Information Sciences 422 (2018) 51–76

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

SSC-EKE: Semi-supervised classification with extensive knowledge exploitation

Pengjiang Qian[a,b,c,*], Chen Xi[a,b,c], Min Xu[a], Yizhang Jiang[a], Kuan-Hao Su[b,c], Shitong Wang[a], Raymond F. Muzic Jr.[b,c]

[a] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, PR China
[b] Case Center for Imaging Research, Case Western Reserve University, Cleveland, Ohio, USA
[c] Department of Radiology, University Hospitals Cleveland Medical Center, Case Western Reserve University, Cleveland, Ohio, USA

ARTICLE INFO

Article history:
Received 9 March 2016
Revised 9 May 2017
Accepted 31 August 2017
Available online 21 September 2017

Keywords:
Support vector machine (SVM)
Semi-supervised classification
Manifold learning
Graph Laplacian
Knowledge
Reproducing kernel Hilbert space (RKHS)

ABSTRACT

We introduce a new, semi-supervised classification method that extensively exploits knowledge. The method has three steps. First, the manifold regularization mechanism, adapted from the Laplacian support vector machine (LapSVM), is adopted to mine the manifold structure embedded in all training data, especially in numerous label-unknown data. Meanwhile, by converting the labels into pairwise constraints, the pairwise constraint regularization formula (PCRF) is designed to compensate for the few but valuable labelled data. Second, by further combining the PCRF with the manifold regularization, the precise manifold and pairwise constraint jointly regularized formula (MPCJRF) is achieved. Third, by incorporating the MPCJRF into the framework of the conventional SVM, our approach, referred to as *semi-supervised classification with extensive knowledge exploitation* (SSC-EKE), is developed. The significance of our research is fourfold: 1) The MPCJRF is an underlying adjustment, with respect to the pairwise constraints, to the graph Laplacian enlisted for approximating the potential data manifold. This type of adjustment plays the correction role, as an unbiased estimation of the data manifold is difficult to obtain, whereas the pairwise constraints, converted from the given labels, have an overall high confidence level. 2) By transforming the values of the two terms in the MPCJRF such that they have the same range, with a trade-off factor varying within the invariant interval [0, 1), the appropriate impact of the pairwise constraints to the graph Laplacian can be self-adaptively determined. 3) The implication regarding extensive knowledge exploitation is embodied in SSC-EKE. That is, the labelled examples are used not only to control the empirical risk but also to constitute the MPCJRF. Moreover, all data, both labelled and unlabelled, are recruited for the model smoothness and manifold regularization. 4) The complete framework of SSC-EKE organically incorporates multiple theories, such as joint manifold and pairwise constraint-based regularization, smoothness in the reproducing kernel Hilbert space, empirical risk minimization, and spectral methods, which facilitates the preferable classification accuracy as well as the generalizability of SSC-EKE.

* Corresponding author at: School of Digital Media, Jiangnan University, Wuxi, Jiangsu, PR China.
E-mail addresses: qianpjiang@jiangnan.edu.cn, qianpjiang@126.com (P. Qian).

http://dx.doi.org/10.1016/j.ins.2017.08.093
0020-0255/

1. Introduction

Classification, which aims at identifying the data instances in a testing set by using the discriminant function learned from a training set, is an important branch of pattern recognition. The effectiveness of conventional classification approaches, such as support vector machines (SVMs) [6,14,20,28,36–38] and artificial neural networks [11,12,24,25,29], is largely dependent on the quantity and quality of training examples. Most of these approaches can obtain satisfactory results only in ideal situations in which the information embedded in the training set is sufficient so that the learned classifiers are insightful. Obtaining informative training examples is sometimes computationally expensive or labour-intensive. Instead, we often have a limited amount of labelled data but many unlabelled instances. In response to this challenge, semi-supervised classification techniques [3,4,13,15–18,20,30,39–41,43,44] have been developed to simultaneously exploit the prior knowledge existing in numerous, label-unknown examples as well as a small quantity of label-known ones in the training set to improve the accuracy and generalizability of the classifier on target data sets. In such cases, the semi-supervised SVMs (S3VMs), which are derivatives of the classic SVM, have motivated extensive research, and quite a few approaches have been reported. Representative work can be briefly reviewed as follows. The transductive support vector machine (TSVM) [13], as one of the pioneering S3VMs, was initially developed for text classification. By taking the transductive rather than any inductive strategy, the TSVM takes into account a particular testing set, i.e., the testing set is used as an additional source of information regarding hyperplane margins, and attempts to minimize the misclassification of only those particular examples in the testing set. The 1-norm linear, SVM-based, semi-supervised model [4] constructs a general SVM model minimizing both the misclassification error and the function capacity by using all of the available data from both the training and working (namely, testing) sets, in which the 1-norm linear SVM is converted to a mixed-integer program (MIP) and then exactly solved using integer programming. Due to the observation that the semi-supervised SVM with known label means of unlabelled data is closely related to the supervised SVM that has access to all the labels of the unlabelled examples, two versions of the MeanS3VM [15], i.e., MeanS3VM-*mkl* and MeanS3VM-*iter*, were separately proposed by maximizing the margin between the label means of the unlabelled data. The former is based on multiple-kernel learning, whereas the latter is based on alternating optimization. The cost-sensitive semi-supervised SVM (CS4VM) [16] simultaneously considers unequal misclassification costs and the utilization of unlabelled data. This is a cost-sensitive extension of the MeanS3VM and likewise is able to closely approximate the supervised, cost-sensitive SVM that has access to the ground-truth labels of all the unlabelled data when given the label means of the unlabelled data. The weakly labelled SVM (WeLlSVM) [17] studies the problem of learning using weakly labelled data where labels of the training examples are incomplete. This includes, e.g., 1) semi-supervised learning where labels are partially known; 2) multi-instance learning where labels are implicitly known; and 3) clustering where labels are completely unknown. Via a convex relaxation of the original MIP, the WeLlSVM is solved by using a sequence of SVM subproblems that are much more scalable than convex, semi-definite programming relaxations. As such, the WeLlSVM obtains improved performance and practicability when facing large data sets. In addition, the Laplacian SVM (LapSVM) [3] and Laplacian-regularized least squares (LapRLS) [3], which benefit from the manifold regularization in which the geometry of the marginal distribution with respect to both labelled and unlabelled data in the training set is used, feature better classification accuracy than the classic SVM approach.

Among these existing S3VMs mentioned above, the LapSVM has particularly captured our interest due to its manifold regularization mechanism, which relies primarily on unlabelled data. There are three terms in the framework of the LapSVM. Specifically, the first controls the empirical risk by using the given labelled examples. The other two regularization terms, respectively, impose the smoothness condition on the possible solutions and the geometric knowledge of the probability distribution. However, it is clear that the few precious labelled examples are primarily recruited to constitute the loss function for measuring the empirical risk in the LapSVM. In other words, the inherent information existing in these given data labels is not completely mined within the framework of the LapSVM. This is the immediate motivation of our research.

To create a semi-supervised SVM that makes extensive use of the knowledge embedded in the entire training data, regardless of the label availability, our strategy is as follows. For the known data labels, in addition to being used to control the empirical risk, the pairwise must-link and/or cannot-link constraints are enlisted to construct the pairwise constraint regularization formula (PCRF) and further the manifold and pairwise constraint jointly regularized formula (MPCJRF). Additionally, based on all of the training data, the smoothness condition is imposed on the possible solutions, and the graph Laplacian is used to embody the geometry structure of the data manifold. The optimization issue of our method can also be reformulated as a solvable quadratic programming problem. We designate our method as *semi-supervised classification with extensive knowledge exploitation (SSC-EKE)*. In summary, the contributions of our work are as follows:

1) The MPCJRF is not merely a simple combination of the manifold and pairwise constraint regularizations. It uses the implicit adjustment of pairwise constraints to the graph Laplacian to facilitate the unbiased approximation of the true data manifold. This is particularly valuable because the pairwise constraints generated from known data labels are known with a high degree of confidence.
2) In terms of the Min-Max theorem-based transformation, the two terms in the MPCJRF have the same magnitude. Therefore, with a trade-off factor varying within a fixed interval [0, 1) and by adopting cross-validation, the extent of the impact of the pairwise constraints on the graph Laplacian can be flexibly determined in any semi-supervised classification scenario.

Table 1
Common abbreviations used throughout this manuscript.

Abbreviation	Meaning
SVM	Support Vector Machine
S3VM	Semi-Supervised Support Vector Machine
LapSVM	Laplacian Support Vector Machine
LapRLS	Laplacian Regularized Least Square
TSVM	Transductive Support Vector Machine
CS4VM	Cost Sensitive Semi Supervised Support Vector Machine
WeLISVM	Weakly Labelled Support Vector Machine
MeanS3VM-*mkl*	Label-Mean-Based Semi-Supervised Support Vector Machine Regarding Multiple-Kernel Learning
MeanS3VM-*iter*	Label-Mean-Based Semi-Supervised Support Vector Machine Using Alternating Optimization
RKHS	Reproducing Kernel Hilbert Space
PCRF	Pairwise Constraint Regularized Formula
MPCJRF	Manifold Pairwise Constraint Jointly Regularized Formula
SSC-EKE	Semi-Supervised Classification with Extensive Knowledge Exploitation

3) SSC-EKE pursues, as much as possible, knowledge exploitation regarding both the labelled and unlabelled data in a training set. The labelled examples are used not only to minimize the empirical risk but also to develop the significant MPCJRF. Moreover, all of the data in the training set, particularly the numerous, unlabelled data instances, are involved in controlling the model smoothness as well as in depicting the underlying data manifold.
4) By incorporating the strengths of multiple theories, including the empirical risk minimization, the smoothness condition in an ambient space, joint manifold and pairwise constraint-based regularization, the spectral graph, and the Min-Max theorem, SSC-EKE features preferable classification effectiveness as well as generalizability.

The remainder of this manuscript is organized as follows. The work related to our research, such as the SVM, Representer Theorem, the manifold regularization, the LapSVM, the knowledge existing in the supervision, and the conversion between data labels and pairwise constraints, are briefly reviewed in Section 2. Our proposed PCRF and MPCJRF, the formulation as well as the Algorithm of SSC-EKE, and several relevant theorems are specifically introduced in Section 3. The comparisons of classification performance with regard to our proposed SSC-EKE and several other state-of-the-art S3VM approaches on both synthetic and real-world data sets are presented in Section 4. The conclusions regarding our work are given in Section 5.

2. Related work

To facilitate comprehension, some common abbreviations used throughout this paper are first listed in Table 1.

2.1. Support vector machine (SVM)

The SVM, proposed by Vapnik et al. [6,36,37], is a well-accepted technique for classification in pattern recognition. Instead of pursuing empirical risk minimization, the SVM is devoted to the overall risk minimization by minimizing the upper bound of the generalization error. By using a certain Mercer kernel, the SVM maps the data in the original feature space into those in a high-dimensional feature space to seek the optimal separating hyperplane in terms of maximizing the margin between two classes.

Let $X = \{\mathbf{x}_i \in \mathrm{R}^d,\ i = 1, \ldots, l\}$ denote the training set, l be the number of training examples, and $y_i \in \{+1, -1\}$ $(i = 1, \ldots, l)$ signify the labels of the corresponding data instances in X. Suppose that $f(.)$ represents the decision function and that H_K denotes the reproducing kernel Hilbert space associated with one Mercer kernel K. The framework of the SVM can then be formulated as

$$\min_{f \in H_k} \left(\frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(\mathbf{x}_i))_+ + \gamma \|f\|_K^2 \right), \tag{1}$$

where $(\ )_+$ is the hinge loss function, $(1 - yf(\mathbf{x}))_+ = \max(0, 1 - yf(\mathbf{x}))$, and $\gamma > 0$ is the regularization parameter.

Theorem 1. *(Representer Theorem [3]). Suppose that H_K denotes the corresponding RKHS. Then, the solution to the SVM optimization problem in the form of (1) can be expressed as*

$$f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}) \tag{2}$$

For the proof of Theorem 1, please refer to Appendix B.1.

Based on Theorem 1, and following the SVM expositions, i.e., with an unregularized bias term b being added to (2), the formulation of the SVM in the form of (1) can be equivalently rewritten as

$$\begin{aligned} &\min_{\alpha\in\mathrm{R}^l,\xi_i\in\mathrm{R}}\left(\frac{1}{l}\sum_{i=1}^{l}\xi_i+\gamma\alpha^{\mathrm{T}}\mathbf{K}\alpha\right),\\ &\text{s.t.}\quad y_i\left(\sum_{j=1}^{l}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)+b\right)\geq 1-\xi_i,\quad i=1,...,l,\\ &\qquad \xi_i\geq 0,\quad i=1,...,l, \end{aligned} \tag{3}$$

where $\boldsymbol{K}$ is the $l\times l$ Gram matrix with $K_{ij}=K(\mathbf{x}_i,\mathbf{x}_j)$; $K(.,.)$ is the enlisted kernel function.

Theorem 2 [3]. *Let $\beta=[\beta_1,...,\beta_l]^{\mathrm{T}}\in\mathrm{R}^l$ be the Lagrange multipliers, $\mathbf{Q}=\mathbf{Y}(\mathbf{K}/2\gamma)\mathbf{Y}$, $\mathbf{Y}=\mathrm{diag}(y_1,...,y_l)$, and diag(.) signify the generating function of the diagonal matrix. Then, the dual form of (3) is*

$$\begin{aligned} &\max_{\beta\in\mathrm{R}^l}\left(\frac{1}{l}\sum_{i=1}^{l}\beta_i-\frac{1}{2}\beta^{\mathrm{T}}\mathbf{Q}\beta\right),\\ &\text{s.t.}\quad \sum_{i=1}^{l}\beta_i y_i=0,\\ &\qquad 0\leq\beta_i\leq\frac{1}{l},\quad i=1,...,l. \end{aligned} \tag{4}$$

For the proof of Theorem 2, please refer to Appendix B.2.

Using the optimum $\boldsymbol{\beta}^*$ of (4), the eventual solution of (3) can be obtained, i.e., $\alpha^*=\mathbf{Y}\beta^*/2\gamma$.

2.2. Manifold regularization and LapSVM

Manifold regularization is essentially devoted not only to the smoothness of possible solutions but also to the utilization of knowledge from all available data instances. Its framework, developed by organically incorporating the theories of manifold learning and the spectral graph into the common regularization formulation of the SVM, was systematically discussed and presented in [3]. Let $S=\{\mathbf{x}_i\in\mathrm{R}^d,\ i=1,\ldots,l+u\}$ denote the training set consisting of l labelled examples and u unlabelled data instances, $V()$ signify the loss function, and the other notations be the same as those in (1); the framework of *manifold regularization* can then be generalized as

$$\min_{f\in H_k}\left(\frac{1}{l}\sum_{i=1}^{l}V(\mathbf{x}_i,y_i,f(\mathbf{x}_i))+\gamma_A\|f\|_K^2+\gamma_I\|f\|_I^2\right), \tag{5}$$

where γ_A and γ_I are the parameters of the second and third regularized terms.

As was previously mentioned, there are three terms in (5). The first one controls the empirical risk by using a certain loss function, the second avoids the overfitting issue by imposing the smoothness condition on possible solutions in the RKHS, and the last exploits the intrinsic geometric distribution of all data instances based on the manifold learning. To embody the intrinsic manifold nature of the data distribution, the structure of the data adjacency graph was used in [3], i.e.,

$$\|f\|_I^2=\frac{1}{(u+l)^2}\sum_{i,j=1}^{l+u}\big(f(\mathbf{x}_i)-f(\mathbf{x}_j)\big)^2W_{ij}=\frac{1}{(u+l)^2}\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}, \tag{6}$$

where $\mathbf{f}=[f(\mathbf{x}_1),...,f(\mathbf{x}_{l+u})]^{\mathrm{T}}$, $W_{ij}\in\mathbf{W}$ $(i,j=1,\ldots,u+l)$ are the edge weights in the data adjacency graph, $\boldsymbol{L}=\boldsymbol{D}$-$\boldsymbol{W}$ is termed the *graph Laplacian*, and $\boldsymbol{D}$ is the degree matrix of which the diagonal entries $D_{ii}=\sum_{j=1}^{l+u}W_{ij}$ and the others equal to 0.

