. The reason is that semi-supervised clustering addresses the problem of exploiting additional labeled data to guide and adjust the clustering of unlabeled data [22]–[24]. While SSCCM belongs to semi-supervised classification methods, which aims to exploit unlabeled data along with labeled data to obtain a better classification model, and thus better predictions on unseen data [6], [25]. Actually, semi-supervised clustering and classification can be viewed as two cousins in semi-supervised learning [6].

The rest of this paper is organized as follows. Section II introduces some related work. Section III presents the motivation and formulation of SSCCM. Section IV presents the optimization and algorithm description of SSCCM. Section V shows the experimental results. Some conclusions are drawn in Section VI.

II. Related Work

During the past decade, lots of semi-supervised classification methods have been developed by adopting the cluster assumption, among which there are the maximum margin ones, e.g., semi-supervised SVM [26], TSVM [27], and meanS3VM [28], etc.

Given labeled data $X_l = \{x_i\}_{i=1}^{n_l}$ with corresponding labels $Y = \{y_i\}_{i=1}^{n_l}$, and unlabeled data $X_u = \{x_j\}_{j=n_l+1}^{n}$, where each $x_i \in R^d$ and $y_i \in \{-1, +1\}$. With a linear decision function $g(x)$, the optimization problem of TSVM can be formulated as

$$\begin{aligned}\min_{g,\ y_j,\ \xi_i,\ \xi_j}\ & \frac{1}{2}\|g\|_{\mathcal{H}}^2 + C\sum_{i=1}^{n_l}\xi_i + C^*\sum_{j=n_l+1}^{n}\xi_j \\ \text{s.t. }\ & y_i g(x_i) \geq 1-\xi_i,\ \xi_i \geq 0,\ i = 1,\ldots,n_l \\ & y_j g(x_j) \geq 1-\xi_j,\ \xi_j \geq 0, \\ & y_j \in \{-1,+1\},\ j = n_l+1,\ldots,n \end{aligned} \tag{1}$$

where $\|\bullet\|_{\mathcal{H}}$ is a norm in the Reproducing Kernel Hilbert Space (or kernel space), ξ_i and ξ_j are error tolerances corresponding to the labeled and unlabeled data, respectively, C and C^* are trade-off parameters between the empirical errors and function complexity. As a result, TSVM seeks the decision function and class labels for unlabeled data simultaneously so that the classification hyper-plane separates both labeled and unlabeled data with the maximum margin, and similar instances would share the same class label.

The optimization problem held in TSVM is a non-linear non-convex optimization problem [29], and researchers have devoted much effort to improve its efficiency. For example, Joachims [27] proposed a label-switch-retraining procedure to optimize the decision function and instance labels iteratively. Chapelle *et al.* [30] replaced the hinge loss in TSVM by a smooth function and solved the problem by gradient descend. Other examples include the use of concave-convex procedure [31], convex relaxation [32], deterministic annealing [33], and branch-and-bound [34], etc.

Recently, Li *et al.* [28] stated that semi-supervised SVM, with knowledge of the label means for unlabeled data, is closely related to supervised SVM with all data labeled. Thus they developed meanS3VM by estimating the label means for unlabeled data and maximizing the margin between those label means. MeanS3VM finally achieves competitive performance and improved efficiency compared with TSVM, as well as some other semi-supervised classification methods.

Obviously, all the above methods implicitly constrain that each instance belongs to a single class, even for boundary instances difficult to be assigned with a crisp class label. In this paper, we present a new semi-supervised classification method through adopting a modified cluster assumption, which allows each instance to belong to all classes with the corresponding memberships. Though classification is made with the class corresponding to the largest membership, the membership information of other classes is also helpful in the learning process.

III. Semi-Supervised Classification Based on Class Membership

In this section, we will present the motivation and formulation of SSCCM in separate sub-sections.

A. Motivation

There are instances which are difficult to be assigned to a single class in real applications, e.g., those boundary instances. In those cases, since the cluster assumption implicitly constrains each instance to have a crisp label assignment, it cannot reflect the real data distribution adequately, and is likely

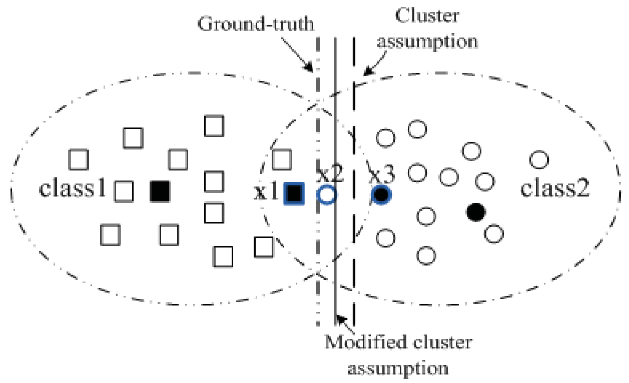


Fig. 1. Toy dataset with “□” and “◯” denoting the unlabeled instances, and “■” and “●” denoting the labeled instances in individual classes, respectively. The decision boundaries w.r.t. the “cluster assumption” and “modified cluster assumption” are also depicted, respectively, compared with the ground-truth boundary.

to be violated over those boundary instances. Accordingly, when applied to semi-supervised classification, the cluster assumption would lead to poor prediction for those boundary instances, especially when some labeled instances lie near the boundary and further “mislead” the classification. However, when adopting the modified cluster assumption, each instance would have label memberships to all given classes rather than a single class label, in this way, the impact of those “misleading” labeled instances can be mitigated.

An illustration can be seen in Fig. 1, in which the unlabeled instances in individual classes are represented by “□” and “◯,” respectively, and the corresponding labeled instances are represented by “■” and “●,” respectively. From Fig. 1, it can be easily observed that the labeled instance x1 in class 1 lies in the overlap region between classes, and moreover, it is closer to the class boundary than the labeled instances in class 2, thus would “mislead” the classification. Consequently, through adopting the cluster assumption, the unlabeled instance x2 (in class 2) would be assigned to class 1 since it is closer to x1 than x3, and thus the corresponding decision boundary would be closer to class 2 and deviate the ground-truth boundary. However, when adopting the modified cluster assumption, x2 would be assigned label memberships to both classes, though a larger membership value to class 1, thus the impact of x1 on the decision boundary can be mitigated, and finally a decision boundary closer to the ground-truth boundary can be obtained.

As a result, it is more reasonable to adopt the modified cluster assumption in semi-supervised classification, and in what follows, we will develop a new semi-supervised classification method based on such assumption.

B. Formulation

Given labeled data $X_l = \{x_i\}_{i=1}^{n_l}$ with the corresponding labels $Y = \{y_i\}_{i=1}^{n_l}$, and unlabeled data $X_u = \{x_j\}_{j=n_l+1}^{n}$, where each $x_i \in R^d$ and $n_u = n - n_l$. The LWM[1] of each instance x_i is defined by

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

where $Ne(x_i)$ denotes the neighbor set of x_i consisting of its k nearest neighbors measured by the Euclidean distance, and S_{ij} is an amount monotonically decreasing as the distance between x_i and x_j increases, e.g., $S_{ij} = \exp(-\|x_i - x_j\|^2)$. $\hat{X}_l = \{\hat{x}_i\}_{i=1}^{n_l}$ and $\hat{X}_u = \{\hat{x}_j\}_{j=n_l+1}^{n}$ denote the LWMs for the labeled and unlabeled data, respectively. The encodings for the C classes are denoted by $\{r_k\}_{k=1}^{C}$, where $y_i = r_k$ if x_i belongs to the kth class. Here both the data labels and the class encodings are encoded by the one-of-c rule so that SSCCM can be directly applied to multi-class classification tasks. Specifically, both data labels and class encodings are C-dimension vectors, the kth entry of each y_i is assigned 1 if x_i belongs to the kth class, and the rest are 0, the kth entry of each r_k is set to 1, and the rest are 0. Aside from the decision function $f(x)$, we also define a label membership function $\upsilon(x)$, for an arbitrary instance x_i, $\upsilon(x_i) \in R^C$ and the kth component $\upsilon_k(x_i)$ expresses the likelihood of x_i belonging to the kth class. Finally, through adopting the modified cluster assumption, and constraining that each instance and its LWM share the same label membership vector according to the local learning principle[2] [17], [18], SSCCM can be formulated as

$$\begin{aligned}
\min_{f,\ \upsilon_k(x_i)} & \sum_{k=1}^{C}\sum_{i=1}^{n} \upsilon_k(x_i)^b \, \|f(x_i) - r_k\|^2 \\
& + \lambda_s \sum_{k=1}^{C}\sum_{i=1}^{n} \upsilon_k(x_i)^b \, \left\|f(\hat{x}_i) - r_k\right\|^2 + \lambda \, \|f\|_{\mathcal{H}}^2 \\
\text{s.t.} & \sum_{k=1}^{C} \upsilon_k(x_i) = 1 \\
& 0 \le \upsilon_k(x_i) \le 1, \ k = 1, \ldots, C, \ i = 1, \ldots, n
\end{aligned} \tag{3}$$

where λ and λ_s are regularization parameters, and b is a weighting exponent on the label memberships. The second term of the objective function in (3) characterizes the consistency between the predictions (or label membership vectors) for each instance and its LWM adjusted by λ_s, and the third term characterizes the model complexity adjusted by λ.

In fact, b controls the degree or uncertainty of instances belonging to multiple classes. More specifically, when $b = 1$, each label membership $\upsilon_k(x_i)$ takes its value from {0, 1}, thus SSCCM degenerates to its hard version in which each instance belongs to a single class. On the other hand, when b approaches infinity, each instance would have equal memberships to all classes. However, in this paper, we concentrate on developing new classification methods based on the modified cluster assumption, and simply set $b = 2$ hereafter.

Note that the Euclidean distance is chosen here for calculating the LWMs, but actually, other suitable distance measures can also be adopted. Moreover, though we use the square loss

[1] The LWM of each instance x_i is actually a robust image of x_i by its k-nearest neighbors, and derived by minimizing $\varepsilon_i = \sum_{x_j \in Ne(x_i)} S_{ij} \|\hat{x}_i - x_j\|^2$. Set the derivative of each ε_i w.r.t. $\hat{x}_i$ to zero, the formulation of LWM in (2) is obtained. In general, the LWMs are nearby with the corresponding original instances.

[2] From the local learning principle [17], [18], the label (output) of any instance can be estimated by its neighbors, in other words, the instance and its neighbors should share the same label [35]. As a result, each instance and its LWM should share the same label membership vector.

function here, other loss functions for classification can be adopted as well to develop different semi-supervised classification methods based on the modified cluster assumption.

For labeled instances, the label memberships are

$$\upsilon_k(x_i) = \begin{cases} 1, & if\ x_i \in X_k \\ 0, & else \end{cases}, \quad i = 1, \ldots, n_l, \ k = 1, \ldots, C \quad (4)$$

where X_k denotes the subset of instances belonging to the kth class. Then (3) can be re-written as

$$\begin{aligned} \min_{f,\ \upsilon_k(x_j)} \ & \sum_{i=1}^{n_l} \|f(x_i) - y_i\|^2 + \lambda_s \sum_{i=1}^{n_l} \left\| f(\hat{x}_i) - y_i \right\|^2 \\ & + \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(x_j) - r_k \right\|^2 \\ & + \lambda_s \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(\hat{x}_j) - r_k \right\|^2 + \lambda \|f\|_{\mathcal{H}}^2 \\ \text{s.t.} \ & \sum_{k=1}^{C} \upsilon_k(x_j) = 1 \\ & 0 \le \upsilon_k(x_j) \le 1, \ k = 1, \ldots, C, \ j = n_l + 1, \ldots, n. \end{aligned} \quad (5)$$

As a result, through adopting the modified cluster assumption, each instance in SSCCM can belong to all given classes with the corresponding memberships, moreover, each instance and its LWM would share the same label memberships.

IV. Optimization and Algorithm Description

In this section, we will present the optimization and algorithm description for SSCCM in separated sub-sections.

A. Optimization

The optimization problem of SSCCM is non-convex with respect to (f, υ), and in this paper, we solve it through an alternating iterative strategy to seek the decision function $f(x)$ and label membership function $\upsilon(x)$, respectively. Fortunately, each step has a closed-form solution.

For fixed $\upsilon(x)$, the optimization problem of SSCCM can be re-written as

$$\begin{aligned} \min_{f} \ & \sum_{i=1}^{n_l} \|f(x_i) - y_i\|^2 + \lambda_s \sum_{i=1}^{n_l} \left\| f(\hat{x}_i) - y_i \right\|^2 \\ & + \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(x_j) - r_k \right\|^2 \\ & + \lambda_s \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(\hat{x}_j) - r_k \right\|^2 + \ \lambda \|f\|_{\mathcal{H}}^2. \end{aligned} \quad (6)$$

Similar to (2), the LWM of each x_i in the kernel space is defined to be $\widehat{\phi(x_i)} = \sum_{x_j \in Ne(x_i)} S_{ij} \phi(x_j) / \sum_{x_j \in Ne(x_i)} S_{ij}$, then for each instance, its LWM is a linear combination of its neighbors, and thus a linear combination of the given instances in the kernel space. Hence, the minimizer of (6) has the form $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$ [36] based on the Representer Theorem, where each $\alpha_i \in R^{C \times 1}$, and the solution is

$$\begin{aligned} \alpha = \ & (Y K_l^T + L \hat{V} J^T K_u^T + \lambda_s Y \bar{K}_l^T + \lambda_s L \hat{V} J^T \bar{K}_u^T) \\ & (K_l K_l^T + K_u J \hat{V} J^T K_u^T + \lambda_s \bar{K}_l \bar{K}_l^T + \lambda_s \bar{K}_u J \hat{V} J^T \bar{K}_u^T \\ & + \lambda K)^{-1} \end{aligned} \quad (7)$$

where $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in R^{C \times n}$ is the Lagrange multiplier matrix. $K = [K_l \ K_u] = \begin{bmatrix} K_{ll} & K_{lu} \\ K_{lu}^T & K_{uu} \end{bmatrix}$ where $K_{ll} = \langle \phi(X_l), \ \phi(X_l) \rangle_{\mathcal{H}}$, $K_{lu} = \langle \phi(X_l), \ \phi(X_u) \rangle_{\mathcal{H}}$ and $K_{uu} = \langle \phi(X_u), \ \phi(X_u) \rangle_{\mathcal{H}}$. $\bar{K} = [\bar{K}_l \ \bar{K}_u] = \begin{bmatrix} \bar{K}_{ll} & \bar{K}_{lu} \\ \bar{K}_{ul} & \bar{K}_{uu} \end{bmatrix}$ where $\bar{K}_{ll} = \langle \phi(X_l), \ \widehat{\phi(X_l)} \rangle_{\mathcal{H}}$, $\bar{K}_{lu} = \langle \phi(X_l), \ \widehat{\phi(X_u)} \rangle_{\mathcal{H}}$, $\bar{K}_{ul} = \langle \phi(X_u), \ \widehat{\phi(X_l)} \rangle_{\mathcal{H}}$ and $\bar{K}_{uu} = \langle \phi(X_u), \ \widehat{\phi(X_u)} \rangle_{\mathcal{H}}$. $J = [\underbrace{I_u \ldots I_u}_{C}] \in R^{n_u \times (C \times n_u)}$ where I_u is a $n_u \times n_u$ identity matrix, $L = [L_1 \ldots L_C] \in R^{C \times (C \times n_u)}$, where each L_k is a $C \times n_u$ matrix with the kth row being an all-one vector and the rest being all-zero vectors. Let $V = [\upsilon(x_1) \ldots \upsilon(x_{n_u})] \in R^{C \times n_u}$ denote the label membership values for the unlabeled data, then $\hat{V}$ denotes a diagonal matrix with the diagonal elements being the squared values of the entries in V arranged by rows.

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

$$\begin{aligned} \min_{\upsilon_k(x_j)} \ & \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(x_j) - r_k \right\|^2 \\ & + \lambda_s \sum_{k=1}^{C} \sum_{j=n_l+1}^{n} \upsilon_k(x_j)^2 \left\| f(\hat{x}_j) - r_k \right\|^2 \\ \text{s.t.} \ & \sum_{k=1}^{C} \upsilon_k(x_j) = 1, \\ & 0 \le \upsilon_k(x_j) \le 1, k = 1, \ldots, C, \ j = n_l + 1, \ldots, n \end{aligned} \quad (8)$$

and the solution is

$$\upsilon_k(x_j) = \frac{1 / \left(\left\| f(x_j) - r_k \right\|^2 + \lambda_s \left\| f(\hat{x}_j) - r_k \right\|^2 \right)}{\sum_{k=1}^{C} 1 / \left(\left\| f(x_j) - r_k \right\|^2 + \lambda_s \left\| f(\hat{x}_j) - r_k \right\|^2 \right)}. \quad (9)$$

Therefore, for an arbitrary instance x, its label membership to the kth class can be derived from

$$\upsilon_k(x) = \frac{1 / \left(\|f(x) - r_k\|^2 + \lambda_s \left\| f(\hat{x}) - r_k \right\|^2 \right)}{\sum_{k=1}^{C} 1 / \left(\|f(x) - r_k\|^2 + \lambda_s \left\| f(\hat{x}) - r_k \right\|^2 \right)}. \quad (10)$$

The detailed derivations for optimizing problems (6) and (8) can be found in Appendices A and B, respectively.

It is easily observed that data prediction can be implemented by either the decision function from $y^* = \arg\max_{k=1,\ldots,C} f_k(x)$, or the label membership function from $y^* = \arg\max_{k=1,\ldots,C} \upsilon_k(x)$. More specifically, $x \in X_k$ by $f(x)$ if $f_k(x) > f_j(x)$, $\forall j = 1, \ldots, C, \ j \neq k$. $x \in X_k$ by $\upsilon(x)$ if $\upsilon_k(x) > \upsilon_j(x)$, or equivalently, $f_k(x) + \lambda_s f_k(\hat{x}) > f_j(x) + \lambda_s f_j(\hat{x})$ for a fixed λ_s, $\forall j = 1, \ldots, C, \ j \neq k$. As a result, when $\lambda_s = 0$, predictions by $f(x)$ and $\upsilon(x)$ are always consistent. When $\lambda_s \neq 0$, the two predictions are consistent if x and $\hat{x}$ share the same label assignment by $f(x)$, i.e., $\arg\max_{k=1,\ldots,C} f_k(x) = \arg\max_{k=1,\ldots,C} f_k(\hat{x})$, or their prediction inconsistency is not so distinct such that $f_j(\hat{x}) - f_k(\hat{x}) < \left(f_k(x) - f_j(x)\right)/\lambda_s, \forall j = 1, \ldots, C,\ j \neq k$. Otherwise, the prediction inconsistency for x and $\hat{x}$ by $f(x)$ is significant, in this case, x is likely to lie near the decision boundary and the prediction for it would be unreliable. Finally, the instances are divided into three categories,

Inherently-Consistent Instance: Instance x and $\hat{x}$ share the same label assignment by $f(x)$ such that the predictions for x by $f(x)$ and $v(x)$ are consistent.