If $V()$ is the hinge loss function, (5) can be expressed as

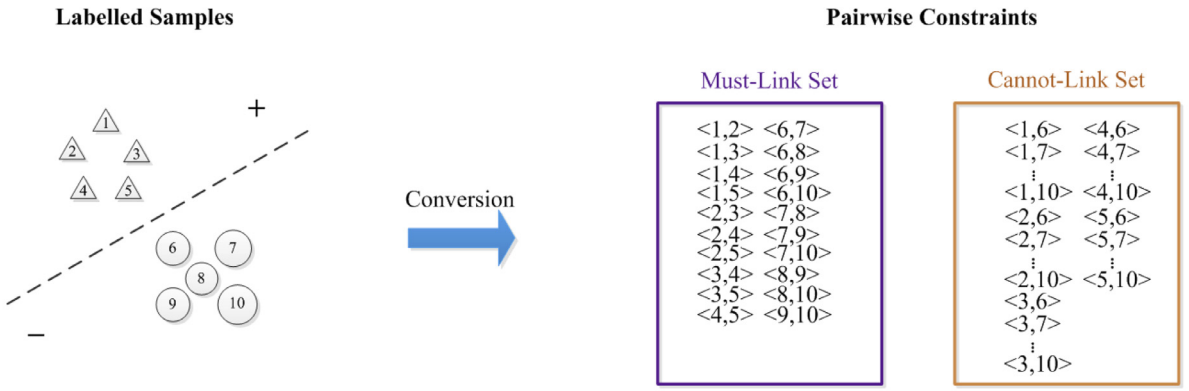
$$\min_{f\in H_k}\left(\frac{1}{l}\sum_{i=1}^{l}(1-y_i f(\mathbf{x}_i))_+ +\gamma_A\|f\|_K^2+\frac{\gamma_I}{(u+l)^2}\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}\right) \tag{7}$$

Theorem 3 [3]. *Let H_K denote the corresponding RKHS. The solution to the LapSVM in the form of (7) can be expressed as*

$$f^*(\mathbf{x})=\sum_{i=1}^{l+u}\alpha_i K(\mathbf{x}_i,\mathbf{x}) \tag{8}$$

For the proof of Theorem 3, please refer to Appendix B.3.

Based on Theorem 3, the problem of (7) is reduced to the optimization of coefficients α_i over the finite $(l+u)$-dimensional space. Following the SVM expositions and incorporating a bias term b into (8), the formulation of the LapSVM



(a) +/- classes and affiliated, labelled examples where the number within one triangle or circle denotes the index of one labelled example in the training set

(b) Attainable entries of pairwise constraints in the must-link/cannot-link sets from the labelled examples in which $<i,j>$ denotes the pairwise indices of examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

Fig. 1. Illustration of the conversion from data labels to pairwise constraints.

is subsequently obtained by reformulating (7) as

$$\begin{aligned} &\min_{\alpha\in\mathrm{R}^{l+u},\xi_i\in\mathrm{R}} \left(\frac{1}{l}\sum_{i=1}^{l}\xi_i + \gamma_A\alpha^{\mathrm{T}}\mathbf{K}\alpha + \frac{\gamma_I}{(u+l)^2}\alpha^{\mathrm{T}}\mathbf{K}\mathbf{L}\mathbf{K}\alpha \right), \\ &\text{s.t.} \quad y_i\left(\sum_{j=1}^{l+u}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)+b\right) \geq 1-\xi_i, \quad i=1,\ldots,l, \\ &\qquad \xi_i \geq 0, \quad i=1,\ldots,l. \end{aligned} \tag{9}$$

Theorem 4 [3]. *Let* $\beta=[\beta_1,\ldots,\beta_l]^{\mathrm{T}}\in\mathrm{R}^l$ *be the Lagrange multipliers,* $\mathbf{Q}=\mathbf{Y}\mathbf{J}\mathbf{K}(2\gamma_A\mathbf{I}+(2\gamma_I/(u+l)^2)\mathbf{L}\mathbf{K})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{Y}$, $\mathbf{J}=[\mathbf{I}\ \mathbf{0}]$ *be a* $l\times(l+u)$ *matrix, with* $\boldsymbol{I}$ *being the* $l\times l$ *identity matrix,* $\mathbf{Y}=\mathrm{diag}(y_1,y_2,\ldots,y_l)$, *and* $\boldsymbol{K}$ *be the* $(l+u)\times(l+u)$ *kernel matrix. Then, the dual form of* (9) *can be expressed as*

$$\begin{aligned} &\max_{\beta\in\mathrm{R}^l}\left(\frac{1}{l}\sum_{i=1}^{l}\beta_i-\frac{1}{2}\beta^{\mathrm{T}}\mathbf{Q}\beta\right), \\ &\text{s.t.} \quad \sum_{i=1}^{l}\beta_i y_i=0, \\ &\qquad 0\leq\beta_i\leq\frac{1}{l}, \quad i=1,\ldots,l. \end{aligned} \tag{10}$$

For the proof of Theorem 4, please refer to Appendix B.4.

As such, by using the solution of (10), the solution in (9) can be obtained using $\alpha^*=(2\gamma_A\mathbf{I}+(2\gamma_I/(u+l)^2)\mathbf{L}\mathbf{K})^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{Y}\beta^*$.

2.3. Knowledge existing in the supervision

In semi-supervised learning, class labels belong to the most common category of supervision, and the straightforward usage of class labels can be the least sophisticated form of knowledge exploitation. However, as another usual form of prior information, pairwise constraints, also referred to as must-link or cannot-link constraints, are usually of greater complexity. Depending on the specific cases offered by users, pairwise constraints can be in the form of a must-link set, in which the couples of any entry must be assigned to the same label, a cannot-link set, where the numbers in each entry must come from separate groups, or both.

The supervision in the form of class labels or pairwise constraints is interdependent, and there actually exist conversions between them. According to the different data labels existing in the supervision, the labelled examples can be divided into several groups. Only one group exists, as a special case, if and only if all the given labels are consistent. Suppose that the data number in each group is more than one; then, any two examples within one group can certainly be used to constitute the must-link set, and any example pair of which the members are separately from two inconsistent groups should certainly be an entry in the cannot-link set. In the special case of only one group, the must-link set is available but the cannot-link set is not.

As an example, Fig. 1 illustrates the feasible conversion from class labels to pairwise constraints, where there are five data instances in each of the positive and negative classes, respectively, as shown in Fig. 1(a). The attainable entries in the must-link/cannot-link set generated by these labelled examples are specifically indicated in Fig. 1(b). Intuitively, regarding the knowledge exploitation, the prior information in the form of must-link/cannot-link constraints appears to be more informative than that of class labels.

3. Semi-supervised classification via extensive knowledge exploitation

Before introducing our own work, we present the following three aspects of comprehension with regard to existing semi-supervised classification techniques.

1) The accuracy and generalizability performance of conventional classifiers depends on the quality and quantity of training examples. However, due to the limited amount of labelled data, many semi-supervised classification methods are designed to effectively exploit the knowledge embedded in many label-unknown data instances rather than in the few labelled examples.
2) In many existing S3VMs, such as the LapSVM, LapRLS [3], MeanS3VMs [15,22], and CS4VM [16], the few but precious label-known examples are primarily used to control the empirical risk, which usually neglects to make extensive use of this form of supervision data.
3) To utilize the label-unknown data instances, many semi-supervised classifiers work based on certain assumptions. For example, the LapSVM relies on the premise that the extracted graph structure of marginal distributions can effectively depict the ground truth of the data manifold. Such assumptions, nevertheless, are sometimes difficult to guarantee and verify, particularly in a situation where much interference information, such as noise or outliers, exists.

Motivated by such challenges, we develop our own scheme, based on the LapSVM, for semi-supervised classification, as follows:

3.1. Pairwise constraint regularization

As described in Section 2.3, because the given data labels can easily be converted into the must-link/cannot-link constraints and because the latter appears to be more insightful than the former, we first devise the following pairwise constraint regularization mechanism.

Definition 1. Let $\mathbf{f} = [f_1, ..., f_l, f_{l+1}, ..., f_{l+u}]^{\mathrm{T}}$ and f_i $(i = 1, ..., l+u)$ denote the prediction results of the data instances in $S = \{\mathbf{x}_i \in \mathrm{R}^d,\ i = 1, \ldots, l+u\}$ via the discriminant function f. Suppose that *MS* and *CS* signify the must-link set and cannot-link set, respectively, derived from the given, insufficient labelled examples and that $|.|$ signifies the entry number in the *MS* or *CS*. The *pairwise constraint regularization formula* (*PCRF*) can then be defined as

$$\min_{\mathbf{f}} \left(\frac{\sum_{<i,j>\in MS} \left(f_i - f_j\right)^2}{|MS|} - \frac{\sum_{<p,q>\in CS} (f_p - f_q)^2}{|CS|} \right), \tag{11}$$

where $i, j, p, q \in [1, l+u]$; $\langle i, j\rangle$ denotes any entry in the *MS*, and i and j are their individual data indices in S. Similarly, $\langle p, q\rangle$ indicates any entry in the *CS*, with p and q being the corresponding data indices.

In light of the fact that any two examples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, corresponding to $<i, j> \in MS$, should have the same label, i.e., either $+1$ or -1, the ideal decision function f should at least keep the signs of $f_i = f(\mathbf{x}_i)$ and $f_j = f(\mathbf{x}_j)$ the same. Such a condition can actually be achieved by minimizing $(f_i - f_j)^2$, which inductively minimizes $\sum_{<i,j>\in MS} (f_i - f_j)^2$. In contrast, for any two examples $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$, corresponding to $<p, q> \in CS$, the goal is to have opposite signs, which is equivalent to minimizing $-(f_p - f_q)^2$ and thus also $\sum_{<p,q>\in CS} (-(f_p - f_q)^2)$. In view of the potential capacity gap between the *MS* and *CS*, the averages of $\sum_{<i,j>\in MS} (f_i - f_j)^2$ and $\sum_{<p,q>\in CS} (-(f_p - f_q)^2)$ are listed in (11).

Theorem 5. *Let us define a matrix* $\mathbf{Q}_{(l+u)\times(l+u)}$ *having elements*

$$Q_{ij} = Q_{ji} = \begin{cases} 1/|MS|, & \forall < i, j > \in MS \text{ or } < j, i > \in MS; \\ -1/|CS|, & \forall < i, j > \in CS \text{ or } < j, i > \in CS; \\ 0 & \text{default}, \end{cases} \tag{12}$$

and use the same notations as those in Definition 1. Then, the proposed PCRF in the form of (11) can be reformulated as

$$\min_{\mathbf{f}} \left(\mathbf{f}^{\mathrm{T}}\mathbf{Z}\mathbf{f}\right), \tag{13-1}$$

$$\mathbf{Z} = \mathbf{H} - \mathbf{Q}, \tag{13-2}$$

$$\mathbf{H} = \mathrm{diag}(\mathbf{Q} \cdot \mathbf{1}_{(l+u)\times 1}), \tag{13-3}$$

where $\mathbf{1}_{(l+u)\times 1}$ *denotes one* $(l+u) \times 1$ *constant vector of which the elements are all 1.*

Proof. As discovered in (6), $\min(\sum_{i,j=1}^{l+u} (f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 W_{ij}) = \min(\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}) = \min(\mathbf{f}^{\mathrm{T}}(\mathbf{D} - \mathbf{W})\mathbf{f})$. With this transformation as a reference, the above theorem can be easily proven. Here, the roles of $\boldsymbol{Q}$, $\boldsymbol{H}$, and $\boldsymbol{Z}$ are similar to those of $\boldsymbol{W}$, $\boldsymbol{D}$, and $\boldsymbol{L}$ in (6), respectively. □

As is evident, by converting the given labels into pairwise constraints and by means of the PCRF, not only do we obtain a novel regularization mechanism, but the information existing in the insufficient labelled example is further expanded and exploited.

3.2. Manifold and pairwise constraint jointly regularized formula

The LapSVM is closely associated with estimating the manifold structure $\|f\|_I^2$. However, it is not guaranteed that the enlisted data adjacency graph in (6) can always depict an unbiased estimate of the ground truth of the underlying manifold, which significantly impacts the performance of the LapSVM. To resolve this problem in our work, we put forward the dedicated countermeasure below.

Because (6) and (13) have a similar composition, with one parameter $\beta > 0$, it is reasonable to combine them as

$$\min_{\mathbf{f}} \left(\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f} + \beta\mathbf{f}^{\mathrm{T}}\mathbf{Z}\mathbf{f}\right) = \min_{\mathbf{f}} \mathbf{f}^{\mathrm{T}}(\mathbf{L} + \beta\mathbf{Z})\mathbf{f}. \tag{14}$$

Theorem 6. *Let* $\boldsymbol{W}$ *and* $\boldsymbol{Q}$ *be the same as those in (6) or (12), respectively. Then, (14) implies that there is a generalized matrix* $\mathbf{W}' = \mathbf{W} + \beta\mathbf{Q}$*, which embodies the adjustment of the pairwise constraints to the estimated manifold structure.*

Proof. Because $\mathbf{L} + \beta\mathbf{Z} = \mathbf{D} - \mathbf{W} + \beta(\mathbf{H} - \mathbf{Q}) = \mathbf{D} + \beta\mathbf{H} - (\mathbf{W} + \beta\mathbf{Q}) = \mathrm{diag}(\mathbf{W} \cdot \mathbf{1}_{(l+u)\times 1}) + \beta\mathrm{diag}(\mathbf{Q} \cdot \mathbf{1}_{(l+u)\times 1}) - (\mathbf{W} + \beta\mathbf{Q}) = \mathrm{diag}((\mathbf{W} + \beta\mathbf{Q}) \cdot \mathbf{1}_{(l+u)\times 1}) - (\mathbf{W} + \beta\mathbf{Q})$, with $\mathbf{W}' = \mathbf{W} + \beta\mathbf{Q}$, i.e.,

$$W'_{ij} = \begin{cases} W_{ij} + \beta/|MS|, & \forall < i, j > \in MS \text{ or } < j, i > \in MS, \\ W_{ij} - \beta/|CS|, & \forall < i, j > \in CS \text{ or } < j, i > \in CS, \\ W_{ij} & otherwise, \end{cases} \tag{15}$$

we arrive at $\mathbf{L} + \beta\mathbf{Z} = \mathrm{diag}(\mathbf{W}' \cdot \mathbf{1}_{(l+u)\times 1}) - \mathbf{W}'$ immediately.

Intuitively, (15) exhibits the manipulation, with respect to the pairwise constraints, of the original data adjacency measurements for the graph Laplacian, i.e., the estimated manifold structure. □

In (15), the parameter β balances the overall impact of the pairwise constraints on the original adjacency weights. However, we have observed that the appropriate scale of the parameter β is sometimes difficult to estimate, particularly when $\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}\mathbf{f}$ in (14) have different orders of magnitude. To address this, we apply Theorem 7.