Pseudo-Consistent Instance: x is not an inherently consistent instance, but $f_j(\hat{x}) - f_k(\hat{x}) < (f_k(x) - f_j(x))/\lambda_s$ such that predictions for x by $f(x)$ and $v(x)$ are still consistent.

Inconsistent Instance: Predictions for x by $f(x)$ and $v(x)$ are not consistent.

In short, SSCCM can provide both the decision function and label membership function for prediction, and their respective predictions are usually consistent (inherently consistent or pseudo-consistent). Otherwise, if the two predictions are not consistent, the corresponding instances are likely to lie near the decision boundary, and the predictions for them may be unreliable. In fact, just one function is needed for predicting new instances, and the label membership function is preferred if the likelihoods of instances to individual classes are also expected. However, one can adopt both functions to predict instances, take advantage of their prediction inconsistency to detect those difficultly classified boundary instances, and give special treatments to them, such as manual labeling, to improve the classification reliability. As a result, the predictions of those two functions can verify each other, and the reliability of semi-supervised classification might be enhanced by checking their consistency.

B. Algorithm Description

The optimization of SSCCM follows an alternating iterative strategy. The initial values for the label memberships of unlabeled instances can be obtained by several strategies, e.g., randomization, some fuzzy clustering technique such as FCM, or simply being set to all zeros, in this case, SSCCM actually starts from an initial decision function learned from the labeled data alone. The iteration terminates when $|M^k - M^{k-1}| < \varepsilon M^{k-1}$, where M^k denotes the objective function value at the kth iteration and ε is a pre-defined threshold. The concrete algorithm description of SSCCM is shown in Table I.

SSCCM differs from semi-supervised EM [37], which also utilizes unlabeled data in an iterative style. The reason is that semi-supervised EM is a generative approach, which assumes a model to fit and thus requires a good model assumption, while SSCCM falls into the category of semi-supervised large margin approaches directly seeking a large margin separator. Actually, an iterative style learning process is almost commonly used by all kinds of semi-supervised learning approaches.

Proposition 1: The sequence $\{J(\alpha_k, v_k)\}$ obtained in the above algorithm w.r.t. SSCCM converges.

TABLE I
ALGORITHM DESCRIPTION OF SSCCM

Input	X_l, X_u — the labeled and unlabeled data Y_l — the labels of X_l λ, λ_{s-}– the regularization parameters ε — the iterative termination parameter σ — the kernel parameter Maxiter — the maximum number for iteration
Output	$f(x)$ — the decision function $v(x)$ — the label membership function

Procedure

```
Initialize the label memberships for unlabeled data;
Obtain the initial $\alpha$ by (7);
Obtain $v(x)$ by (10);
Compute the objective function value $M^0$;
For $k = 1\ldots$Maxiter
    Update $\alpha$ by (7);
    Update $v(x)$ by (10);
    Update the objective function $M^k$;
    If $|M^k - M^{k-1}| < \varepsilon M^{k-1}$
        Break, return$f(x)$ and $v(x)$;
    Endif
    Endfor
```

Proof: First, the sequence of objective function values generated by the above algorithm decreases monotonically. In fact, the objective function $J(\alpha, v)$ is biconvex [38] in (α, v). Specifically, for fixed v_k, the objective function is convex in α, thus the optimal α^* can be obtained by minimizing $J(\alpha, v_k)$, or equivalently optimizing (6). Now set $\alpha_{k+1} = \alpha^*$, then $J(\alpha_{k+1}, v_k) = J(\alpha^*, v_k) \leq J(\alpha_k, v_k)$. Simultaneously, with current α_{k+1}, the objective function is convex in v, thus the optimal v^* can be obtained by minimizing $J(\alpha_{k+1}, v)$, or equivalently optimizing (8). Now set $v_{k+1} = v^*$, then $J(\alpha_{k+1}, v_{k+1}) = J(\alpha_{k+1}, v^*) \leq J(\alpha_{k+1}, v_k)$. Finally, $J(\alpha_{k+1}, v_{k+1}) \leq J(\alpha_{k+1}, v_k) \leq J(\alpha_k, v_k)$, $\forall k \in N$. Hence, the consequence $\{J(\alpha_k, v_k)\}$ decreases monotonically.

Further, since the objective function is non-negative, thus lower-bounded. As a result, the sequence $\{J(\alpha_k, v_k)\}$ converges.

However, though the iteration is convergent, it only results in a local minimum, thus an optimization strategy for SSCCM generating a global solution is still worth studying.

V. EXPERIMENT

In this section, we evaluate SSCCM on both toy and real (UCI [39] and Benchmark [3]) datasets by comparing with the state-of-the-art semi-supervised classification methods, as well as a degenerated variant of SSCCM named hardSSCCM (described in detail below). In what follows, we will briefly introduce the compared methods, and show the results on toy and real datasets, respectively, in separated sub-sections. In our experiments, the initial values for the label memberships of unlabeled instances in SSCCM are all set to zeros. Moreover, since the performances by the decision function and label membership function are always comparable with fixed λ_s, we only give the performances by the decision function, along with their consistency rates.

A. Compared Methods

Five state-of-the-art semi-supervised classification methods are compared here, including LapSVM [36], LapRLS [36], TSVM [27], and meanS3VMs [28].

TABLE II
ATTRIBUTE DESCRIPTION OF THE TOY DATASET

Class	Mean	Variance	Sample Number
Class 1	[0, 0]	$\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$	200
Class 2	[3, 0]		200

LapSVM: Laplacian SVM. It adopts the manifold assumption for semi-supervised classification, and seeks a maximum margin decision function which is smooth over the whole data distribution according to the graph Laplacian.

LapRLS: Laplacian RLS. It also adopts the manifold assumption as LapSVM does, but uses the least square loss rather than the hinge loss.

TSVM: Transductive SVM. It adopts the cluster assumption, and seeks the maximum margin boundary on both the labeled and unlabeled data so as to guide the classification boundary passing through the low density region.

MeanS3VM: Semi-supervised SVM based on the label means of the unlabeled data. It also adopts the cluster assumption, and actually contains two versions [28], i.e., **meanS3VM-iter** based on alternating optimization and **meanS3VM-mkl** based on multiple kernel learning.

Furthermore, we also compare SSCCM with a degenerated version hardSSCCM, a method which is identical to SSCCM except that it adopts the cluster assumption, or more specifically, it assigns a crisp class label to each instance rather than class label memberships. It is formulated as

$$\begin{aligned}\min_{f,\ y_j}\ & \sum_{i=1}^{n_l} \|f(x_i) - y_i\|^2 + \lambda_s \sum_{i=1}^{n_l} \left\|f(\hat{x}_i) - y_i\right\|^2 \\ & + \sum_{j=n_l+1}^{n} \left\|f(x_j) - y_j\right\|^2 \\ & +\lambda_s \sum_{j=n_l+1}^{n} \left\|f(\hat{x}_j) - y_j\right\|^2 + \lambda \|f\|_{\mathcal{H}}^2 \\ \text{s.t. } & y_j \in \{e_1, \ldots, e_C\},\ \ j = n_l + 1, \ldots, n. \end{aligned} \tag{11}$$

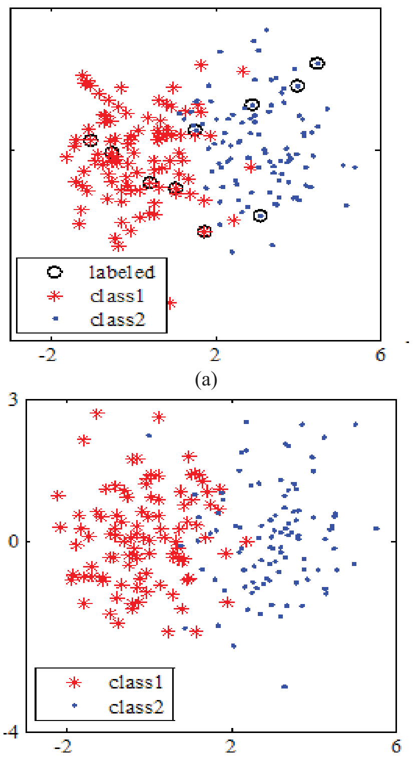
where each e_i is a C-length vector with the ith entry being 1 and the others being 0. Similarly, (11) can be solved by an alternating iterative strategy for optimizing $f(x)$ and y_js, respectively. Specifically, the solution for $f(x)$ with fixed y_js is

$$\begin{aligned}\alpha = & (YK_l^T + Y_uK_u^T + \lambda_s Y\bar{K}_l^T + \lambda_s Y_u\bar{K}_u^T) \\ & (K_lK_l^T + K_uK_u^T + \lambda_s \bar{K}_l\bar{K}_l^T + \lambda_s \bar{K}_u\bar{K}_u^T + \lambda K)^{-1} \end{aligned} \tag{12}$$

with $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$, and the solution for each label vector y_j with fixed $f(x)$ can be obtained by minimizing $\|f(x_j) - y_j\|^2 + \lambda_s \|f(\hat{x}_j) - y_j\|^2$, where each $y_j \in \{e_1, \ldots, e_C\}$, $j = n_l + 1, \ldots, n$.

B. Experiments on Toy Dataset

The toy dataset consists of two Gaussian distributions with its attributes described in Table II; half of the instances in each class are selected for training and the rest for testing. As shown in Fig. 2, symbols "*" and ".", respectively, represent instances in individual classes, and symbol "o" further denotes the labeled instances in both classes for training. Gaussian kernel is adopted here with the width parameter set to 1. For the compared methods, the regularization parameters C_1 and C_2 are fixed to 1 and 0.1, respectively. For SSCCM and hardSSCCM, the regularization parameters λ and λ_s are fixed to 1 and 0.1, respectively, and the neighbor number k and termination parameter ε are set to 5 and 10^{-3}, respectively.



(a)

(b)

Fig. 2. (a) Training dataset with the labeled instances marked. (b) Testing dataset.

Table III lists the testing accuracies of SSCCM and the compared methods, along with the consistency rates between the predictions by $f(x)$ and $\upsilon(x)$, respectively, in SSCCM. From Table III, it can be observed that SSCCM achieves better performance than all the compared methods, and the consistency rate is up to 1. As a result, SSCCM can provide competitive results compared to the state-of-the-art semi-supervised classification methods, and the predictions by $f(x)$ and $\upsilon(x)$, respectively, are usually consistent.

Table IV lists the training and testing accuracies of SSCCM compared with hardSSCCM, with LRLS (on the labeled data alone) and RLS (with all data labeled) as the base lines. From Table IV, we can observe that SSCCM obtains better training and testing performances than LRLS, thus SSCCM can obtain performance improvement from the unlabeled data. Furthermore, SSCCM achieves better performance than hardSSCCM on both training and testing datasets, indicating that SSCCM can boost the classification performance by seeking the label memberships rather than the crisp labels, and the modified clustering assumption is really helpful.

TABLE III
TESTING PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART METHODS

Classifier	LapSVM	LapRLS	TSVM	Means3vm-iter	Means3vm-mkl	SSCCM	Consis. rate
Testing accuracy	0.87	0.865	0.855	0.87	0.87	0.89	1

TABLE IV
PERFORMANCES OF SSCCM, HARDSSCCM, LRLS (ON THE LABELED DATA ALONE), AND RLS (WITH ALL DATA LABELED)

Classifier	LRLS	hardSSCCM	SSCCM	RLS
Training acc.	0.86	0.915	0.935	0.95
Testing acc.	0.86	0.865	0.89	0.92

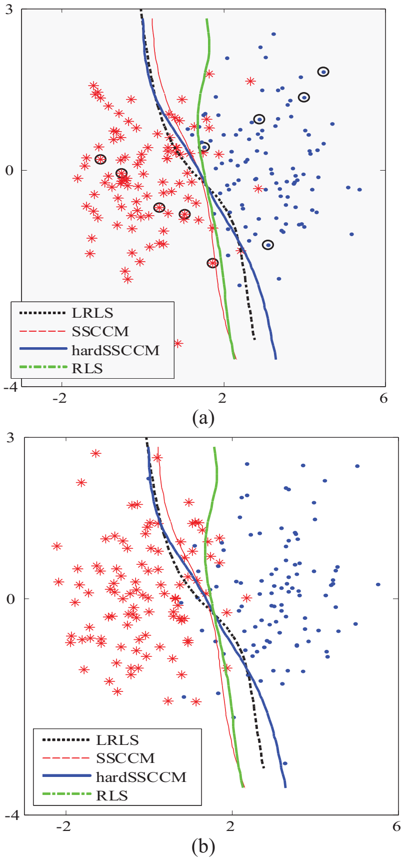


Fig. 3. Decision boundaries of SSCCM, hardSSCCM, LRLS, and RLS on (a) training and (b) testing datasets.

The superiority of SSCCM can also be observed from Fig. 3, which shows the decision boundaries of SSCCM, hardSSCCM, LRLS and RLS, respectively. In Fig. 3, the decision boundary of LRLS is determined by the labeled data alone. Moreover, since the boundary labeled instance in class 1 (its posterior probabilities to individual classes are 0.4988 and 0.5012 respectively) lies relatively at the bottom, and the boundary labeled instance in class 2 (its posterior probabilities to individual classes are 0.4720 and 0.5820, respectively) lies relatively at the top, the decision function from hardSSCCM is close to an "S" curve, that is, closer to class 1 at the top and closer to class 2 at the bottom, and thus provides poor classifications for the boundary instances. While the decision function from SSCCM is less influenced by those two "misleading" instances, and closer to that of RLS, which is close to a vertical line. More specifically, the cosine similarities between the normal vectors of decision boundaries derived from LRS and from LRLS, hardSSCCM and SSCCM are 0.6968, 0.8012, and 0.9098, respectively, thus SSCCM obtains a decision boundary closer to that of RLS, and naturally predicts more instances correctly.

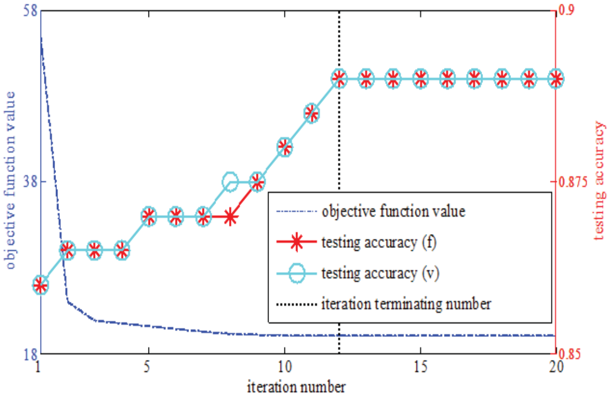


Fig. 4. Objective function values (left y-axis) and testing accuracies (right y-axis) of SSCCM by $f(x)$ and $v(x)$, respectively, in the first 20 iterative rounds, and the iteration terminates at the 12th round.

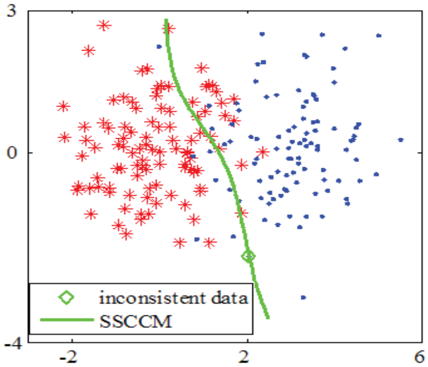


Fig. 5. Inconsistent instance and decision boundary of SSCCM in the 7th iterative step.

Fig. 4 shows the testing accuracies and objective function values of SSCCM in the first 20 iterative rounds, in which the objective function value decreases monotonically in the iterative count, and the iteration terminates within 12 rounds, demonstrating that SSCCM is convergent. On the other hand, the testing performances by $f(x)$ and $v(x)$, respectively, both tend to increase with the increase of iterative count, thus SSCCM can really gain performance promotion from the unlabeled data.

Furthermore, in the first 12 rounds, inconsistent instance appears only in the 7th step, thus Fig. 5 reveals such incon-

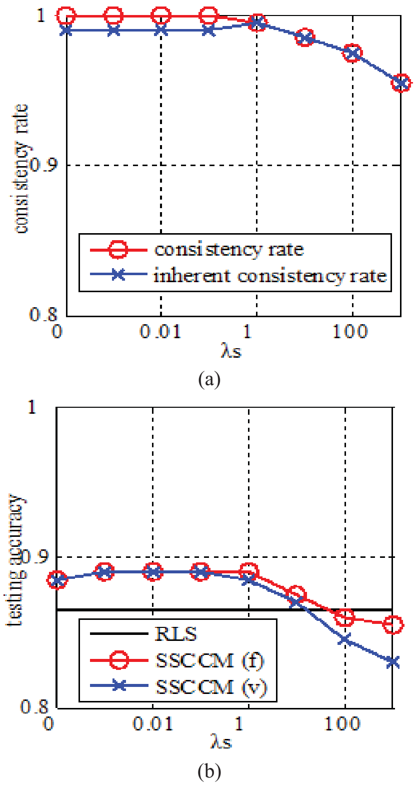


Fig. 6. (a) (Inherent) consistency rates and (b) accuracies of SSCCM by $f(x)$ and $v(x)$, respectively, with respect to different values of λ_s on the testing dataset.

sistent instance, along with the decision boundary of SSCCM in the 7th step. In Fig. 5, the inconsistent instance is exactly a difficultly classified boundary instance for the current decision hyper-plane, and the prediction for it is unreliable, which is in accordance with our analysis in Section IV-A.