Theorem 7. *Suppose that* $\boldsymbol{M}$ *is any* $(l+u) \times (l+u)$ *symmetric matrix and that* $\boldsymbol{f}$ *is the same as that in Definition 1. For* $\mathbf{f}^{\mathrm{T}}\mathbf{M}\mathbf{f}$*, the range of which was previously uncertain, by using the transformation* $\mathbf{M}' = \frac{\mathbf{M} - \lambda_{\min\_\mathbf{M}}\mathbf{I}}{\lambda_{\max\_\mathbf{M}} - \lambda_{\min\_\mathbf{M}}}$*, where* $\lambda_{\min\_\mathbf{M}}$ *and* $\lambda_{\max\_\mathbf{M}}$ *refer to the minimal and maximal eigenvalues of* $\boldsymbol{M}$*, respectively, and* $\boldsymbol{I}$ *is the identity matrix, one can arrive at*

$$0 \le \mathbf{f}^{\mathrm{T}}\mathbf{M}'\mathbf{f} \le \mathbf{f}^{\mathrm{T}}\mathbf{f}. \tag{16}$$

Proof. According to the *Rayleigh quotient* [31,32] and the *Min-Max theorem* [21], one can obtain the following inequality:

$$\lambda_{\min\_\mathbf{M}} \le \frac{\mathbf{f}^{\mathrm{T}}\mathbf{M}\mathbf{f}}{\mathbf{f}^{\mathrm{T}}\mathbf{f}} \le \lambda_{\max\_\mathbf{M}}. \tag{17}$$

Furthermore, (17) equals to

$$\begin{aligned} &\lambda_{\min\_\mathbf{M}}\mathbf{f}^{\mathrm{T}}\mathbf{f} \le \mathbf{f}^{\mathrm{T}}\mathbf{M}\mathbf{f} \le \lambda_{\max\_\mathbf{M}}\mathbf{f}^{\mathrm{T}}\mathbf{f} \\ \Leftrightarrow\ &0 \le \mathbf{f}^{\mathrm{T}}\mathbf{M}\mathbf{f} - \lambda_{\min\_\mathbf{M}}\mathbf{f}^{\mathrm{T}}\mathbf{f} \le \lambda_{\max\_\mathbf{M}}\mathbf{f}^{\mathrm{T}}\mathbf{f} - \lambda_{\min\_\mathbf{M}}\mathbf{f}^{\mathrm{T}}\mathbf{f} \\ \Leftrightarrow\ &0 \le \mathbf{f}^{\mathrm{T}}(\mathbf{M} - \lambda_{\min\_\mathbf{M}}\mathbf{I})\mathbf{f} \le (\lambda_{\max\_\mathbf{M}} - \lambda_{\min\_\mathbf{M}})\mathbf{f}^{\mathrm{T}}\mathbf{f}. \end{aligned} \tag{18}$$

Therefore, this theorem can be proven by rearranging (18). □

Because both $\boldsymbol{L}$ and $\boldsymbol{Z}$ are symmetric, based on Theorem 7 and by using

$$\mathbf{L}' = \frac{\mathbf{L} - \lambda_{\min\_\mathbf{L}}\mathbf{I}}{\lambda_{\max\_\mathbf{L}} - \lambda_{\min\_\mathbf{L}}} \tag{19}$$

and

$$\mathbf{Z}' = \frac{\mathbf{Z} - \lambda_{\min\_\mathbf{Z}}\mathbf{I}}{\lambda_{\max\_\mathbf{Z}} - \lambda_{\min\_\mathbf{Z}}}, \tag{20}$$

where $\lambda_{\min\_\mathbf{L}}$ and $\lambda_{\max\_\mathbf{L}}$ are the minimal and maximal eigenvalues of $\boldsymbol{L}$, respectively, and $\lambda_{\min\_\mathbf{Z}}$ and $\lambda_{\max\_\mathbf{L}}$ are those of $\boldsymbol{Z}$, we can attain $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$, which have the same range.

Thus far, we can propose the significant manifold and pairwise constraint jointly regularized formula as follows.

Definition 2. Derived from (14), by using (19) and (20), the manifold and pairwise constraint jointly regularized formula is defined as

$$\min_{\mathbf{f}} \left(\Phi_{MPCJRF}(\mathbf{f}) = (1-\tau)\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f} + \tau\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f} = \mathbf{f}^{\mathrm{T}}\left((1-\tau)\mathbf{L}' + \tau\mathbf{Z}'\right)\mathbf{f}\right). \tag{21}$$

Differing from $\mathbf{f}^{\mathrm{T}}\mathbf{L}\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}\mathbf{f}$ in (14), the ranges of $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$ in (21) are now consistent. Thus, a simple trade-off coefficient, $\tau \in [0, 1)$, can self-adaptively control their individual significance in any data scenario.

3.3. Semi-supervised classification based on extensive knowledge exploitation

3.3.1. The framework of SSC-EKE

Now, incorporating the MPCJRF in the form of (21) into (1), we can derive our method for semi-supervised classification as follows.

Definition 3. Using the same notations as those in (1) and (7) and following the principle of minimum structure risk of the SVM, the formulation of our semi-supervised classification with extensive knowledge exploitation can be finally defined as

$$\begin{aligned}&\min_{f\in H_k}\left(\frac{1}{l}\sum_{i=1}^{l}(1-y_i f(\mathbf{x}_i))_+ + \gamma_A\|f\|_K^2 + \gamma_J\Phi_{MPCJRF}(\mathbf{f})\right)\\ &=\min_{f\in H_k}\left(\frac{1}{l}\sum_{i=1}^{l}(1-y_i f(\mathbf{x}_i))_+ + \gamma_A\|f\|_K^2 + \gamma_J\mathbf{f}^{\mathrm{T}}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\mathbf{f}\right),\end{aligned} \tag{22}$$

where $\gamma_J>0$ is the regularization parameter for the term of the MPCJRF.

Likewise, referring to Theorem 3 and following the SVM expositions, we can reformulate (22) as

$$\begin{aligned}&\min_{\alpha\in\mathrm{R}^{l+u},\xi_i\in\mathrm{R}}\left(\frac{1}{l}\sum_{i=1}^{l}\xi_i+\gamma_A\alpha^{\mathrm{T}}\mathbf{K}\alpha+\gamma_J\alpha^{\mathrm{T}}\mathbf{K}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\mathbf{K}\alpha\right),\\ &\text{s.t.}\quad y_i\left(\sum_{j=1}^{l+u}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)+b\right)\ge 1-\xi_i,\quad i=1,\ldots,l,\\ &\qquad \xi_i\ge 0,\quad i=1,\ldots,l.\end{aligned} \tag{23}$$

Theorem 8. *Let us define a matrix* $\mathbf{P}=\begin{bmatrix} y_1K_{11} & y_2K_{21} & \cdots & y_lK_{l1}\\ y_1K_{12} & y_2K_{22} & \cdots & y_lK_{l2}\\ \vdots & \vdots & \cdots & \vdots\\ y_1K_{1(l+u)} & y_2K_{2(l+u)} & \cdots & y_lK_{l(l+u)}\end{bmatrix}_{(l+u)\times l}$, *where* $K_{ij}=K(\mathbf{x}_i,\mathbf{x}_j),\ i\in[1,l],\ j\in[1,l+u]$. *Suppose that* $\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_l)$ *denotes the Lagrange multipliers. Then, (23) is equivalent to the following optimization problem:*

$$\begin{aligned}&\max_{\beta\in\mathrm{R}^l}\left(\sum_{i=1}^{l}\beta_i-\tfrac{1}{4}\beta^{\mathrm{T}}\mathbf{S}\beta\right),\\ &\text{s.t.}\quad \sum_{i=1}^{l}\beta_i y_i=0,\\ &\qquad 0\le\beta_i\le\tfrac{1}{l},\quad i=1,\ldots,l,\end{aligned} \tag{24}$$

where $\mathbf{S}=\mathbf{P}^{\mathrm{T}}(\gamma_A\boldsymbol{K}+\gamma_J\boldsymbol{K}((1-\tau)\mathbf{L}'+\tau\mathbf{Z}')\boldsymbol{K})^{-1}\mathbf{P}$.

Proof. By using the Lagrange multipliers $\boldsymbol{\beta}=(\beta_1,\beta_2,\ldots,\beta_l)$ and $\boldsymbol{\gamma}=(\gamma_1,\gamma_2,\ldots,\gamma_l)$, we first obtain the Lagrange function:

$$\begin{aligned}L\big(\boldsymbol{\alpha},b,\xi,\boldsymbol{\beta},\boldsymbol{\gamma}\big)=&\tfrac{1}{l}\sum_{i=1}^{l}\xi_i+\alpha^{\mathrm{T}}\big(\gamma_A\boldsymbol{K}+\gamma_J\boldsymbol{K}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\boldsymbol{K}\big)\boldsymbol{\alpha}\\ &-\sum_{i=1}^{l}\beta_i\left(y_i\left(\sum_{j=1}^{l+u}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j)+b\right)-1+\xi_i\right)-\sum_{i=1}^{l}\gamma_i\xi_i.\end{aligned} \tag{25}$$

According to the KKT conditions, we have

$$\begin{aligned}&\tfrac{\partial L}{\partial\alpha}=0\Leftrightarrow 2\big(\gamma_A\boldsymbol{K}+\gamma_J\boldsymbol{K}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\boldsymbol{K}\big)\boldsymbol{\alpha}-\sum_{i=1}^{l}\begin{bmatrix}\beta_i y_i K_{i1}\\ \vdots\\ \beta_i y_i K_{i(l+u)}\end{bmatrix}_{(l+u)\times 1}=0\\ &\quad\Leftrightarrow 2\big(\gamma_A\boldsymbol{K}+\gamma_J\boldsymbol{K}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\boldsymbol{K}\big)\boldsymbol{\alpha}=\mathbf{P}\beta\\ &\quad\Leftrightarrow\boldsymbol{\alpha}=\tfrac{1}{2}\big(\gamma_A\boldsymbol{K}+\gamma_J\boldsymbol{K}\big((1-\tau)\mathbf{L}'+\tau\mathbf{Z}'\big)\boldsymbol{K}\big)^{-1}\mathbf{P}\beta;\end{aligned} \tag{26}$$

$$\frac{\partial L}{\partial b}=0\Leftrightarrow\sum_{i=1}^{l}\beta_i y_i=0; \tag{27}$$

$$\frac{\partial L}{\partial\xi_i}=0\Leftrightarrow\frac{1}{l}-\beta_i-\gamma_i=0\Leftrightarrow 0\le\beta_i\le\frac{1}{l}. \tag{28}$$

Substituting (26)–(28) into (25), the dual of (23) is achieved, i.e., (24). □

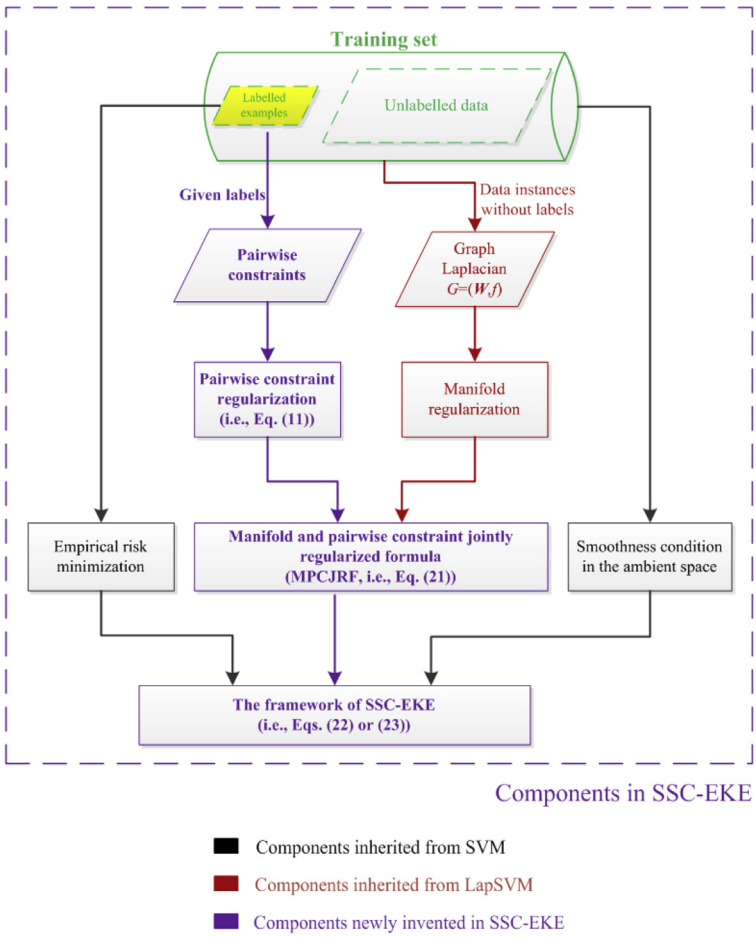


Fig. 2. The composition of the formulation of SSC-EKE.

In terms of the solution $\boldsymbol{\beta}^*$ of (24), the original solution of (23) can be given by

$$\alpha^* = \frac{1}{2}\left(\gamma_A \boldsymbol{K} + \gamma_I \boldsymbol{K}\left((1-\tau)\mathbf{L}' + \tau \mathbf{Z}'\right)\boldsymbol{K}\right)^{-1} \mathbf{P}\beta^*, \tag{29–1}$$

$$b^* = \frac{1}{l}\sum_{i=1}^{l}\left(y_i - \sum_{j=1}^{l+u}\alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{29–2}$$

and the final classification decision function can be expressed as

$$f^*(\mathbf{x}) = \sum_{i=1}^{l+u}\alpha_i^* K(\mathbf{x}, \mathbf{x}_i) + b^*. \tag{30}$$

3.3.2. Other explanations with respect to SSC-EKE

To facilitate comprehension, we describe the meanings and origins of the components in the formulation of SSC-EKE in Fig. 2. The SVM and LapSVM are two of the foundations of our work. Except for the newly devised MPCJRF, SSC-EKE inherits the manifold regularization from the LapSVM and the other components from the SVM.

It should be noted that with the overall framework in the form of (22) or (23), SSC-EKE manifests a significant advantage: the knowledge embedded in the two categories of training data for semi-supervised classification, i.e., few label-known examples and numerous label-unknown data instances, is extensively exploited. The detailed explanations are as follows.

i. Explicit usages regarding the labelled and unlabelled data for training in SSC-EKE

The labelled examples are recruited to control the empirical risk (see $\sum_{i=1}^{l}(1-y_i f(\mathbf{x}_i))_+$ in (22)) and to impose the pairwise constraint regularization (see $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$ in (22)) on the objective function. Meanwhile, many unlabelled data in-

stances, along with few labelled ones, are involved in estimating the underlying manifold structure (see $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ in (22)) and controlling the model smoothness in the reproducing kernel Hilbert space in terms of $\|f\|_K^2$.

i. Implicit efficacies regarding the labelled and unlabelled data for training in SSC-EKE

Because the MPCJRF in the form of (21) is derived from (14), as revealed in Theorem 2, by using the implicit, generalized adjacency matrix $\mathbf{W}' = \mathbf{W} + \beta\mathbf{Q}$, the underlying adjustment of the pairwise constraints to the estimated manifold structure occurs. In addition, based on Theorem 3, we are able to transform $\boldsymbol{L}$ and $\boldsymbol{Z}$ into $\mathbf{L}'$ and $\mathbf{Z}'$, respectively, and then we obtain the same range for $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$, i.e., $[0, \mathbf{f}^{\mathrm{T}}\mathbf{f}]$. Thus, with the trade-off factor τ taking values between 0 and 1, it is viable for us to flexibly determine the individual impacts of $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$ in (21). As such, the manifold and pairwise constraint jointly regularized mechanism is achieved.

Lastly, let us come back to the drawbacks of existing semi-supervised classification techniques, which we mentioned at the beginning of Section 3. All of them are addressed by means of our SSC-EKE schema. Specifically, the first two problems are resolved by converting labels into many must-link and cannot-link constraints and further presenting the pairwise constraint regularization mechanism. The third problem is resolved by devising the MPCJRF in the form of (21) to obtain an effective pathway to flexibly correct the estimated data manifold structure by using the given labels.

3.4. The algorithm of SSC-EKE

In this section, we detail the Algorithm of the proposed SSC-EKE method.

Algorithm: Semi-Supervised Classification with Extensive Knowledge Exploitation (SSC-EKE).

Input:	l labelled example $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ and u unlabelled data instances $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$.
Output:	Decision function $f(\mathbf{x})$.
Step 1:	Construct the data adjacent graph via the $(l+u)$ data instances and then generate the edge weight matrix $\boldsymbol{W}$ as well as graph Laplacian matrix $\boldsymbol{L}$;
Step 2:	Generate the must-link and cannot-link sets (i.e., *MS* and *CS*) in terms of the l labelled examples, referring to Section 2.3;
Step 3:	Constitute the pairwise constraint matrix $\boldsymbol{Q}$ and further the matrix $\boldsymbol{Z}$ in terms of *MS* and *CS*, according to Theorem 5;
Step 4:	Transform L and Z into $\mathbf{L}'$ and $\mathbf{Z}'$ so that $\mathbf{f}^{\mathrm{T}}\mathbf{L}'\mathbf{f}$ and $\mathbf{f}^{\mathrm{T}}\mathbf{Z}'\mathbf{f}$ have the same range—$[0, \mathbf{f}^{\mathrm{T}}\mathbf{f}]$, according to (19) and (20), respectively;
Step 5:	Compute the optimum solution β^* of (24), and then generate the optima,$\alpha^*$and b^*, of (23) via (29);
Step 6:	Output the discriminant function $f(x)$ using (30).