Fig. 6(a) and (b) exhibits the testing performances and (inherently) consistency rates[3] of SSCCM with respect to different values of λ_s from {0, 0.001, 0.01, 0.1, 1, 10, 100, 1000}, respectively. In Fig. 6(a), the consistency rate of SSCCM can reach 1 when λ_s is small, but gradually decreases to the inherent-consistency rate as λ_s increases, attributing to that the pseudo-consistent instances become inconsistent ones with large λ_s. On the other hand, the inherent-consistency rate increases till λ_s reaches 1, and then decreases, because when λ_s is much smaller or larger than 1, SSCCM pays much more attention to the classification of given instances or their LWMs alone than their prediction consistency. At the same time, in Fig. 6(b), the performances obtained by $f(x)$ and $v(x)$, respectively, begin yielding difference when $\lambda_s \geq 0.1$, since in this case, their consistency rate becomes smaller than 1 in Fig. 6(a). On the other hand, when λ_s increases from 0 to 1000, both performances first ascend and then descend, which can be due to that the objective of SSCCM would be dominated by either the classification for given instances when λ_s is far smaller than 1, or the classification for their LWMs when λ_s is far larger than 1. As a result, the constraint that each instance and its LWM share the same label memberships is helpful for classification.

In summary, from such a toy illustration, we can observe that SSCCM can yield competitive performance compared to the state-of-the-art semi-supervised classification methods, as well as hardSSCCM, thereby, the modified cluster assumption is indeed helpful. Moreover, the class assignments for instances by the decision function and label membership function, respectively, are usually consistent, except for some boundary instances, for which the predictions are less reliable.

C. Experiments on Real Data

In this sub-section, SSCCM is compared with the state-of-the-art semi-supervised classification methods and hardSSCCM on both UCI and benchmark datasets, aiming to investigate the following issues.

1) How does SSCCM compare with the state-of-the-art semi-supervised classification methods?
2) How does SSCCM compare with hardSSCCM?
3) How does the regularization parameters λ_s affect the (inherent) consistency rate of SSCCM?
4) How does the regularization parameters λ_s affect the testing performance of SSCCM?

In what follows, we will detail the experimental setup, and the above aspects, respectively, in separate sub-sections.

1) Experimental Setup: For the UCI datasets, the experimental setups here follow those in [25] and [28]. Specifically, each dataset is randomly split into two halves, one for training and the other for testing, and the training set contains only ten labeled instances with the rest unlabeled. This process along with classifier learning is repeated 20 times and the average testing accuracies are reported. Linear kernel is adopted. For the compared methods, the regularization parameters C_1 and C_2 are fixed to 1 and 0.1, respectively, and the one-vs-all strategy is adopted for addressing multi-class problems. For SSCCM and hardSSCCM, λ and λ_s are fixed to 0.1 and 0.1, respectively, and k and ε are set to 5 and 10^{-3}, respectively. The results of the compared methods (excluding LapRLS) over the binary UCI datasets are taken from [25] and [28] (the partitions between the labeled and unlabeled instances follow [25], [28] over each dataset in each run).

For the benchmark datasets, the experimental setups follow those in [3] and [28]. Specifically, for each dataset, there are two settings, one including ten labeled instances and the other including 100 instances. Further, for each dataset and each setting, there are 12 subsets of labeled data and the average performances on unlabeled data are reported. The regularization parameters C_1 and C_2 in the compared methods are set to 100 and 0.1, respectively, and the parameters λ, λ_s, k, and ε in SSCCM and hardSSCCM are set to 1, 0.1, 5, and 10^{-3} respectively. Both linear and RBF kernels are used here, the width parameter in RBF kernel is set to the average distance between instances when ten instances are labeled, and selected via 10-fold cross-validation on labeled training data when

[3] Consistency rate indicates the ratio of instances with consistent predictions by the decision function and label membership function in the whole dataset, and inherent-consistency rate indicates the ratio of instances assigned the same prediction with its LWM by the decision function (such that its predictions by the decision function and label membership function are consistent) in the whole dataset.

TABLE V
TESTING PERFORMANCES OF SSCCM COMPARED WITH THE STATE-OF-THE-ART METHODS ON NINE BINARY UCI DATASETS

Algorithm	austra12	ethn12	Heart	House	isolet12	optdigits24	sat16	vehicle23	wdbc	Ave. Rank
LapSVM[13], [16]	0.7438(2)	**0.7460(1)**	0.7796(2)	0.8995(4)	0.9393(4)	0.9834(5)	0.9912(3)	0.7138(5)	**0.9107(1)**	3
LapRLS[13], [16]	**0.7568(1)**	0.7351(3)	**0.7811(1)**	0.8790(5)	0.9392(5)	0.9875(4)	0.9867(4)	0.7253(4)	0.8959(2)	3.2
TSVM[13], [16]	0.7338(3)	0.5469(6)	0.7763(3)	0.8655(6)	0.9038(6)	0.9234(6)	0.9826(5)	0.6362(6)	0.8640(3)	4.9
means3vm-*iter*[16]	0.6812(5)	0.7321(4)	0.7456(4)	0.9172(3)	0.9875(3)	0.9893(3)	**0.9956(1)**	0.8247(2)	0.7939(6)	3.4
means3vm-*mkl*[16]	0.6759(6)	0.7357(2)	0.7322(5)	0.9190(2)	0.9898(2)	0.9909(2)	**0.9956(1)**	0.8215(3)	0.8019(5)	3.1
SSCCM	0.7249(4)	0.7153(5)	0.7072(6)	**0.9258(1)**	**0.9967(1)**	**0.9934(1)**	0.9941(2)	**0.8358(1)**	0.8044(4)	**2.8**
Consis. rate	0.9977	1	0.9901	1	0.9993	0.9993	0.9991	0.9922	0.9979	/

TABLE VI
TESTING PERFORMANCES OF SSCCM COMPARED WITH THE STATE-OF-THE-ART METHODS ON NINE MULTI-CLASS UCI DATASETS

Algorithm	Balan.	CMC	Dermatology	Glass	Lungcancer	Soybean	TAE	Vehicle	Wine	A. ra.
LapSVM	0.7843(2)	0.6407(3)	0.9237(3)	0.8689(2)	0.6856(4)	**0.9013(1)**	0.5831(5)	0.6997(4)	0.8448(5)	3.2
LapRLS	**0.7896(1)**	0.6312(4)	0.9142(4)	**0.877(1)**	0.6822(5)	0.8980(2)	0.5964(4)	0.6964(5)	0.8456(4)	3.3
TSVM	0.7502(6)	0.5861(6)	0.9057(5)	0.8532(5)	0.6867(3)	0.8493(6)	0.5702(6)	0.6743(6)	0.8431(6)	5.4
Means3vm-*iter*	0.7643(5)	**0.6443(1)**	0.9367(2)	0.8605(4)	**0.6911(1)**	0.8804(5)	0.5982(3)	0.7123(2)	0.8523(3)	2.9
Means3vm-*mkl*	0.7649(4)	0.6430(2)	0.9237(3)	0.8416(6)	**0.6911(1)**	0.8942(4)	0.6004(2)	**0.7183(1)**	0.8494(2)	2.8
SSCCM	0.7703(3)	0.6293(5)	**0.9448(1)**	0.8673(3)	0.6900(2)	0.8970(3)	**0.6102(1)**	0.7009(3)	**0.8615(1)**	**2.4**
Consis. rate	0.996	0.9987	0.9988	0.9979	0.9867	0.9989	0.996	0.9982	0.9924	/

100 instances are labeled, finally the better result between those two kernels on each dataset is reported. The results of the compared methods are taken from [3] and [28] (the partitions between the labeled and unlabeled instances follow [3], [28] over each dataset in each run).

2) Comparison With the State-of-the-Art Semi-Supervised Classification Methods: SSCCM is compared with the state-of-the-art methods over nine binary UCI datasets, and the comparison results are listed in Table V. Each row (except the last one) corresponds to the testing accuracies (with performance-rank in the bracket) of each method on individual datasets, the last row gives the consistency rates of SSCCM on individual datasets, and the last column shows the average performance-rank of individual methods on all datasets. The bold value in each column indicates the best performance or average-performance-rank among all compared methods. From Table V, it is clear that SSCCM outperforms the other methods on four out of the nine datasets and its average-performance-rank is the best. In addition, its consistency rates are all close to 1, with the smallest to 0.9901 on the heart dataset.

SSCCM is compared with those methods on nine multi-class UCI datasets and the results are shown in Table VI, which has the same structure with Table V. From Table VI, we can find that SSCCM outperforms the other methods on three datasets and achieves the best average-performance-rank as well. Likewise, the consistency rates of SSCCM are all close to 1, with the smallest to 0.9867 on the *lungcancer* dataset.

SSCCM is also compared with those methods on six benchmark datasets with the results shown in Table VII, in which the upper-half and lower-half parts correspond to results with 10 and 100 labeled instances, respectively. In individual parts, each row (except the last one) gives the performances of each method (with performance-rank in the bracket) on individual datasets, the last row shows the consistency rates of SSCCM on individual datasets, and the last column shows the average performance-rank of individual methods on all datasets. The bold value in each column indicates the method achieving the best testing accuracy or average performance-rank. From Table VII, it is clear that when there are 10 labeled instances, SSCCM achieves the best performances on two out of the six datasets and the best average performance-rank, the consistency rates are all close to 1, with the smallest to 0.9932 on the $G241c$ dataset. At the same time, when there are 100 labeled instances, SSCCM achieves the best performances on two datasets and the best average performance-rank as well. Likewise, the consistency rates are all close to 1, with the smallest to 0.9901 on the $G241d$ dataset.

As a result, SSCCM can provide encouraging performance for semi-supervised classification compared with the state-of-the-art methods, and the decision function and the label membership function usually provide consistent data predictions.