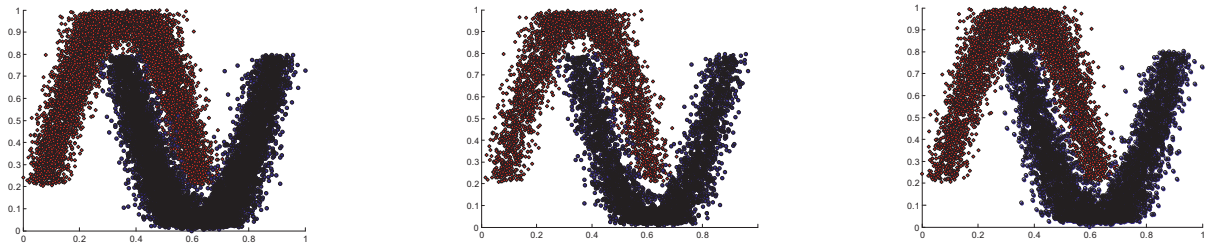
4. Experimental studies

4.1. Setup

To evaluate the performance of our proposed SSC-EKE approach, we systematically compare it with eight other state-of-the-art methods, including the classic SVM (see (3)), LapSVM (see (9)), LapRLS [3], CS4VM [16], TSVM [13], WeLlSVM [17], and two versions of the MeanS3VM –MeanS3VM-*iter* and MeanS3VM-*mkl* [15]. Among these, the TSVM is one of the predecessors in semi-supervised classification; the LapRLS and LapSVM are two representatives of manifold regularization-based S3VMs, with the LapSVM also being the foundation of our SSC-EKE approach; and the other four, as introduced in Section 1, are well-established S3VMs of which the semi-supervised mechanisms differ from our SSC-EKE strategy. Except for the classic SVM, the others are all semi-supervised classification methods.

To measure the realistic classification performance of all enlisted algorithms, the conventional accuracy index (ACC) [7,19] is used. Moreover, to specifically differentiate the performances of different algorithms, the well-established F1 score [46] is also investigated in standard binary classification issues in our experiments. Each approach is performed 20 times on each employed data set using inconsistent supervision subsets, which will be subsequently described. To achieve a balance between good readability and appropriate manuscript length, we separate our experimental content into two parts. The classification performance measured using the ACC and the statistical analysis metric of all methods on all data sets are listed in this section, and some comments regarding the experimental outcomes are also presented in this section. The supplemental content, such as the F1 scores of all algorithms on some binary classification data sets, are reported in the Appendix.

The parameter settings of all nine algorithms are given as follows. Both the linear and Gaussian RBF kernels were used in our experiments, with the width σ in the Gaussian RBF kernel, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$, being set to the average distance among all data instances. Both parameters C_1 and C_2 were selected to be within {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1} in the WeLlSVM, whereas the trial ranges were {0.1, 0.5, 0.7, 1, 10, 50, 100, 200} in the TSVM, MeanS3VM-*iter*, MeanS3VM-*mkl*, and CS4VM. The KNN was used in the LapSVM, LapRLS, and SSC-EKE to constitute the data adjacency graph, and the number of nearest neighbours was sought within {1, 3, 5, 7, 9} throughout our experiments. The parameter values of γ_A and γ_I in the LapSVM and LapRLS, γ_A and γ_I in SSC-EKE, and γ in the SVM were chosen to be within {10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 10^2}. In addition, τ in SSC-EKE varied from 0.05 to 1, with the step size being 0.05. These parameters in related algorithms were eventually determined using the cross-validation strategies. More specifically, the leave-one-out

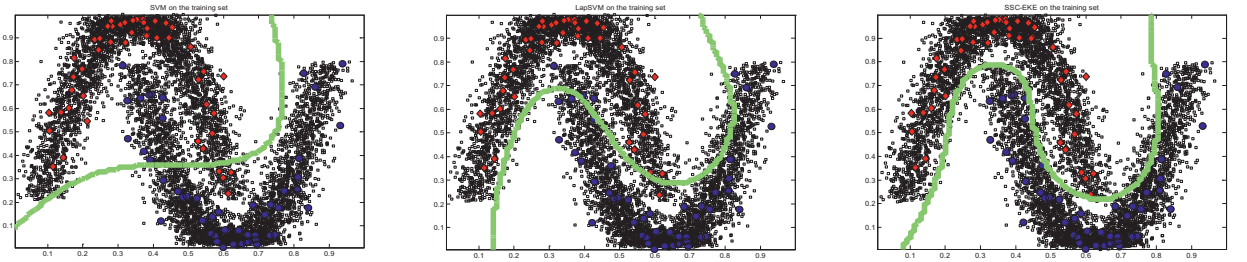


(a) The whole data distribution in *DS*1 (b) The training data in *DS*1 (c) The testing data in *DS*1

Fig. 3. The synthetic data set *DS*1.

Table 2
Classification accuracies of all nine algorithms on *DS*1 (with RBF kernel).

Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
ACC	87.3 ± 1.2(7)	88.5 ± 1.7(6)	94.6 ± 2.1(2)	90.1 ± 8.2(5)	86.9 ± 0.5(8)	82.1 ± 5.7(9)	92.2 ± 4.7(4)	93.7 ± 1.6(3)	**96.4 ± 1.3(1)**



(a) Classification hyperplane of SVM (b) Classification hyperplane of LapSVM (c) Classification hyperplane of SSC-EKE

Fig. 4. Learned classification hyperplanes of SVM, LapSVM, and SSC-EKE.

cross-validation [2,5] was adopted when labelled data sizes were less than or equal to 20; otherwise, the fivefold cross-validation was used.

All enlisted data sets were normalized before they were used in our experiments by using the formula $x'_{id} = \frac{x_{id} - \min\{x_{1d},\ldots x_{Nd}\}}{\max\{x_{1d},\ldots x_{Nd}\} - \min\{x_{1d},\ldots x_{Nd}\}}$, where i and d denote the indices of the data instance and dimension, respectively. Moreover, all experiments were conducted using a PC with an i5-4590 3.30 GHz CPU, 4GB of RAM, Microsoft Windows 7 (64 bit), and MATLAB R2013a (64 bit).

4.2. Experiment on synthetic data set

We first verify the performance of all of the involved approaches using synthetic data wherein the true answer is known. To this end, as shown in Fig. 3, we artificially generated one two-dimensional, two-moon-shaped data set denoted as *DS*1, in which the data size was 16,040. To simulate the situation of (semi-)supervised classification, the original *DS*1 was arbitrarily divided into two groups, with the data numbers being 7000 and 9040. The group of 7000 records was selected to act as the training set, while the other as the testing set. The 100 examples randomly selected from the training set were enlisted as the supervision subset, i.e., the labelled data, and only the RBF kernel was used during our experiment because *DS*1 is apparently non-linearly separable.

We separately ran the nine algorithms on *DS*1, and the classification accuracies, in the form of ACC means and standard deviations, are listed in Table 2, where the ranks achieved by all algorithms are shown in the parentheses. For the detailed classification correctness of each algorithm with respect to the positive and negative classes in the testing set, one can refer to Table A.1, which is additionally listed in the Appendix.

In addition, because *DS*1 is two-dimensional, we illustrate the learning performance of all approaches in terms of their learned classification hyperplanes. Due to the limited manuscript length, here we only show one of the scenes of the SVM, LapSVM, and our SSC-EKE in Fig. 4, where the 100 labelled examples in the positive and negative classes are shown in red and blue, respectively, and the classification hyperplanes are shown in bright green. Our SSC-EKE algorithm ranks first on *DS*1. This is due to the benefits of the extensive exploitation of knowledge contained in both labelled and unlabelled training

Table 3
Details of eighteen, involved Benchmark/UCI/KEEL data sets.

Dataset	Training set		Testing set data size	Dimension	Class number
	Data size	Labelled sample size			
wine	80	10,20,30,40,50	98	13	2
sonar	100	10,20,30,40,50	108	60	2
house	130	10,20,30,40,50	102	16	2
spectfheart	160	10,20,30,40,50	107	44	2
ionosphere	200	10,20,30,40,50	151	34	2
house-votes	290	10,20,30,40,50	145	16	2
WDBC	300	10,20,30,40,50	269	14	2
monk2	300	10,20,30,40,50	301	6	2
breast	400	10,20,30,40,50	299	10	2
diabetes	400	10,20,30,40,50	368	8	2
vehicle	500	10,20,30,40,50	346	18	2
german	500	10,20,30,40,50	500	59	2
BCI	200	10,20,30,40,50	200	117	2
USPS	1000	10,20,30,40,50	500	241	2
digit1	1000	10,20,30,40,50	500	241	2
DNA	1600	10,20,30,40,50	1586	180	2
Iris	70	10,20,30,40,50	80	4	3
balance-scale	440	10,20,30,40,50	225	4	3

data. Moreover, the classification hyperplane of SSC-EKE shown in Fig. 4(c) creates a more natural separation between the groups than those of the other methods.

4.3. Experiments on Benchmark/UCI/KEEL data sets

Next, eighteen well-established data sets from three famous repositories, i.e., the *Benchmark data sets*,[1] *UCI*,[2] and *KEEL* (Knowledge Extraction based on Evolutionary Learning) ,[3] were used in our experiments. The details of these data sets are listed in Table 3. Please note that the last two data sets in Table 3, i.e., *iris* and *balance-scale*, contain three classes. Therefore, the voting strategy [1,9,10,33,42,45] was recruited in our studies to solve multiclass classification problems. Specifically, regarding the given labelled examples, we first divided them into different groups according to their labels. Then, with any two different groups acting as the positive and negative classes, respectively, we separately trained multiple classifiers. Last, the labels of the data instances in the testing set were determined according to the majority principle.

To compose the (semi-)supervised classification scenes and evaluate the classification performance of different approaches with respect to different supervision capacities, i.e., different numbers of labelled examples, we randomly sampled each original training set of each data set twenty times, with the sample sizes being 10, 20, 30, 40, and 50, respectively. In this way, we obtained twenty inconsistent subsets matching each sampling capacity on each data set. With the twenty subsets of each sample size acting separately as the supervision for (semi-)supervised learning, we ran the nine classification approaches on each data set and obtained twenty classification outcomes from each of them.

We report the classification performance of these nine algorithms in Tables 4 and 5. The accuracies of the nine algorithms on each data set with the 20 and 40 sample sizes are shown in Tables 4 and 5, respectively, in the form of ACC means and standard deviations. The best accuracy on each data set is denoted using bold font. Moreover, statistical analyses were conducted in our experiments, including the average ACCs, ranks of all algorithms, and paired t-test scores [17,18,40], i.e., the win/tie/loss counts, of all semi-supervised approaches versus the classic SVM and of SSC-EKE against the LapSVM, both at the significance level of 0.05. In addition, the classification accuracies regarding these nine approaches with respect to multiple supervision capacities are illustrated in Figs. 5 and 6, in which, due to the limited manuscript length, we only show the representative cases of the nine algorithms on 10 data sets, i.e., *sonar, house, house-votes, mpnk2, diabetes, vehicle, german, BCI, USPS*, and *digit1*, in the cases of linear and RBF kernels, respectively. In addition, the specific classification correctness of the positive and negative classes in the testing sets of all the algorithms on all data sets, measured in terms of the F1 scores, was also calculated in our experiments. However, due to the limited manuscript length, only the outcomes on twelve binary classification data sets are shown in Tables A.2 and A.3 in the Appendix.

The analysis results regarding the performances of all tested algorithms are as follows.

1) The SSC-EKE algorithm generally performs well on most of the involved data sets. It achieves the best average ACC, highest average rank (Tables 4 and 5), and the best overall performance versus the other seven semi-supervised classification methods according to the t-tests (Tables 4 and 5). As indicated in Tables A.2 and A.3, SSC-EKE generally achieves the

[1] http://olivier.chapelle.cc/ssl-book/benchmarks.html.
[2] http://archive.ics.uci.edu/ml/.
[3] http://www.keel.es/.

Table 4
Classification accuracies of all nine algorithms with the supervision size of 20.

Linear kernel

Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
Wine	96.7 ± 3.0(5)	97.1 ± 2.9(3)	91.8 ± 4.6(8)	97.6 ± 2.9(2)	95.4 ± 3.6(7)	95.5 ± 3.6(6)	96.9 ± 2.1(4)	90.7 ± 4.5(9)	**99.0 ± 1.2(1)**
Sonar	71.9 ± 4.1(8)	73.5 ± 5.9(5)	70.2 ± 5.0(9)	73.2 ± 3.3(6.5)	75.5 ± 4.1(4)	**76.0 ± 3.4(1.5)**	75.7 ± 4.9(3)	73.2 ± 3.8(6.5)	**76.0 ± 3.1(1.5)**
House	93.3 ± 2.8(6)	93.7 ± 2.2(4)	94.3 ± 3.2(3)	96.0 ± 2.12(2)	89.1 ± 8.3(8.5)	89.1 ± 8.3(8.5)	92.0 ± 4.7(7)	93.4 ± 2.9(5)	**96.7 ± 2.2(1)**
spectfheart	70.9 ± 6.4(8)	78.9 ± 2.3(6)	65.7 ± 4.4(9)	75.9 ± 3.2(7)	80.8 ± 2.9(3)	**81.4 ± 2.6(1)**	81.2 ± 3.2(2)	79.4 ± 2.0(5)	79.9 ± 1.8(4)
ionosphere	81.8 ± 4.1(7)	82.3 ± 4.4(6)	78.8 ± 4.3(8)	83.0 ± 3.7(5)	83.1 ± 4.5(4)	83.4 ± 3.8(3)	**84.5 ± 3.5(1)**	76.3 ± 3.9(9)	84.4 ± 4.4(2)
house-votes	91.3 ± 2.8(6)	92.1 ± 2.3(3)	91.2 ± 2.7(7)	91.8 ± 2.31(4)	90.3 ± 4.5(9)	90.5 ± 4.6(8)	91.6 ± 3.6(5)	92.4 ± 3.4(2)	**93.8 ± 1.6(1)**
WDBC	88.8 ± 5.3(8)	89.5 ± 5.3(7)	87.2 ± 6.2(9)	90.9 ± 4.78(6)	92.6 ± 4.9(3)	**92.7 ± 4.9(1.5)**	**92.7 ± 5.3(1.5)**	91.3 ± 6.8(5)	91.5 ± 4.6(4)
monk2	54.7 ± 5.1(9)	57.0 ± 7.4(7)	56.5 ± 5.7(8)	61.1 ± 5.9(3)	58.2 ± 5.3(6)	59.2 ± 5.8(4)	59.0 ± 5.3(5)	**62.9 ± 5.6(2)**	63.6 ± 5.0(1)
Breast	95.4 ± 1.4(4)	95.2 ± 2.8(6)	88.6 ± 4.9(9)	95.7 ± 1.5(2)	95.2 ± 3.1(6)	95.2 ± 3.1(6)	95.5 ± 3.1(3)	89.3 ± 1.4(8)	**95.9 ± 1.5(1)**
Diabetes	71.2 ± 3.1(7)	71.7 ± 2.9(6)	63.8 ± 4.3(9)	71.8 ± 3.1(5)	72.2 ± 2.3(3.5)	72.2 ± 2.8(3.5)	**73.1 ± 2.1(1.5)**	66.0 ± 2.5(8)	**73.1 ± 2.8(1.5)**
vehicle	70.0 ± 5.2(9)	71.9 ± 4.7(8)	72.8 ± 3.0(7)	73.7 ± 3.4(4)	73.2 ± 3.4(5.5)	73.2 ± 3.4(5.5)	73.8 ± 2.9(3)	75.7 ± 1.6(2)	**77.6 ± 2.2(1)**
german	61.1 ± 4.7(9)	62.8 ± 4.2(8)	64.6 ± 6.0(7)	68.4 ± 3.8(3.5)	68.4 ± 4.9(3.5)	68.2 ± 4.1(6)	68.3 ± 4.5(5)	70.5 ± 4.0(2)	**71.3 ± 1.4(1)**
BCI	55.6 ± 4.3(9)	56.2 ± 4.8(7)	61.5 ± 3.3(2)	60.9 ± 4.0(3)	57.0 ± 2.4(4.5)	57.0 ± 2.5(4.5)	56.3 ± 2.6(6)	55.8 ± 2.4(8)	**63.9 ± 4.5(1)**
USPS	80.9 ± 5.0(5)	79.8 ± 5.7(6)	83.7 ± 3.4(2.5)	83.7 ± 2.9(2.5)	77.5 ± 5.2(9)	77.8 ± 5.1(7)	77.6 ± 5.1(8)	81.0 ± 5.0(4)	**84.8 ± 2.2(1)**
digit1	83.0 ± 5.9(8)	83.6 ± 5.9(6)	82.5 ± 5.3(9)	83.3 ± 5.9(7)	**88.2 ± 4.3(1)**	87.4 ± 4.2(3)	87.8 ± 4.0(2)	86.7 ± 3.6(4)	84.0 ± 5.6(5)
DNA	73.5 ± 2.6(4)	74.0 ± 4.2(3)	72.9 ± 2.5(6)	76.4 ± 0.8(2)	68.5 ± 2.8(8)	68.7 ± 2.9(7)	68.2 ± 2.7(9)	73.8 ± 2.1(5)	**76.6 ± 0.8(1)**
iris	81.3 ± 5.2(7)	86.9 ± 5.5(2)	73.8 ± 5.1(8)	83.9 ± 5.0(5)	83.0 ± 7.4(6)	85.3 ± 8.4(4)	**87.5 ± 5.3(1)**	70.1 ± 5.5(9)	86.8 ± 3.8(3)
balance-scale	86.5 ± 4.0(4)	85.6 ± 5.0(7)	85.9 ± 3.0(6)	88.4 ± 2.1(2)	83.2 ± 4.6(9)	84.8 ± 3.4(8)	87.9 ± 2.6(3)	86.1 ± 3.2(5)	**89.1 ± 1.7(1)**
Avg. ACC	78.2	79.5	77.0	80.9	79.5	79.9	80.5	78.6	**82.7**
Avg. rank	6.8	5.6	7	4.0	5.6	4.9	3.9	5.5	**1.8**
Win/Tie/Loss against SVM		3/14/1	4/9/5	13/5/0	7/9/2	7/9/2	10/6/2	4/9/5	**16/2/0**
Win/Tie/Loss against LapSVM									14/4/0

RBF kernel

Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
wine	97.9 ± 1.9(6)	98.7 ± 2.2(2)	97.5 ± 2.2(7)	98.4 ± 1.9(5)	97.2 ± 3.2(8)	98.6 ± 2.1(3)	98.5 ± 1.8(4)	96.7 ± 1.7(9)	**99.3 ± 1.1(1)**
sonar	74.2 ± 3.9(6.5)	74.2 ± 4.0(6.5)	75.3 ± 5.2(4)	75.0 ± 4.2(5)	**76.8 ± 3.5(1)**	72.2 ± 3.4(8.5)	76.1 ± 4.3(3)	72.2 ± 3.4(8.5)	76.4 ± 4.2(2)
house	93.1 ± 2.4(4.5)	93.1 ± 2.7(4.5)	93.9 ± 2.2(2)	93.7 ± 2.2(3)	89.0 ± 6.5(9)	89.1 ± 13.1(8)	92.6 ± 3.3(6)	91.7 ± 2.5(7)	**94.2 ± 2.3(1)**
spectfheart	76.92 ± 3.3(8)	79.5 ± 1.6(6)	76.5 ± 4.1(9)	78.7 ± 3.2(7)	81.2 ± 2.6(3)	80.0 ± 2.1(4)	**82.1 ± 3.0(1)**	79.9 ± 2.0(5)	81.3 ± 2.3(2)
ionosphere	85.8 ± 5.5(5)	85.6 ± 5.4(6)	84.8 ± 2.8(8)	**87.6 ± 3.6(2)**	85.2 ± 5.7(7)	**87.6 ± 5.6(2)**	87.2 ± 5.4(4)	82.5 ± 6.4(9)	**87.6 ± 3.8(2)**
house-votes	90.7 ± 2.6(7)	92.0 ± 2.7(2)	91.5 ± 1.7(4.5)	91.2 ± 2.1(6)	90.3 ± 4.6(8)	88.8 ± 8.7(9)	91.5 ± 2.9(4.5)	91.7 ± 3.1(3)	**92.3 ± 2.3(1)**
WDBC	88.0 ± 6.0(9)	88.9 ± 5.7(7)	89.5 ± 6.0(4)	88.3 ± 6.8(8)	91.2 ± 5.5(3)	89.0 ± 6.7(6)	91.3 ± 6.0(2)	**91.5 ± 5.8(1)**	89.4 ± 6.7(5)
monk2	61.4 ± 3.9(9)	62.9 ± 3.9(6)	63.9 ± 4.9(4)	64.8 ± 3.8(2.5)	62.7 ± 3.8(7)	64.8 ± 4.5(2.5)	62.3 ± 4.0(8)	63.2 ± 4.7(5)	**67.7 ± 3.2(1)**
breast	95.3 ± 2.8(9)	95.5 ± 2.2(7)	95.7 ± 1.2(4.5)	95.7 ± 2.1(4.5)	95.4 ± 2.9(8)	95.6 ± 2.8(6)	95.9 ± 2.6(3)	96.2 ± 1.9(2)	**96.3 ± 1.5(1)**
diabetes	70.8 ± 3.6(7)	71.1 ± 2.6(6)	69.0 ± 4.1(8)	68.7 ± 3.6(9)	**72.5 ± 2.0(1.5)**	71.8 ± 3.6(4.5)	**72.5 ± 2.5(1.5)**	71.8 ± 2.5(4.5)	72.2 ± 2.8(3)
vehicle	72.7 ± 4.0(9)	73.0 ± 4.0(7)	74.5 ± 2.8(3.5)	74.5 ± 3.0(3.5)	72.9 ± 3.7(8)	75.1 ± 2.6(2)	73.4 ± 3.2(5)	73.3 ± 3.1(6)	**78.0 ± 2.0(1)**
german	67.6 ± 4.5(9)	68.6 ± 2.6(8)	70.5 ± 2.5(3)	70.1 ± 2.8(6)	70.1 ± 4.0(6)	70.4 ± 2.5(4)	70.1 ± 3.8(6)	70.7 ± 1.4(2)	**71.6 ± 1.2(1)**
BCI	54.1 ± 2.4(9)	54.3 ± 2.2(8)	57.2 ± 2.5(3.5)	57.2 ± 2.2(3.5)	56.0 ± 3.4(5)	57.4 ± 2.0(2)	55.5 ± 3.0(6)	55.3 ± 2.6(7)	**60.7 ± 3.5(1)**
USPS	83.2 ± 4.0(6)	83.0 ± 4.0(7)	**87.4 ± 3.8(1)**	86.2 ± 3.6(3)	80.8 ± 5.7(9)	84.6 ± 3.1(4)	81.2 ± 5.5(8)	84.3 ± 3.0(5)	86.8 ± 3.6(2)
digit1	83.2 ± 5.9(9)	84.6 ± 4.9(8)	**90.8 ± 6.2(1.5)**	89.5 ± 6.6(3)	88.8 ± 4.5(4)	86.6 ± 4.5(6)	88.7 ± 4.6(5)	85.2 ± 5.1(7)	**90.8 ± 5.7(1.5)**
DNA	76.1 ± 1.2(7)	72.0 ± 4.4(9)	**76.6 ± 0.8(3)**	**76.6 ± 0.9(3)**	76.4 ± 0.9(6)	**76.6 ± 0.9(3)**	**76.6 ± 0.9(3)**	74.5 ± 2.3(8)	**76.6 ± 0.9(3)**
iris	94.6 ± 4.0(5)	92.6 ± 5.5(6)	95.4 ± 1.4(3)	95.5 ± 3.5(2)	91.5 ± 4.6(8)	89.8 ± 4.1(9)	95.0 ± 3.9(4)	91.8 ± 4.1(7)	**95.9 ± 4.5(1)**
balance-scale	73.0 ± 3.7(7)	71.1 ± 9.4(8)	79.5 ± 2.7(5)	78.5 ± 3.1(6)	81.9 ± 4.6(4)	72.3 ± 9.8(9)	83.3 ± 3.3(2)	82.7 ± 3.6(3)	**84.3 ± 3.9(1)**
Avg. ACC	79.9	80.0	81.6	81.7	81.1	80.6	81.9	80.8	**83.4**
Avg. rank	7.3	6.3	4.4	4.6	5.9	5.1	4.2	5.5	**1.7**
Win/Tie/Loss against SVM		4/12/2	11/5/2	**13/5/0**	8/8/2	4/13/1	6/12/0	6/7/5	**13/5/0**
Win/Tie/Loss against LapSVM									8/10/0

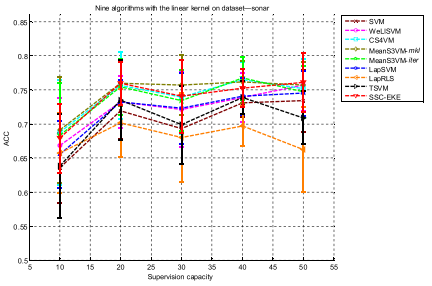
Table 5
Classification accuracies of all nine algorithms with the supervision size of 40.

Linear kernel

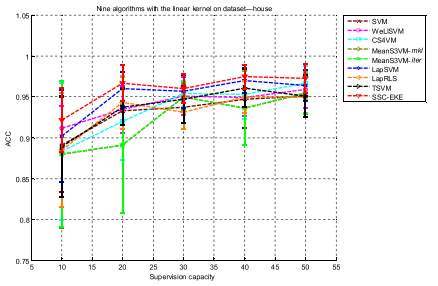
Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
wine	97.6 ± 0.8(5)	98.5 ± 1.1(3)	93.6 ± 1.8(8.5)	98.9 ± 0.8(2)	96.0 ± 5.9(7)	96.1 ± 5.9(6)	98.4 ± 1.2(4)	93.6 ± 2.0(8.5)	**99.6 ± 0.7** (1)
sonar	73.2 ± 2.2(8)	73.9 ± 2.4(6.5)	69.7 ± 2.9(9)	74.1 ± 2.7(5)	**76.8 ± 3.2(1)**	76.2 ± 3.1(3)	76.5 ± 2.6(2)	73.9 ± 3.6(6.5)	75.3 ± 2.8(4)
house	94.7 ± 3.5(7)	96.1 ± 2.4(3)	95.0 ± 2.0(5)	96.9 ± 1.6(2)	93.6 ± 4.5(8.5)	93.6 ± 4.5(8.5)	95.3 ± 3.1(4)	94.9 ± 2.2(6)	**97.5 ± 1.5(1)**
spectfheart	72.9 ± 6.5(8)	78.2 ± 2.0(7)	65.0 ± 5.4(9)	79.3 ± 1.9(5)	80.5 ± 1.5(3)	**81.2 ± 1.8(1)**	81.0 ± 1.5(2)	79.1 ± 1.1(6)	80.2 ± 1.9(4)
ionosphere	83.6 ± 6.4(7)	83.7 ± 6.3(6)	80.2 ± 3.9(8)	84.7 ± 5.4(4)	84.1 ± 5.8(5)	85.3 ± 4.9(3)	**85.5 ± 6.6(1.5)**	80.0 ± 5.6(9)	**85.5 ± 5.2(1.5)**
house-votes	90.9 ± 2.1(9)	91.5 ± 1.9(8)	93.0 ± 1.5(6.5)	94.2 ± 1.5(2)	93.0 ± 1.2(6.5)	93.2 ± 1.3(4.5)	93.2 ± 1.2(4.5)	93.5 ± 1.4(3)	**95.1 ± 1.4(1)**
WDBC	93.7 ± 2.9(5)	93.8 ± 3.3(4)	93.0 ± 3.5(7.5)	94.8 ± 2.5(2)	93.1 ± 5.7(6)	93.0 ± 5.5(7.5)	94.5 ± 3.2(3)	91.5 ± 6.8(9)	**95.5 ± 2.5(1)**
monk2	63.8 ± 3.1(3)	60.1 ± 3.7(8)	57.9 ± 7.8(9)	64.7 ± 1.8(2)	60.6 ± 2.4(7)	62.4 ± 2.6(6)	63.2 ± 3.3(4)	62.7 ± 5.8(5)	**66.7 ± 2.7(1)**
breast	96.3 ± 0.9(4)	96.6 ± 0.9(2)	89.7 ± 3.5(8)	96.4 ± 0.9(3)	94.2 ± 4.7(6)	94.1 ± 4.9(7)	95.2 ± 3.3(5)	87.8 ± 4.0(9)	**96.6 ± 0.8(1)**
diabetes	73.6 ± 2.2(7)	74.1 ± 2.7(6)	65.0 ± 2.2(9)	74.8 ± 1.7(5)	75.2 ± 1.5(4)	75.3 ± 1.2(3)	75.4 ± 1.4(2)	66.4 ± 1.4(8)	**75.5 ± 1.9(1)**
vehicle	74.9 ± 1.3(8)	73.9 ± 2.0(9)	76.8 ± 1.9(2)	76.4 ± 1.8(4)	75.1 ± 3.7(6.5)	75.1 ± 2.7(6.5)	76.7 ± 2.1(3)	75.2 ± 1.4(5)	**77.8 ± 1.9(1)**
german	63.4 ± 2.8(8)	64.5 ± 2.3(7)	63.8 ± 3.2(9)	70.0 ± 2.8(3)	68.8 ± 2.4(5)	68.8 ± 2.5(5)	68.8 ± 2.6(5)	70.4 ± 1.8(2)	**71.2 ± 1.8(1)**
BCI	61.1 ± 4.1(7)	60.8 ± 4.2(8)	64.4 ± 2.8(3)	64.9 ± 3.2(2)	61.3 ± 3.4(6)	61.6 ± 3.6(5)	62.4 ± 4.2(4)	58.7 ± 3.0(9)	**67.2 ± 2.7(1)**
USPS	82.0 ± 3.6(8)	80.7 ± 3.4(9)	84.3 ± 2.2(4)	85.1 ± 1.3(2)	83.0 ± 2.7(7)	83.1 ± 2.8(6)	83.4 ± 2.7(5)	84.5 ± 2.7(3)	**86.0 ± 1.0(1)**
digit1	86.3 ± 2.8(8)	86.6 ± 3.7(7)	84.5 ± 4.2(9)	86.9 ± 3.2(6)	88.1 ± 5.1(4)	88.5 ± 5.1(2)	88.2 ± 5.2(3)	**88.9 ± 5.9(1)**	87.8 ± 3.2(5)
DNA	71.4 ± 3.2(5)	76.2 ± 1.3(3)	71.1 ± 3.2(6)	76.5 ± 0.7(2)	68.8 ± 3.1(8)	68.9 ± 3.3(7)	68.6 ± 3.2(9)	74.3 ± 2.6(4)	**76.7 ± 0.7(1)**
iris	81.4 ± 7.6(7)	88.9 ± 5.0(4)	76.4 ± 5.9(8)	83.4 ± 7.1(6)	88.5 ± 5.0(5)	89.5 ± 5.7(3)	**91.6 ± 4.1(1)**	71.9 ± 7.4(9)	89.6 ± 5.2(2)
balance-scale	87.2 ± 4.4(8)	85.1 ± 4.2(9)	87.6 ± 2.9(6.5)	89.1 ± 2.3(3)	87.6 ± 2.7(6.5)	88.8 ± 2.3(4)	89.3 ± 2.4(2)	88.5 ± 2.3(5)	**90.3 ± 1.9(1)**
Avg. ACC	80.4	81.3	78.4	82.8	81.6	81.9	82.6	79.8	**84.1**
Avg. rank	6.8	6.1	7.1	3.3	5.7	4.9	3.6	6	**1.6**
Win/Tie/Loss against SVM		4/13/1	4/5/9	14/4/0	5/11/2	6/11/1	9/8/1	6/7/5	**18/0/0**
Win/Tie/Loss against LapSVM									13/5/0