3) Comparison With HardSSCCM: The comparison results between SSCCM and hardSSCCM on binary UCI datasets are shown in Table VIII, in which each row corresponds to the testing performances of each method on individual datasets, and the last column shows the average performance of each method on all datasets. From Table VIII, it can be observed that SSCCM performs better than hardSSCCM on seven datasets, and its average performance is better than that of hardSSCCM.

The comparison results on multi-class UCI datasets are shown in Table IX, which has the same structure with Table VIII. From Table IX, it can also be observed that SSCCM achieves better performances on seven datasets, and the better average performance as well.

Moreover, the comparison results on the benchmark datasets are shown in Table X, in which the upper-half and lower-

TABLE VII
PERFORMANCES OF SSCCM AND THE STATE-OF-ART METHODS ON SIX BENCHMARK DATASETS WITH 10 AND 100 LABELED INSTANCES, RESPECTIVELY

No. of Label	Algorithm	G241c	G241d	Digit1	USPS	BCI	TEXT	Average Rank
10	LapSVM[1], [16]	**0**.5379(6)	0.5485(4)	0.9103(2)	**0.8095(1)**	0.5075(6)	0.6272(6)	4.2
	LapRLS[1], [16]	0.5605(4)	0.5432(5)	**0.9456(1)**	0.8101(2)	0.5103(4)	0.6632(5)	3.5
	TSVM[1], [16]	**0.7529(1)**	0.4992(6)	0.8223(5)	0.7480(6)	0.5085(5)	0.6879(3)	4.3
	means3vm-iter[16]	0.7222(2)	0.5700(2)	0.8298(4)	0.7634(5)	0.5188(3)	0.6957(2)	3
	means3vm-mkl[16]	0.6548(5)	**0.5894(1)**	0.8300(3)	0.7784(4)	0.5207(2)	0.6691(4)	3.2
	SSCCM	0.6699(3)	0.5691(3)	0.7917(6)	0.8088(3)	**0.5404(1)**	**0.7105(1)**	**2.8**
	Consistency rate	0.9932	0.9974	0.9978	0.9978	0.9979	0.9955	/
100	LapSVM[1], [16]	0.7618(5)	0.7364(5)	0.9687(2)	0.9530(2)	0.6761(5)	0.7614(5)	4
	LapRLS[1], [16]	0.7564(6)	0.7354(6)	**0.9708(1)**	**0.9532(1)**	0.6864(4)	0.7643(4)	3.7
	TSVM[1], [16]	**0.8154(1)**	0.7758(2)	0.9385(6)	0.9023(6)	0.6675(6)	0.7548(6)	4.5
	means3vm-iter[16]	0.8000(3)	0.7752(3)	0.9568(5)	0.9383(4)	0.7131(3)	0.7674(2)	3.3
	means3vm-mkl[16]	0.8025(2)	**0.7758(1)**	0.9591(4)	0.9317(5)	0.7144(2)	0.7660(3)	2.8
	SSCCM	0.7902(4)	0.7484(4)	0.9682(3)	0.9445(3)	**0.7326(1)**	**0.7832(1)**	**2.7**
	Consistency rate	0.9978	0.9901	1	0.9996	0.9905	0.9979	/

TABLE VIII
PERFORMANCES OF SSCCM COMPARED WITH HARDSSCCM ON NINE BINARY UCI DATASETS

Algorithm	Austra12	Ethn12	Heart	House	Isolet12	Optdigits24	Sat16	Vehicle23	Wdbc	Mean
HardSSCCM	0.7026	0.6642	**0.7141**	0.8960	0.9904	0.9783	0.9902	0.8264	**0.8423**	0.8449
SSCCM	**0.7249**	**0.7153**	0.7072	**0.9258**	**0.9967**	**0.9934**	**0.9941**	**0.8358**	0.8044	**0.8553**

TABLE IX
PERFORMANCES OF SSCCM COMPARED WITH HARDSSCCM ON NINE MULTI-CLASS UCI DATASETS

Algorithm	Balance	CMC	Dermatology	Glass	Lungcancer	Soybean	TAE	Vehicle	Wine	Mean
HardSSCCM	0.7533	0.5994	0.9169	0.8625	**0.6989**	0.8604	0.5857	**0.7094**	0.8450	0.7591
SSCCM	**0.7703**	**0.6293**	**0.9448**	**0.8673**	0.6900	**0.8970**	**0.6102**	0.7009	**0.8615**	**0.7745**

TABLE X
PERFORMANCES OF SSCCM COMPARED WITH HARDSSCCM ON SIX BENCHMARK DATASETS WITH 10 AND 100 LABEL INSTANCES, RESPECTIVELY

No. of Label	Algorithm	G241c	G241d	Digit1	USPS	BCI	Text	Mean
10	HardSSCCM	0.6558	0.5473	0.7725	0.8094	0.5205	0.6932	0.6665
	SSCCM	**0.6699**	**0.5691**	**0.7917**	0.8088	**0.5404**	**0.7105**	**0.6817**
100	HardSSCCM	0.7689	0.7562	0.9505	0.9235	0.7202	0.7746	0.8156
	SSCCM	**0.7902**	0.7484	**0.9682**	**0.9445**	**0.7326**	**0.7832**	**0.8278**

half parts correspond to the results with 10 and 100 labeled instances, respectively. In individual parts, each row gives the performances of each method on individual datasets, and the last column shows the average performance of individual methods on all datasets. From Table X, it is clear that when 10 instances are labeled, SSCCM achieves better performances on five datasets, and the better average performance. Similarly, when 100 instances are labeled, SSCCM also achieves better performances on five datasets, and the better average performance.

As a result, through adopting the modified cluster assumption, SSCCM achieves better performance than hardSSCCM on most datasets, indicating that the modified cluster assumption is effective for semi-supervised classification. However, there are also datasets on which SSCCM performs no better than hardSSCCM. One possible reason is the iterative optimization strategy along with its termination strategy. The iterative optimization strategy only results in a local solution, and moreover, the iteration process may terminate without achieving the "best" performance. Hence, designing better termination strategy for iteration and optimization strategy for SSCCM are both important works in the future.

4) Consistency With Respect to Different Values of λ_s: Fig. 7 reveals the (inherent) consistent rates of SSCCM with respect to different values of λ_s from {0, 0.001, 0.01, 0.1, 1, 10, 100, 1000} on six UCI datasets. In Fig. 7, when λ_s is small enough, the consistency rate can reach 1, then gradually decreases with the increase of λ_s and finally becomes equal to the inherent-consistency rate, since now the pseudo-consistent instances become inconsistent ones. At the same time, the inherent-consistency rate increases till λ_s reaches one and then decreases, the reason can be that when λ_s is far smaller or

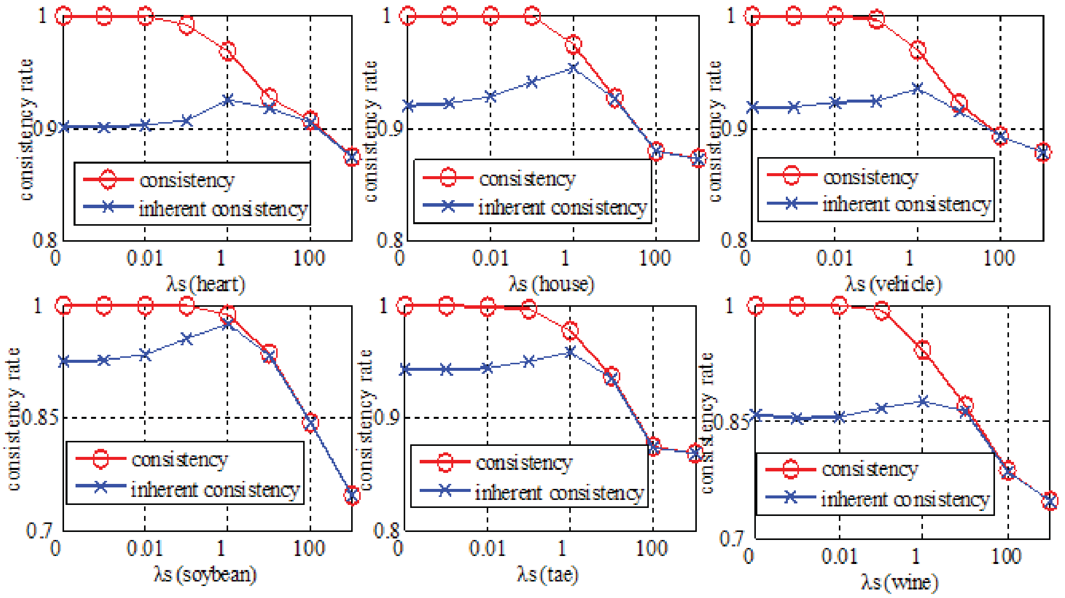


Fig. 7. Consistency and inherent-consistency rates between data predictions by $f(x)$ and $\upsilon(x)$, respectively, corresponding to different values of λ_s on six UCI datasets.

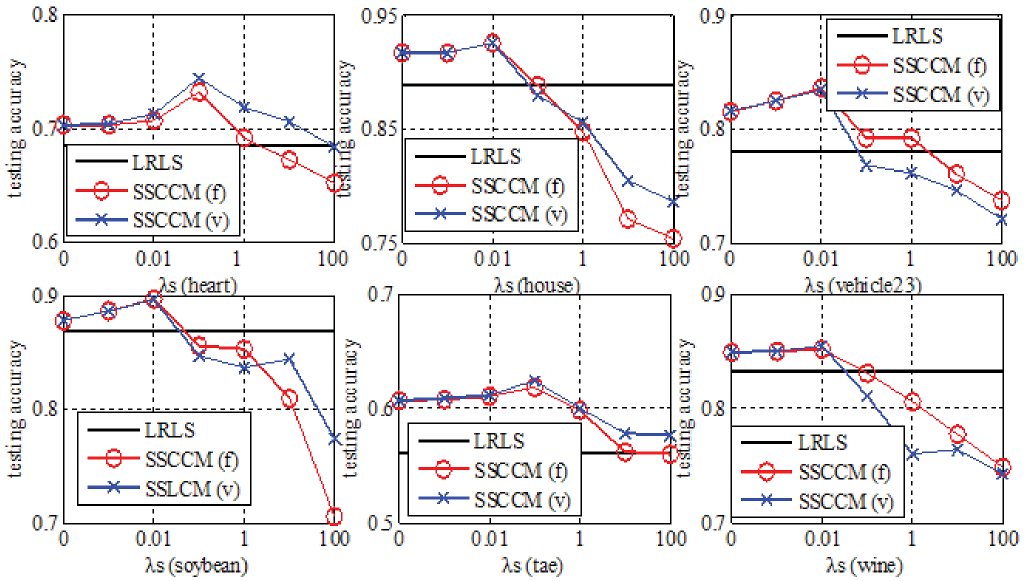


Fig. 8. Testing accuracies of SSCCM by $f(x)$ and $\upsilon(x)$, respectively, corresponding to different values of λ_s on six UCI datasets, with the performances of the LRLS classifier as the base line.

larger than one, the objective of SSCCM would focus much more on the classification for data or their LWMs alone than their prediction consistency.

5) Testing Accuracy With Respect to Different Values of λ_s: Fig. 8 exhibits the testing accuracies of SSCCM with respect to different values of λ_s from {0, 0.001, 0.01, 0.1, 1, 10, 100, 1000} on the six UCI datasets. Moreover, the performances by LRLS and the label membership function in SSCCM are also shown, respectively.

In Fig. 8, when λ_s is small, the performances obtained by $f(x)$ and $\upsilon(x)$, respectively, are nearly the same, attributing to that their predictions are consistent for most data. When λ_s becomes large, their performances become different, since the consistency rate decreases with the increase of λ_s. On the other hand, as λ_s increases from 0 to 1000, both performances first ascend and then descend, the reason can be that when λ_s is much smaller or larger than 1, the objective of SSCCM pays much more attention to the classification for data or their LWMs alone than their prediction consistency. As a result, the constraint that each instance and its LWM share the same label memberships is reasonable and can help boost the classification performance.

VI. Conclusion

In this paper, we present a modified cluster assumption for semi-supervised classification, and further develop a new classification method SSCCM based on it. SSCCM seeks the decision function and label membership function simultaneously, and provides a unified objective function for the labeled, unlabeled data and their derived LWMs. Its optimization follows an iterative procedure, and the iteration is provably convergent. It returns the decision function as well as the label membership function, whose prediction results can verify each other, and the reliability of semi-supervised learning might be enhanced by checking their prediction consistency. Experiments on both toy and real datasets validate the competitiveness of SSCCM compared with the state-of-the-art semi-supervised classification methods and hardSSCCM, consequently validate the effectiveness of the modified cluster assumption.

In the future, there are still several directions worthy to study.

1) *Optimization of SSCCM:* The iteration process of SSCCM may terminate without achieving the "best" performance, thus some heuristic termination strategy for iteration is needed. At the same time, an iteration optimization strategy for SSCCM only results in a local solution, thereby some optimization strategy for obtaining a global solution is worth studying.
2) *More Delicate Cluster Structure:* In both cluster and modified cluster assumption, the cluster number for the unlabeled instances is implicitly assumed to be the given class number, while we can also consider a cluster number larger than the class number, that is, each class corresponds to multiple clusters, and develop new semi-supervised classification methods based on it.
3) *Value of b:* Though the value of b is simply fixed to 2 in this paper, it can be selected via some heuristic strategies, optimized in the learning phase, or even changed in the learning process (e.g., descends with a large initial value to weaken the influence of the initial values).
4) *Additional Prior Information:* We can incorporate other data geometrical information, e.g., the manifold information into SSCCM through additional regularizations on either the decision function or the label membership function. We can also impose the entropy regularization on the label memberships to guarantee uncertainty [40].

Appendix

A. From the Representer Theorem, $f(x) = \sum_{i=1}^{n} \alpha_i K(x_i, x)$, then (6) can be translated into

$$\begin{aligned}\min_{\alpha}\ & M_1 = tr\left((\alpha K_l - Y)(\alpha K_l - Y)^T\right)\\ &+\lambda_s tr\left((\alpha \bar{K}_l - Y)(\alpha \bar{K}_l - Y)^T\right)\\ &+tr\left((\alpha K_u J - L)\hat{V}(\alpha K_u J - L)^T\right)\\ &+\lambda_s tr\left((\alpha \bar{K}_u J - L)\hat{V}(\alpha \bar{K}_u J - L)^T\right) + \lambda tr(\alpha K \alpha^T)\end{aligned} \tag{13}$$

where $\alpha = [\alpha_1, \alpha_2 \ldots, \alpha_n] \in R^{C\times n}$ is the Lagrange multiplier matrix. $K = [K_l\ K_u] = \begin{bmatrix} K_{ll} & K_{lu} \\ K_{lu}^T & K_{uu} \end{bmatrix}$ where $K_{ll} = \langle\phi(X_l),\ \phi(X_l)\rangle_{\mathcal{H}}$, $K_{lu} = \langle\phi(X_l),\ \phi(X_u)\rangle_{\mathcal{H}}$, and $K_{uu} = \langle\phi(X_u),\ \phi(X_u)\rangle_{\mathcal{H}}$. $\bar{K} = [\bar{K}_l\ \bar{K}_u] = \begin{bmatrix} \bar{K}_{ll} & \bar{K}_{lu} \\ \bar{K}_{ul} & \bar{K}_{uu} \end{bmatrix}$, where $\bar{K}_{ll} = \langle\phi(X_l),\ \widehat{\phi(X_l)}\rangle_{\mathcal{H}}$, $\bar{K}_{lu} = \langle\phi(X_l),\ \widehat{\phi(X_u)}\rangle_{\mathcal{H}}$, $\bar{K}_{ul} = \langle\phi(X_u),\ \widehat{\phi(X_l)}\rangle_{\mathcal{H}}$, and $\bar{K}_{uu} = \langle\phi(X_u),\ \widehat{\phi(X_u)}\rangle_{\mathcal{H}}$. Each entry in $\bar{K}$ can be written as

$$\begin{aligned}\bar{K}_{ij} &= \left\langle\phi(x_i), \widehat{\phi(X_j)}\right\rangle_{\mathcal{H}} = \left\langle \phi(x_i), \frac{\sum_{x_s\in Ne(x_j)} S_{sj}\phi(x_s)}{\sum_{x_s\in Ne(x_j)} S_{sj}} \right\rangle_{\mathcal{H}}\\ &= \frac{\sum_{x_s\in Ne(x_j)} S_{sj}\ \langle\phi(x_i), \phi(x_s)\rangle_{\mathcal{H}}}{\sum_{x_s\in Ne(x_j)} S_{sj}}\\ &= \frac{\sum_{x_s\in Ne(x_j)} S_{sj} K_{is}}{\sum_{x_s\in Ne(x_j)} S_{sj}}.\end{aligned} \tag{14}$$

$J = [\underbrace{I_u \ldots I_u}_{C}] \in R^{n_u\times(C\times n_u)}$, where I_u is a $n_u \times n_u$ identity matrix, $L = [L_1 \ldots L_C] \in R^{C\times(C\times n_u)}$, where each L_k is a $C \times n_u$ matrix with the kth row being an all-one vector and the rest being all-zero vectors. Let $V = [v(x_1) \ldots v(x_{n_u})] \in R^{C\times n_u}$ denote the label membership values for the unlabeled data, then $\hat{V}$ denotes a diagonal matrix with the diagonal elements being the squared values of the entries in V arranged by rows.