RBF kernel

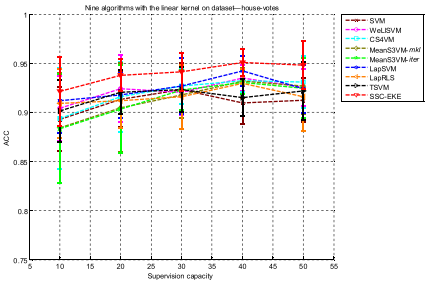
Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
wine	99.1 ± 0.8(5)	99.2 ± 0.7(4)	98.7 ± 0.8(6)	99.5 ± 0.5(2.5)	97.8 ± 3.4(8)	98.8 ± 3.1(7)	99.5 ± 0.7(2.5)	97.7 ± 1.9(9)	**99.7 ± 0.4(1)**
sonar	75.9 ± 3.3(6)	75.8 ± 3.0(7)	78.2 ± 3.2(2)	77.9 ± 3.2(4)	77.3 ± 3.8(5)	72.5 ± 3.2(9)	78.0 ± 3.5(3)	75.6 ± 5.0(8)	**79.4 ± 3.8(1)**
house	93.7 ± 2.9(7)	94.4 ± 3.3(6)	95.1 ± 2.7(3)	94.7 ± 3.2(4.5)	92.6 ± 4.5(8)	**96.0 ± 2.2(1.5)**	94.7 ± 3.1(4.5)	91.8 ± 2.9(9)	**96.0 ± 2.4(1.5)**
spectfheart	78.4 ± 2.4(9)	79.1 ± 2.2(8)	80.2 ± 2.9(6)	80.7 ± 3.9(5)	81.2 ± 2.0(3.5)	79.5 ± 1.8(7)	**82.2 ± 1.5(1.5)**	81.2 ± 1.9(3.5)	**82.2 ± 2.4(1.5)**
ionosphere	89.0 ± 5.4(5)	88.7 ± 6.3(8)	88.1 ± 3.6()	91.7 ± 4.2(2)	88.9 ± 6.2(7)	90.7 ± 4.8(3)	90.1 ± 6.1(4)	85.4 ± 7.9(9)	**91.8 ± 4.2(1)**
house-votes	92.0 ± 2.0(6.5)	92.0 ± 2.0(6.5)	93.0 ± 1.6(4)	93.4 ± 2.0(2.5)	92.5 ± 1.7(5)	91.5 ± 4.8(9)	93.4 ± 1.9(2.5)	91.6 ± 1.7(8)	**93.7 ± 1.5(1)**
WDBC	92.4 ± 2.7(4.4)	92.5 ± 3.6(3)	90.8 ± 2.9(8)	91.8 ± 2.3(6)	91.5 ± 4.0(7)	90.4 ± 6.2(9)	92.6 ± 3.9(2)	92.4 ± 6.5(4.5)	**93.8 ± 2.9(1)**
monk2	64.5 ± 2.4(8)	64.6 ± 3.5(7)	67.1 ± 3.1(3)	67.9 ± 3.2(2)	64.4 ± 2.8(9)	65.1 ± 3.6(4.5)	64.8 ± 3.0(6)	65.1 ± 4.5(4.5)	**69.3 ± 2.5(1)**
breast	96.2 ± 0.8(2)	96.1 ± 1.2(3.5)	95.5 ± 1.3(6.5)	96.0 ± 1.3(5)	94.4 ± 5.3(8.5)	94.4 ± 5.2(8.5)	95.5 ± 3.8(6.5)	96.1 ± 1.1(3.5)	**96.7 ± 1.1(1)**
diabetes	72.6 ± 2.9(5.5)	72.6 ± 2.6(5.5)	69.2 ± 2.9(9)	71.7 ± 2.4(7)	73.6 ± 1.9(3)	71.1 ± 2.9(8)	73.5 ± 2.2(4)	74.2 ± 2.0(2)	**74.4 ± 1.8(1)**
vehicle	74.5 ± 1.6(8)	74.4 ± 5.2(9)	75.1 ± 2.1(7)	75.9 ± 1.4(2)	75.3 ± 2.9(5.5)	75.6 ± 1.8(4)	75.8 ± 2.4(3)	75.3 ± 1.7(5.5)	**76.9 ± 1.8(1)**
german	68.9 ± 2.9(9)	69.6 ± 5.2(8)	69.7 ± 2.8(6)	69.7 ± 2.6(6)	69.7 ± 1.9(6)	70.4 ± 1.8(3)	70.1 ± 2.5(4)	**71.5 ± 1.9(1)**	71.3 ± 1.9(2)
BCI	57.0 ± 4.0(8)	57.8 ± 3.9(5.5)	59.9 ± 3.7(2)	59.6 ± 3.7(3)	57.8 ± 2.1(5.5)	57.4 ± 4.0(7)	58.2 ± 2.5(4)	56.7 ± 2.8(9)	**62.8 ± 4.4(1)**
USPS	85.0 ± 2.7(9)	85.5 ± 2.7(8)	**89.2 ± 3.5** (1)	88.5 ± 3.3(3)	85.6 ± 2.7(7)	86.6 ± 1.8(5)	86.4 ± 2.7(6)	87.3 ± 2.6(4)	89.0 ± 3.5(2)
digit1	87.7 ± 4.6(8)	87.9 ± 2.6(7)	93.0 ± 4.1(2)	92.8 ± 4.4(3)	89.2 ± 5.6(4.5)	88.6 ± 6.9(6)	89.2 ± 5.7(4.5)	88.9 ± 6.0(9)	**93.7 ± 3.2(1)**
DNA	76.6 ± 0.7(6.5)	76.5 ± 0.8(8.5)	76.7 ± 0.7(4.5)	76.7 ± 0.7(4.5)	76.5 ± 0.7(8.5)	76.6 ± 0.7(6.5)	76.7 ± 0.7(4.5)	76.7 ± 0.8(4.5)	**76.8 ± 0.7(1)**
iris	96.6 ± 2.4(5)	95.9 ± 2.3(8)	97.0 ± 2.1(4)	97.8 ± 1.9(2.5)	96.4 ± 2.2(6)	96.0 ± 3.3(7)	97.8 ± 1.9(2.5)	95.0 ± 3.0(9)	**98.1 ± 1.9(1)**
balance-scale	81.2 ± 5.7(7)	79.5 ± 5.3(8)	84.3 ± 3.2(6)	84.6 ± 3.2(5)	86.9 ± 2.2(4)	78.4 ± 8.5(9)	87.2 ± 2.1(3)	**87.5 ± 2.4(1.5)**	**87.5 ± 2.6(1.5)**
Avg. ACC	82.3	82.4	83.4	83.9	82.9	82.2	83.7	82.8	**85.2**
Avg. rank	6.6	6.7	4.7	3.9	6.2	6.3	3.8	5.8	**1.2**
Win/Tie/Loss against SVM		2/16/0	7/9/2	11/7/0	2/16/0	2/15/1	7/11/0	4/11/3	**15/3/0**
Win/Tie/Loss against LapSVM									7/11/0



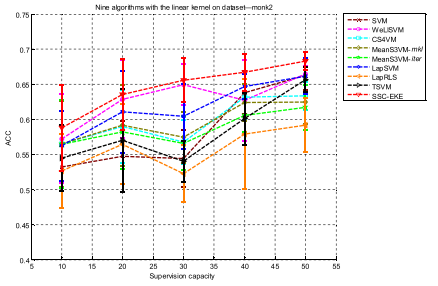
(a) On *sonar*



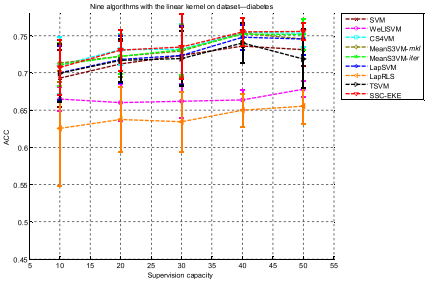
(b) On *house*



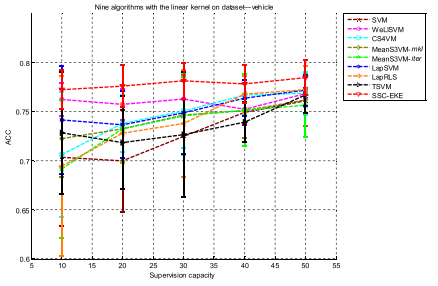
(c) On *house-votes*



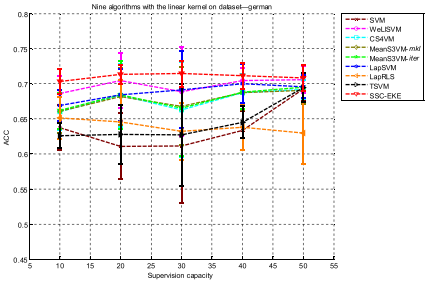
(d) On *monk2*



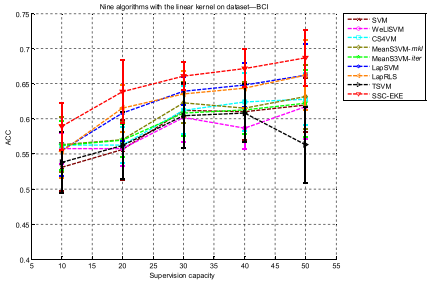
(e) On *diabetes*



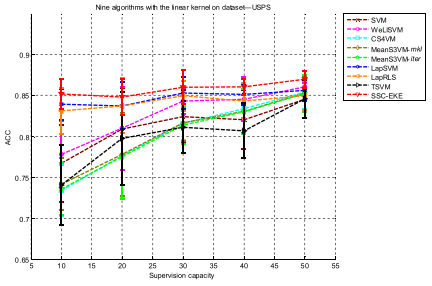
(f) On *vehicle*



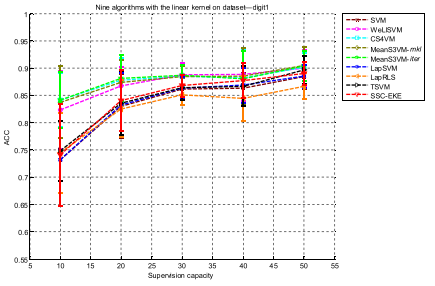
(g) On *german*



(h) On *BCI*

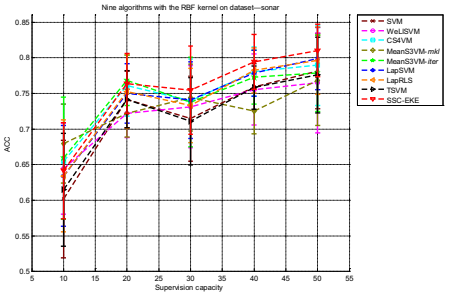


(i) On *USPS*

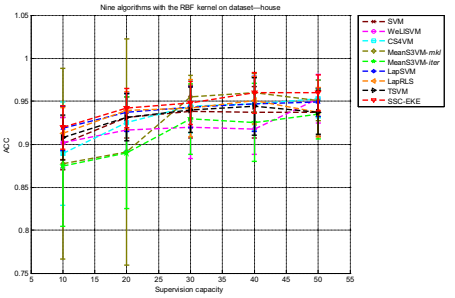


(j) On *digit1*

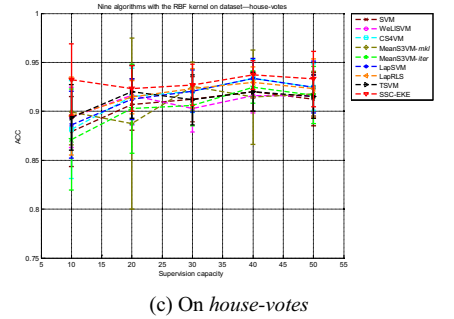
Fig. 5. Performance curves of the nine algorithms with respect to the varied labelled data sizes on partial Benchmark/UCI/KEEL semi-supervised data sets with the linear kernel.



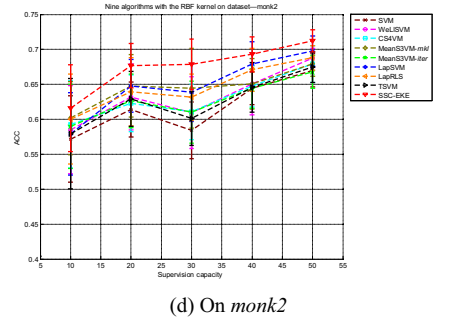
(a) On *sonar*



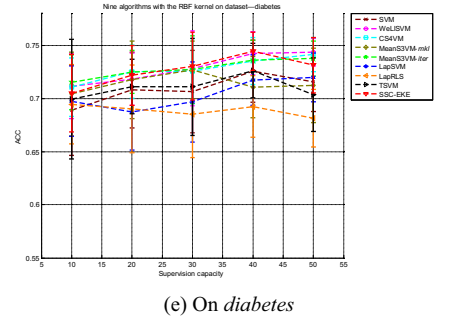
(b) On *house*



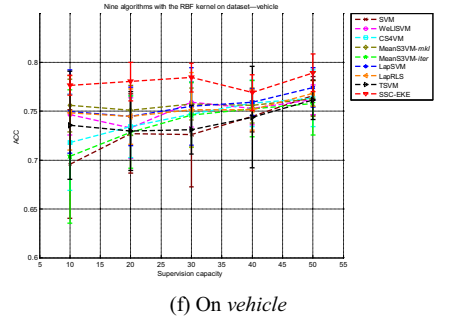
(c) On *house-votes*



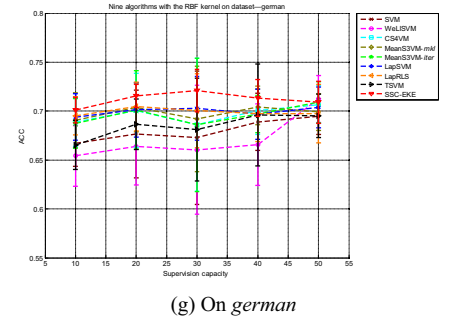
(d) On *monk2*



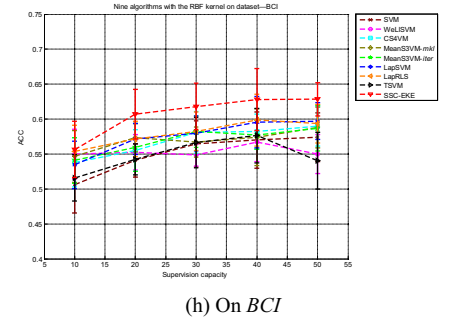
(e) On *diabetes*



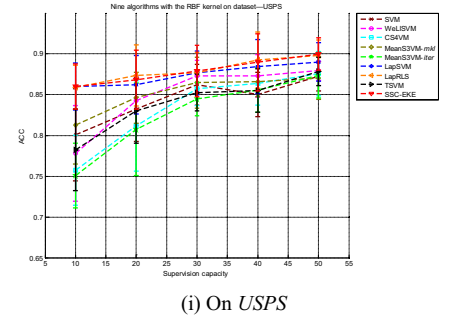
(f) On *vehicle*



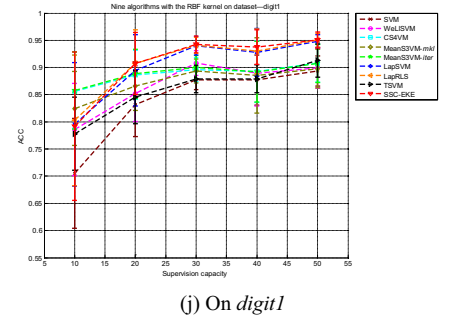
(g) On *german*



(h) On *BCI*



(i) On *USPS*



(j) On *digit1*

Fig. 6. Performance curves of the nine algorithms with respect to the varied labelled data sizes on partial Benchmark/UCI/KEEL semi-supervised data sets with the RBF kernel.

highest F1 scores in both the positive and negative classes of all data sets. On one data set, if one algorithm achieves the highest F1 scores in both the positive and negative classes, it certainly achieves the best ACC. This is the reason why our SSC-EKE outperforms the others.