Set the derivative of M_1 w.r.t. α to zero

$$\begin{aligned}\partial M_1/\partial\alpha &= (\alpha K_l - Y)K_l^T + (\alpha K_u J - L)\hat{V} J^T K_u^T\\ &\quad +\lambda_s(\alpha \bar{K}_l - Y)\bar{K}_l^T + \lambda_s(\alpha \bar{K}_u J - L)\\ &\quad \hat{V} J^T \bar{K}_u^T + \lambda\alpha K\\ &= 0\end{aligned} \tag{15}$$

leading to the following solution:

$$\begin{aligned}\alpha &= (Y K_l^T + L\hat{V} J^T K_u^T + \lambda_s Y \bar{K}_l^T + \lambda_s L \hat{V} J^T \bar{K}_u^T)\\ &\quad (K_l K_l^T + K_u J \hat{V} J^T K_u^T + \lambda_s \bar{K}_l \bar{K}_l^T + \lambda_s \bar{K}_u J \hat{V} J^T \bar{K}_u^T\\ &\quad +\lambda K)^{-1}.\end{aligned} \tag{16}$$

B. Using the Lagrange multiplier method, we have

$$\begin{aligned}M_2 = \sum_{k=1}^{C}\sum_{j=n_l+1}^{n} & v_k(x_j)^2\\ &\left(\left\|f(x_j) - r_k\right\|^2 + \lambda_s\left\|f(\hat{x}_j) - r_k\right\|^2\right)\\ &- \sum_{j=n_l+1}^{n}\lambda_j\left(\sum_{k=1}^{C} v_k(x_j) - 1\right).\end{aligned} \tag{17}$$

Likewise, the derivative of M_2 w.r.t. each $v_k(x_j)$ vanishes at the minimizer, $\forall k = 1, \ldots, C,\ \ j = n_l + 1, \ldots, n$

$$\begin{aligned}\frac{\partial M_2}{\partial v_k(x_j)} &= 2\left(\left\|f(x_j) - r_k\right\|^2 + \lambda_s\left\|f(\hat{x}_j) - r_k\right\|^2\right)\\ &\quad v_k(x_j) - \lambda_j = 0\end{aligned} \tag{18}$$

thus

$$v_k(x_j) = \frac{\lambda_j}{2}\left(\left\|f(x_j) - r_k\right\|^2 + \lambda_s\left\|f(\hat{x}_j) - r_k\right\|^2\right). \tag{19}$$

Further, from the constraint $\sum_{k=1}^{C} v_k(x_j) = 1$, we have

$$v_k(x_j) = \frac{1/\left(\left\|f(x_j) - r_k\right\|^2 + \lambda_s\left\|f(\hat{x}_j) - r_k\right\|^2\right)}{\sum_{k=1}^{C} 1/\left(\left\|f(x_j) - r_k\right\|^2 + \lambda_s\left\|f(\hat{x}_j) - r_k\right\|^2\right)}. \tag{20}$$

ACKNOWLEDGMENT

The authors would like to thank the three anonymous referees and the editor for their helpful comments and suggestions.

REFERENCES

[1] Z. Xu, I. King, M. R. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.

[2] E. Hu, S. Chen, D. Zhang, and X. Yin, "Semisupervised kernel matrix learning by kernel propagation," *IEEE Trans. Neural Netw.*, vol. 21, no. 11, pp. 1831–1841, Nov. 2010.

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

[4] X. Zhu and A. B. Goldberg, *Introduction to Semi-Supervised Learning*. San Rafael, Argentina: Morgan & Claypool, 2009.

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

[6] X. Zhu, "Semi-supervised learning literature survey," Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, Tech. Rep. 1530, Jul. 2008.

[7] H. Xue, S. Chen, and Q. Yang, "Structural regularized support vector machine: A framework for structural large margin classifier," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 573–587, Apr. 2011.

[8] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[9] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 103–134, 2000.

[10] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[11] N. V. Chawla and G. Karakoulas, "Learning from labeled and unlabeled data: An empirical study across techniques and domains," *J. Artif. Intell. Res.*, vol. 23, no. 1, pp. 331–366, Jan. 2005.

[12] F. G. Cozman, I. Cohen, and M. C. Cirelo, "Semi-supervised learning of mixture models," in *Proc. 20th Int. Conf. Mach. Learn.*, Washington D.C., 2003, pp. 99–106.

[13] S. Ben-David, T. Lu, and D. Pál, "Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning," in *Proc. 21st Annu. Conf. Learn. Theory*, Helsinki, Finland, 2008, pp. 33–44.

[14] A. Singh, R. D. Nowak, and X. Zhu, "Unlabeled data: Now it helps, now it doesn't," in *Proc. Adv. Neural Inf. Process. Syst. 22*, Vancouver, BC, Canada, 2008, pp. 1513–1520.

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

[16] Y.-F. Li and Z.-H. Zhou, "Improving semi-supervised support vector machines through unlabeled instances selection," in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, 2011, pp. 386–391.

[17] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Comput.*, vol. 4, no. 6, pp. 888–900, Nov. 1992.

[18] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *Artif. Intell. Rev.*, vol. 11, nos. 1–5, pp. 11–73, Feb. 1997.

[19] H. Xue and S. Chen, "Alternative robust local embedding," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, 2007, pp. 591–596.

[20] A. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke, "Partially supervised clustering for image segmentation," *Pattern Recognit.*, vol. 29, no. 5, pp. 895–871, May 1996.

[21] W. Pedrycz and J. Waletzky, "Fuzzy clustering with partial supervision," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 27, no. 5, pp. 787–795, Sep. 1997.

[22] H. Zhang and J. Lu, "Semi-supervised fuzzy clustering: A kernel-based approach," *Knowl.-Based Syst.*, vol. 22, no. 6, pp. 477–481, Aug. 2009.

[23] N. Grira, M. Crucianu, and N. Boujemaa, "Unsupervised and semi-supervised clustering: A brief survey," in *Proc. Rev. Mach. Learn. Tech. Process. MUSCLE Eur. Netw. Excell.*, 2004, pp. 1–12.

[24] W. Cai, S. Chen, and D. Zhang, "A multiobjective simultaneous learning framework for clustering and classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 185–200, Feb. 2010.

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

[26] K. P. Bennett and A. Demiriz, "Semi-supervised support vector machines," in *Proc. Adv. Neural Inf. Process. Syst. 11*, 1998, pp. 368–374.

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

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

[29] O. Chapelle, V. Sindhwani, and S. Keerthi, "Optimization techniques for semi-supervised support vector machines," *J. Mach. Learn. Res.*, vol. 9, pp. 203–233, Jun. 2008.

[30] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. 10th Int. Workshop Artif. Intell. Stat.*, 2005, pp. 57–64.

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

[32] Z. Xu, R. Jin, J. Zhu, I. King, and M. Lyu, "Efficient convex relaxation for transductive support vector machine," in *Proc. Adv. Neural Inf. Process. Syst. 21*, 2008, pp. 1641–1648.

[33] V. Sindhwani, S. S. Keerthi, and O. Chapelle, "Deterministic annealing for semi-supervised kernel machines," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 841–848.

[34] O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Branch and bound for semi-supervised support vector machines," in *Proc. Adv. Neural Inf. Process. Syst. 20*, 2007, pp. 217–224.

[35] V. Vural, G. Fung, J. Dy, and B. Rao, "Semi-supervised classifiers using a-priori metric information," *Optim. Methods Softw. J.*, vol. 23, no. 4, pp. 521–532, 2006.

[36] 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, pp. 2399–2434, Nov. 2006.

[37] K. Nigam, A. K. McCallum, and S. Thrun, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2–3, pp. 103–134, 2000.

[38] 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.

[39] A. Frank and A. Asuncion. (2010). *UCI Machine Learning Repository*. School Inf. Comput. Sci., Univ. California, Irvine, CA [Online]. Available: http://archive.ics.uci.edu/ml

[40] L. Zhang, L. Qiao, and S. Chen, "Graph-optimized locality preserving projections," *Pattern Recognit.*, vol. 43, no. 6, pp. 1993–2002, Jun. 2010.

Yunyun Wang received the B.S. degree in computer science and technology from Anhui Normal University, Wuhu, China, in 2006. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China.

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 merged into Zhejiang University), Zhejiang, China, the M.S. degree in computer applications from Shanghai Jiaotong University, Shanghai, China, and the Ph.D. degree in communication and information systems from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1983, 1985, and 1997, respectively.

He was at NUAA in January 1986. Since 1998, he has been a full-time Professor with the Department of Computer Science and 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.

Zhi-Hua Zhou (S'00–M'01–SM'06) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Nanjing University, Nanjing, China, in 1996, 1998, and 2000, respectively, all with the highest honors.

He joined the Department of Computer Science and Technology, Nanjing University, as an Assistant Professor in 2001, and is currently Cheung Kong Professor and Director of the LAMDA Group. He has published over 70 papers in leading international journals or conference proceedings. His current research interests include artificial intelligence, machine learning, data mining, pattern recognition, information retrieval, evolutionary computation, and neural computation.

Dr. Zhou has won various awards/honors including the National Science and Technology Award for Young Scholars of China in 2006, the Award of National Science Fund for Distinguished Young Scholars of China in 2003, the National Excellent Doctoral Dissertation Award of China in 2003, and the Microsoft Young Professorship Award in 2006. He is an Associate Editor of the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. He is the Associate Editor-in-Chief of the Chinese Science Bulletin and the editorial boards of the Artificial Intelligence in Medicine, Intelligent Data Analysis, Science in China. He is the founder of the Asian Conference on Machine Learning (ACML) and Steering Committee Member of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD) and the Pacific Rim International Conference on Artificial Intelligence (PRICAI). He has served as a Program Committee Chair/Co-Chair of PAKDD in 2007, PRICAI in 2008, and ACML in 2009, Vice Chair or Area Chair of conferences including IEEE ICDM in 2006 and 2008, SIAM DM in 2009, and ACM CIKM in 2009. He was the General Chair/Co-Chair or Program Committee Chair/Co-Chair of 12 native conferences. He is the Chair of the Machine Learning Society of the Chinese Association of Artificial Intelligence, Vice Chair of the Artificial Intelligence and Pattern Recognition Society of the China Computer Federation, and the Chair of the IEEE Computer Society Nanjing Chapter. He is a member of the Australian Association of Architectural Illustrators and ACM, the IEEE Computer Society, and the IEEE Computational Intelligence Society.