2) As one of the theoretical bases of our research, the classification performance of the LapSVM on these data sets is also compared with that of our SSC-EKE algorithm. As shown in Table 4, in which the sample size for the supervision is 20, the win/tie/loss counts of SSC-EKE against the LapSVM are 14/4/0 and 8/10/0 regarding the linear and RBF kernels, respectively. This indicates that SSC-EKE overcomes the LapSVM overwhelmingly in the case of the linear kernel, and under the condition of not being defeated, the former outperforms the latter on nearly half of the recruited data sets in the case of the RBF kernel. For the experimental results with the supervision size of 40, as listed in Table 5, the superiority of our SSC-EKE versus the LapSVM appears to be more substantial than in the case with the sample size of 20.
3) As indicated in Figs. 5 and 6, the overall classification effectiveness of SSC-EKE is roughly positively proportional to the supervision capacity. Specifically, as the labelled sample size increases, the number of must-link/cannot-link constraints increases accordingly; consequently, this strengthens the efficacy of the MPCJRF in the form of (21) with respect to the whole framework of SSC-EKE (see (22)).
4) The validity of the MPCJRF in SSC-EKE sometimes cannot be manifested when the supervision capacity is too small, such as in the cases of data sets with 10 labelled examples. For example, on the *sonar* and *digit1* data sets with the RBF kernel, as indicated in Fig. 6(a) and 6(j), the accuracies of SSC-EKE are distinctly less than those of some competitors with the sample size of 10, whereas SSC-EKE ranks first when the sample sizes are 30, 40, and 50. The reason is that the MPCJRF cannot obtain relatively sufficient information from the pairwise constraints when the number of labelled examples is quite small.
5) It is worth further discussing the outcomes of SSC-EKE on the *digit1* data set. Here, we notice two phenomena: i) As illustrated in Fig. 5(j), in the case of the linear kernel, neither SSC-EKE nor the LapSVM achieves a desirable rank, whereas other approaches relying on label means, such as CS4VM and MeanS3VMs, obtain considerably better scores. ii) As illustrated in Figs. 5(j) and 6(j), the advantage of SSC-EKE over the LapSVM in terms of the average ACC are nearly unobservable, despite the increase of the supervision data sizes from 10 to 50 in both cases of the linear and RBF kernels. This implies that the MPCJRF did not play the due role in the entire framework of SSC-EKE in these cases. We believe that these phenomena occurred due to the data inconsistency existing in the original data set. In the *digit1* data set, there is much interference information, e.g., mislabelled data or data pollution due to noise, and this results in its non-linear separateness. Therefore, the classification accuracies of SSC-EKE and the LapSVM with the linear kernel are distinctly worse than those with the RBF kernel (see Figs. 5(j) and 6(j)). Moreover, our proposed MPCJRF mechanism is shown to depend on the data purity in the supervision subset, i.e., only correct labels can offer us beneficial must-link/cannot-link constraints. Conversely, the mistakes in the given labelled examples negatively impact the entire performance of SSC-EKE.

4.4. Experiments on real-world data sets

For the purpose of further verifying the realistic performance of the proposed SSC-EKE, we have also conducted our experiments in three real-world data scenarios: text data classification, image recognition, and handwritten digit recognition. To this end, the well-established *20 Newsgroups text database*[4] [8], *Object Categories image repository*[5] [26], *NIPS 2003 feature selection database*,[6] and *MNIST handwritten digit database*[7] were used in our experiments. The constructions regarding the data sets used in this subsection are as follows.

1) For the text data classification scenario, four major text categories in the *20 Newsgroups text database*, i.e., *comp, rec, sci*, and *talk*, were used. We generated the six text data sets, which are shown in Table 6, by using all possible pairwise combinations among these categories. Each data set had 1000 records by randomly selecting 250 records from each category. Each data set was evenly divided into two parts to generate the training and testing sets. To construct the (semi-)supervised classification scenes, we further randomly subsampled each training set 20 times, using the sample size of 10, to produce 20 inconsistent subsets as the supervision data. In addition, the BOW toolkit [23] was used to reduce the data dimension, as it was originally as high as 43,586. The details of these data sets used in our (semi-)supervised classification experiments are summarized in Table 7.
2) For the image recognition scenario, as also indicated in Table 6, two pairs of image categories from the *Object Categories image repository* were used in our experiments: *coast VS highway* and *mountain VS street*. The number of images contained in the categories of *coast, highway, mountain*, and *street* are 360, 260, 374, and 292, respectively, and the total data sizes of *coast VS highway* and *mountain VS street* are, respectively, 620 and 666. Eight representative examples of each of the four image categories are shown in Fig. 7. We randomly selected 300 images from each data set for training and the remainder for testing. We further subsampled each training data set 20 times to obtain the supervision data with the

[4] http://www.cs.nyu.edu/~roweis/data.html.
[5] http://www.vision.caltech.edu/feifeili/Datasets.htm.
[6] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html (or http://www.clopinet.com/isabelle/Projects/NIPS2003/).
[7] http://yann.lecun.com/exdb/mnist/.

Table 6
Compositions of categories of 10 real-world data sets.

Scenario	Dataset	Composition		Data Size	
Text data classification	*comp VS rec*	+	comp.sys.ibm.pc.hardware	250	1000
			comp.sys.mac.hardware	250	
		−	rec.sport.baseball	250	
			rec.sport.hockey	250	
	comp VS sci	+	comp.sys.ibm.pc.hardware	250	1000
			comp.sys.mac.hardware	250	
		−	sci.med	250	
			sci.space	250	
	comp VS talk	+	comp.sys.ibm.pc.hardware	250	1000
			comp.sys.mac.hardware	250	
		−	talk.politics.guns	250	
			talk.politics.misc	250	
	rec VS sci	+	rec.sport.baseball	250	1000
			rec.sport.hockey	250	
		−	sci.med	250	
			sci.space	250	
	rec VS talk	+	rec.sport.baseball	250	1000
			rec.sport.hockey	250	
		−	talk.politics.guns	250	
			talk.politics.misc	250	
	sci VS talk	+	sci.med	250	1000
			sci.space	250	
		−	talk.politics.guns	250	
			talk.politics.misc	250	
Image recognition	*coast VS highway*	+	coast	360	620
		−	highway	260	
	mountain VS street	+	mountain	374	666
		−	street	292	
Handwritten digit recognition	*gisette_4 VS 9*	+	handwritten digit '4'	3500	7000
		−	handwritten digit '9'	3500	
	mnist_3 VS 8	+	handwritten digit '3'	7141	13,966
		−	handwritten digit '8'	6825	

Table 7
Details of ten, real-world, (semi-)supervised classification data sets.

Dataset	Training set		Testing set data size	Dimension
	Data size	Labelled example size		
comp VS rec	500	50	500	318
comp VS sci	500	50	500	358
comp VS talk	500	50	500	255
rec VS sci	500	50	500	242
rec VS talk	500	50	500	297
sci VS talk	500	50	500	333
coast VS highway	300	30	320	300
mountain VS street	300	30	366	300
gisette_4 VS 9	4000	100	3000	5000
mnist_3 VS 8	6000	100	7966	784

sample size of 10%. Because the number of pixels in each image, i.e., $256 \times 256 = 65{,}536$, is too large to be directly used as the data features, we performed the principal component analysis (PCA) to reduce the input feature dimensionality to 300.

3) For the handwritten digit recognition scenario, in a case that we term *gisette_4 VS 9*, the *gisette* data set from the *NIPS 2003 feature selection database* was used to test the ability to distinguish the handwritten digits '4' and '9', which are often confused for each other. As detailed in Tables 6 and 7, there are 3500 records related to these two digits in *gisette_4 VS 9*, and the data dimension can be as high as 5000. We randomly selected 4000 records as the training set, using 100 arbitrarily selected examples as the labelled data, and used the remainder as the testing set. Similarly, in a test that we denote as *mnist_3 VS 8*, we extracted all records containing the digits '3' and '8' from the well-known *MNIST handwritten digit database* and tested the ability to differentiate these two digits. As shown in Tables 6 and 7, the feature dimension of *mnist_3 VS 8* is 784, and the total number of records is 13,966, of which 6000 were used for training and the remainder for testing. One hundred randomly selected examples in the training subset were used as the supervision information. In contrast to our other experiments, we did not reduce the original data dimensions, as here we attempted to investigate the classification performance of all competitors against high-dimensional data.

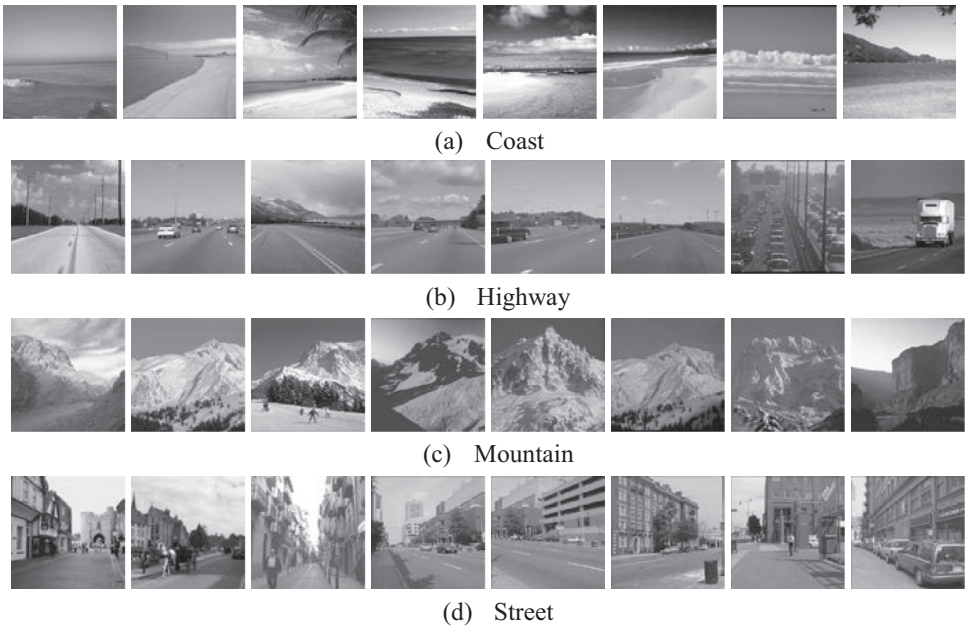


Fig. 7. Illustration of the image categories involved in our experiments.

For each of these real-world, semi-supervised data sets, the performances of all nine algorithms were tested using twenty inconsistent supervision subsets. The outcomes of these algorithms, reported in terms of the ACC means and standard deviations, are listed in Table 8. Due to the limited manuscript length, Table 8 only lists the individual scores of all candidates and the statistical results of the paired *t*-test associated with the RBF kernel. The F1 scores of the nine algorithms on partial real-world data sets are presented in Table A.4 in the Appendix. The results of these tests generally show the performance advantage of our SSC-EKE algorithm, which is consistent with the findings observed from the Benchmark/UCI/KEEL (semi-)supervised data sets shown in the previous subsection. In addition, despite the fact that both *gisette_4 VS 9* and *mnist_3 VS 8* belong to high-dimensional data sets, our SSC-EKE algorithm also ranks as the best of all the algorithms. These results, along with those in the previous subsections, confirm the effectiveness of our proposed SSC-EKE method. Benefiting from the MPCJRF developed in (21), the knowledge embedded in both the few, precious labelled examples and plenty of unlabelled data instances are concurrently, extensively exploited. This exploitation consequently facilitates the preferable classification performance of SSC-EKE.

4.5. Computational time comparisons

To compare the computational time of all employed algorithms, we also recorded their running time including both training and testing on all involved data sets. To reduce the manuscript length, their average running time on nine Benchmark/UCI/KEEL data sets with the RBF kernel and 20 labelled examples and on three larger-scale data sets, i.e., *DS1, gisette_4 VS 9*, and *mnist_3 VS 8*, with 100 labelled examples are shown in Table 9. As disclosed, the running time of the conventional SVM slightly varied on these data sets regardless whether they were small or large, as it only uses a few labelled examples to train the classifiers. The TSVM is generally the most time-consuming algorithm due to its iterative trials on testing data points. Specifically, after initially assigning the labels to all testing data points, the TSVM tried to iteratively correct the assigned labels of any two points if their assigned labels violated the predicted ones until the termination of the iterations. Therefore, the TSVM is clearly unsuitable for large-scale data sets. Although the WeLlSVM also assigned the labels for all unlabelled data points during the training of classifiers, using the cutting plane-based label generation strategy, this issue can be solved via a sequence of SVM subproblems that are more scalable than conventional convex semi-definite programming (SDP) relaxations. This facilitates the overall, much shorter computational time of the WeLlSVM compared with that of the TSVM. The MeanS3VM-*iter*, MeanS3VM-*mkl*, and CS4VM are three time-saving algorithms, as they only use the means of classes of unlabelled data instead of the unlabelled data themselves to constitute their formulations derived from the S3VM. Their computing efficiencies are particularly manifested on larger data sets, such as *DS1, gisette_4 VS 9*, and *mnist_3 VS 8* in our experiments. Both the LapRLS and LapSVM are manifold learning-based S3VM methods. Whereas the LapRLS has an analytical solution that avoids time-consuming quadratic programming computations that commonly occur in the LapSVM, the former is generally faster than the latter. Compared with the LapSVM, our proposed SSC-EKE algorithm has one more regularization term in its objective function, i.e., the pairwise constraint regularization. Therefore, in theory, the computational cost of SSC-EKE should be higher than that of the LapSVM. Our experimental results agree with this supposition on

Table 8
ACCs and statistical results of the paired t-test of all candidates on real-world, (semi-)supervised data sets (with RBF kernel).

Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLlSVM	SSC-EKE
comp VS rec	73.0 ± 5.4(7)	76.7 ± 2.5(2)	73.0 ± 5.7(6)	75.9 ± 3.2(3)	73.3 ± 5.7(5)	71.0 ± 3.4(9)	74.9 ± 6.3(4)	71.8 ± 6.8(8)	**77.4 ± 2.5(1)**
comp VS sci	69.0 ± 2.3(7)	70.3 ± 2.5(4)	68.8 ± 3.4(8)	71.1 ± 3.2(2)	69.6 ± 3.5(5.5)	67.6 ± 6.0(9)	70.9 ± 3.5(3)	69.6 ± 5.3(5.5)	**71.8 ± 3.8(1)**
comp VS talk	72.5 ± 4.1(8)	**76.7 ± 3.2(1)**	74.5 ± 4.0(6)	75.4 ± 4.6(4)	74.3 ± 4.2(7)	72.4 ± 5.9(9)	75.3 ± 4.6(5)	76.2 ± 4.5(3)	76.6 ± 3.7(2)
rec VS sci	66.1 ± 3.0(8)	67.2 ± 2.8(4)	66.5 ± 1.7(6)	68.9 ± 3.0(2)	66.1 ± 3.7(7)	63.7 ± 3.8(9)	66.6 ± 3.4(5)	68.2 ± 4.3(3)	**70.1 ± 3.4(1)**
rec VS talk	70.3 ± 4.0(6)	70.7 ± 4.1(4)	68.6 ± 4.0(9)	70.6 ± 4.9(5)	69.6 ± 3.5(7)	69.2 ± 7.0(8)	70.8 ± 4.2(3)	72.0 ± 4.6(2)	**72.5 ± 4.9(1)**
sci VS talk	64.3 ± 5.9(8)	66.6 ± 4.1(4.5)	65.6 ± 4.6(6)	67.3 ± 4.5(3)	66.6 ± 5.2(4.5)	62.9 ± 6.8(9)	66.9 ± 5.3(2)	65.2 ± 6.2(7)	**68.3 ± 4.6(1)**
coast VS highway	57.3 ± 3.2(8)	57.8 ± 3.6(7)	59.5 ± 2.6(3.5)	59.1 ± 4.0(5)	59.5 ± 3.2(3.5)	58.7 ± 3.2(6)	59.8 ± 3.2(2)	56.5 ± 7.6(9)	**60.4 ± 4.7(1)**
mountain VS street	63.0 ± 4.5(8)	63.7 ± 5.0(7)	65.2 ± 4.0(2.5)	64.9 ± 3.4(4)	65.2 ± 4.9(2.5)	62.2 ± 6.0(9)	64.5 ± 5.6(6)	64.7 ± 5.2(5)	**65.3 ± 4.1(1)**
gisette_4 VS 9	90.9 ± 2.1(9)	91.3 ± 2.1(6)	91.4 ± 1.9(4)	91.5 ± 1.9(2)	91.1 ± 2.2(7.5)	91.1 ± 1.7(7.5)	91.4 ± 1.5(4)	91.4 ± 1.6(4)	**92.8 ± 1.6(1)**
mnist_3 VS 8	94.5 ± 0.5(5)	93.2 ± 0.5(9)	95.0 ± 1.0(2)	94.9 ± 1.1(3.5)	93.4 ± 2.3(8)	93.6 ± 1.5(7)	94.9 ± 1.3(3.5)	94.0 ± 2.4(6)	**95.8 ± 1.0(1)**
Avg. ACC	72.09	73.42	72.81	73.96	72.87	71.24	73.6	72.96	**75.1**
Avg. rank	7.4	4.85	5.3	3.35	5.75	8.25	3.75	5.25	**1.1**
Win/Tie/Loss against SVM		3/6/1	5/5/0	6/4/0	3/7/0	0/10/0	4/6/0	4/6/0	**9/1/0**
Win/Tie/Loss against LapSVM									6/4/0

Table 9
Average running time (in seconds) of all algorithms on partial data sets (with RBF kernel).

Dataset	SVM	TSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	LapRLS	LapSVM	SSC-EKE
wine	0.02	0.94	0.11	0.18	0.17	0.05	0.001	0.02	0.07
house	0.06	3.31	0.09	0.20	0.18	0.04	0.01	0.04	0.08
spectfheart	0.01	1.13	0.14	0.17	0.17	0.05	0.01	0.02	0.07
WDBC	0.01	17.11	0.15	0.20	0.22	0.12	0.02	0.05	0.10
breast	0.01	63.01	0.09	0.20	0.24	0.15	0.03	0.09	0.13
diabetes	0.01	28.71	0.12	0.20	0.21	0.35	0.05	0.12	0.14
german	0.01	21.08	0.12	0.18	0.22	0.09	0.06	0.15	0.18
USPS	0.01	85.38	0.32	0.40	0.51	0.39	0.26	0.71	0.54
digit1	0.01	97.71	0.26	0.36	0.53	0.42	0.25	0.75	0.55
DS1	0.27	1.68E4	10.89	19.46	18.95	107.95	33.22	198.31	136.77
gisette_4 VS 9	0.17	1290.73	15.63	19.00	17.52	16.90	20.62	57.40	40.10
mnist_3 VS 8	0.17	2218.18	11.17	18.53	13.63	11.50	17.75	86.80	58.80

most of the involved data sets. However, on some larger data sets, e.g., data sets of which the numbers of training examples are more than 1000, SSC-EKE surpasses the LapSVM with respect to running time. The potential reason is probably due to the delicate MPCJRF in the form of (21), in which the values of both the manifold learning and pairwise constraint terms are transformed such that they have the same range; this could eventually benefit the optimization problem in the form of (23), especially for larger-scale data sets.

5. Conclusions

Our research is motivated by the lack of knowledge exploitation regarding few but valuable labelled examples in many existing S3VMs. To address this problem, the PCRF is devised by converting the given data labels into many pairwise constraints. Subsequently, by merging the PCRF with the manifold regularization term and converting their individual values such that they have the same range, the MPCJRF is further developed. Key to our SSC-EKE method is the systematic incorporation of empirical risk minimization, regularization in the RKHS, joint manifold and pairwise constraint-based regularization, graph Laplacian, etc., in which the connotation of extensive knowledge exploitation is embodied. Compared with several other state-of-the-art S3VM approaches on many semi-supervised data sets, the proposed SSC-EKE algorithm demonstrates preferable classification accuracy as well as generalizability.

Regarding our future work, we plan to investigate the countermeasures for our SSC-EKE on large-scale data sets. In this regard, the strategy regarding the core vector machine [27,34,35] could be one of the countermeasures tested. Also worthy of further study are the practicable methodologies for prompt, self-adaptive parameter setting in SSC-EKE, which could facilitate its applicability to real-world problems.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61772241 and 61702225, by the Natural Science Foundation of Jiangsu Province under Grant BK20160187, and by the Fundamental Research Funds for the Central Universities under Grant JUSRP51614A. Research reported in this publication was also supported by the National Cancer Institute of the National Institutes of Health, USA, under award number R01CA196687. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, USA.

In addition, we would like to thank Bonnie Hami, MA (USA) for her editorial assistance in the preparation of this manuscript.

Appendix

A. Tables

Table A.1
F1 scores of all nine algorithms on the synthetic data set *DS*1.

Dataset	SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
*DS*1 F1_+	0.8645	0.8814	0.9456	0.9147	0.8651	0.8133	0.9193	0.9363	**0.9650**
F1_−	0.8810	0.8914	0.9482	0.8992	0.8738	0.8292	0.9285	0.9380	**0.9645**

Table A.2
F1 scores of all nine algorithms on partial binary classification Benchmark/UCI/KEEL data sets with 20 labelled examples.

Linear kernel

Dataset		SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
wine	F1_+	0.9523	0.96	0.9206	0.9655	0.9407	0.9404	0.9558	0.8767	**0.9852**
	F1_−	0.9755	0.9795	0.9556	0.9826	0.9659	0.966	0.9775	0.9298	**0.9935**
sonar	F1_+	0.7063	0.7015	0.686	0.7236	0.7256	0.7258	0.7185	0.6853	**0.7428**
	F1_−	0.7394	0.7743	0.7437	0.7468	0.7897	**0.793**	0.7903	0.7718	0.7788
house	F1_+	0.9351	0.9377	0.9563	0.9601	0.9062	0.9067	0.9259	0.9355	**0.9663**
	F1_−	0.9319	0.9374	0.9576	0.9596	0.891	0.891	0.9178	0.9345	**0.9671**
spectfheart	F1_+	**0.3793**	0.046	0.2799	0.1884	0.2583	0.3192	0.3759	0.016	0.1272
	F1_−	0.8125	0.8813	0.8413	0.8597	0.8906	**0.8927**	0.8909	0.8848	0.8859
ionosphere	F1_+	0.7168	0.7126	0.644	0.7293	0.7734	**0.7737**	0.7666	0.5971	0.7613
	F1_−	0.8667	0.8759	0.8584	0.8773	0.8714	0.8738	**0.8854**	0.8337	0.8838
house-votes	F1_+	0.8994	0.9061	0.914	0.9018	0.8859	0.8852	0.9006	0.911	**0.9261**
	F1_−	0.9244	0.9328	0.938	0.927	0.9259	0.9255	0.9309	0.9365	**0.9487**
WDBC	F1_+	0.8424	0.8492	0.85	0.871	**0.9002**	0.8965	0.8979	0.8727	0.8799
	F1_−	0.917	0.9275	0.9254	0.9327	**0.9478**	0.9465	0.9464	0.9376	0.9368
monk2	F1_+	**0.3967**	0.3742	0.3723	0.2398	0.387	0.3946	0.3752	0.0939	0.2654
	F1_−	0.6392	0.6759	0.6701	0.7405	0.6861	0.6949	0.6972	0.7647	**0.77**
breast	F1_+	0.9658	0.9597	0.9332	0.968	0.9665	0.9657	0.9675	0.9188	**0.9692**
	F1_−	0.9312	0.9172	0.8626	0.9358	0.9329	0.9312	0.9356	0.8503	**0.9383**
diabetes	F1_+	0.7952	0.8056	0.755	0.7981	0.7964	0.7972	0.805	0.7732	**0.8097**
	F1_−	0.5269	0.5104	0.3942	0.5441	**0.5811**	0.5783	0.5756	0.3188	0.5538
vehicle	F1_+	0.3335	0.3272	0.4393	0.4347	0.4707	**0.4727**	0.4139	0.0273	0.3281
	F1_−	0.8068	0.8204	0.8311	0.8327	0.8215	0.8213	0.831	0.8614	**0.8664**
german	F1_+	0.7105	0.7458	0.7878	0.7969	0.7949	0.7808	0.7825	0.8139	**0.8236**
	F1_−	0.4107	0.3906	0.3415	0.2967	**0.457**	0.4341	0.4302	0.323	0.2532

RBF kernel

Dataset		SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
wine	F1_+	0.9687	0.9817	0.9633	0.9743	0.9583	0.9813	0.9764	0.9536	**0.9891**
	F1_−	0.985	0.9921	0.9821	0.9882	0.9767	0.989	0.9889	0.9761	**0.9948**
sonar	F1_+	0.7096	0.6975	0.717	0.7189	**0.7448**	0.6811	0.7264	0.6808	0.7205
	F1_−	0.7754	0.7845	0.7908	0.7845	**0.7991**	0.7759	0.7946	0.7816	0.7981
house	F1_+	0.933	0.9332	0.9402	0.9382	0.9067	0.9141	0.929	0.9172	**0.9428**
	F1_−	0.9304	0.9313	0.939	0.9368	0.8938	0.8879	0.9241	0.9169	**0.942**
spectfheart	F1_+	0.3101	0.2252	0.4009	0.4029	0.3811	0.1496	**0.4134**	0.1168	0.2776
	F1_−	0.8639	0.8883	0.8569	0.8727	0.8901	0.8887	**0.8944**	0.8876	0.894
ionosphere	F1_+	0.7855	0.7949	0.7609	0.8164	0.8088	0.8081	**0.8198**	0.7143	0.8078
	F1_−	0.8975	0.8973	0.8904	0.9071	0.8894	**0.9111**	0.9043	0.8719	0.9067
house-votes	F1_+	0.8902	0.9077	0.8933	0.9044	0.8828	0.8488	0.8966	0.8977	**0.9135**
	F1_−	0.9209	0.9354	0.9214	0.9298	0.9232	0.9175	0.9306	0.9289	**0.9374**
WDBC	F1_+	0.8326	0.8324	0.8523	0.8553	0.8795	0.8348	0.8789	**0.8797**	0.8494
	F1_−	0.913	0.9128	0.9237	0.9165	0.9362	0.9208	0.9363	**0.9391**	0.9239
monk2	F1_+	0.4553	0.1892	**0.4915**	0.4803	0.4574	0.2122	0.4619	0.335	0.4403
	F1_−	0.7067	0.7583	0.7332	0.7468	0.7178	0.773	0.7166	0.719	**0.7787**
breast	F1_+	0.9563	0.9674	0.9679	0.968	0.9668	0.9677	0.9698	0.971	**0.9726**
	F1_−	0.931	0.9348	0.9368	0.9365	0.9343	0.9353	0.9408	0.9427	**0.9464**
diabetes	F1_+	0.7917	0.8051	0.7866	0.7798	0.7991	**0.8117**	0.7994	0.807	0.8103
	F1_−	0.5307	0.4722	0.4901	0.4805	**0.5959**	0.4641	0.58	0.5321	0.5025
vehicle	F1_+	0.4655	0.3468	0.3798	0.427	**0.4706**	0.1025	0.4532	0.3728	0.3223
	F1_−	0.8218	0.8319	0.8411	0.8426	0.8182	0.8555	0.8252	0.8342	**0.8695**
german	F1_+	0.7882	0.8233	0.816	0.8114	0.8166	0.8185	0.8113	0.8231	**0.8263**
	F1_−	0.3288	0.016	0.2817	0.3067	**0.3359**	0.1696	0.3007	0.2386	0.2593

Table A.3
F1 scores of all nine algorithms on partial binary classification Benchmark/UCI/KEEL data sets with 40 labelled examples.

Linear kernel

Dataset		SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLlSVM	SSC-EKE
wine	F1_+	0.9664	0.9718	0.9049	0.9832	0.9449	0.9458	0.9769	0.9081	**0.9938**
	F1_−	0.9819	0.9854	0.9485	0.9916	0.973	0.9738	0.9876	0.9525	**0.997**
sonar	F1_+	0.7136	0.7109	0.6628	0.7148	**0.7456**	0.7434	0.7455	0.7111	0.7311
	F1_−	0.7496	0.7592	0.7261	0.7613	**0.7873**	0.7804	0.782	0.7666	0.7741
house	F1_+	0.9499	0.9629	0.9533	0.9721	0.9412	0.9392	0.9555	0.9527	**0.9766**
	F1_−	0.9461	0.9565	0.9467	0.9667	0.9392	0.937	0.9508	0.9453	**0.9719**
spectfheart	F1_+	0.3989	**0.4425**	0.3656	0.2119	0.2095	0.36	0.3907	0.016	0.2031
	F1_−	0.8321	0.869	0.7588	0.8817	0.8887	**0.8903**	0.8879	0.8831	0.8872
ionosphere	F1_+	0.7225	0.7145	0.643	0.7267	0.7646	**0.7775**	0.7723	0.6247	0.7537
	F1_−	0.8866	0.8903	0.8637	0.8923	0.8819	0.8904	0.8935	0.8639	**0.8991**
house-votes	F1_+	0.8973	0.9028	0.918	0.9314	0.9155	0.918	0.9193	0.9227	**0.9423**
	F1_−	0.9207	0.9308	0.9417	0.9504	0.9398	0.9428	0.9431	0.945	**0.958**
WDBC	F1_+	0.9129	0.9116	0.8964	0.922	0.9035	0.9023	0.9229	0.8772	**0.9372**
	F1_−	0.9516	0.9531	0.9446	0.9567	0.9501	0.9495	0.9582	0.9403	**0.9651**
monk2	F1_+	0.1494	0.1868	0.2513	0.2299	0.3974	**0.4291**	0.3029	0.1075	0.3295
	F1_−	0.7738	0.7556	0.6854	0.7783	0.7225	0.7275	0.7561	0.7656	**0.7897**
breast	F1_+	0.9716	0.9676	0.9294	0.9721	0.9577	0.9574	0.9641	0.9078	**0.9741**
	F1_−	0.947	0.9394	0.8536	0.9477	0.9165	0.9158	0.9315	0.8296	**0.9518**
diabetes	F1_+	0.802	0.8152	0.7435	0.8125	0.8152	0.8175	**0.822**	0.7646	0.8201
	F1_−	0.6134	0.5938	0.4381	0.6261	0.63	0.6281	0.6071	0.4058	**0.6403**
vehicle	F1_+	0.2243	0.3498	**0.4369**	0.3259	0.4237	0.3922	0.4271	0.0534	0.4359
	F1_−	0.8511	0.8329	0.8543	0.8571	0.8444	0.846	0.8558	0.8577	**0.8644**
german	F1_+	0.7264	0.7519	0.7378	0.8122	0.7796	0.7827	0.7801	0.8068	**0.824**
	F1_−	0.4474	0.4021	0.4168	0.2626	**0.4821**	0.4623	0.4805	0.3878	0.217

RBF kernel

Dataset		SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLlSVM	SSC-EKE
wine	F1_+	0.9862	0.9902	0.9808	0.9938	0.9701	0.9835	0.9925	0.9644	**0.9954**
	F1_−	0.9931	0.9945	0.9898	0.997	0.9847	0.9913	0.9961	0.9816	**0.9977**
sonar	F1_+	0.736	0.7321	0.7514	0.7515	0.7668	0.7033	0.7645	0.714	**0.7762**
	F1_−	0.7792	0.7841	0.8008	0.8023	0.8026	0.7497	0.7947	0.7743	**0.8147**
house	F1_+	0.9415	0.9501	0.9508	0.9512	0.9282	**0.9636**	0.9508	0.944	0.9633
	F1_−	0.9339	0.9405	0.9442	0.9438	0.9254	0.9539	0.9439	0.915	**0.957**
spectfheart	F1_+	0.2042	0.1159	**0.4882**	0.4192	0.3364	0.0482	0.4648	0.2682	0.3706
	F1_−	0.877	0.8837	0.8603	0.8842	0.8942	0.8854	0.8944	0.8932	**0.8983**
ionosphere	F1_+	0.8507	0.8305	0.789	0.8717	0.8363	0.8509	0.8549	0.7757	**0.873**
	F1_−	0.9279	0.9216	0.9071	0.9385	0.9154	0.9331	0.9244	0.8966	**0.9397**
house-votes	F1_+	0.9066	0.9068	0.9121	0.9217	0.9152	0.8926	0.9196	0.9075	**0.9254**
	F1_−	0.9322	0.9264	0.938	0.944	0.9396	0.9338	0.9443	0.9338	**0.9472**
WDBC	F1_+	0.8972	0.8971	0.8741	0.8994	0.8874	0.8617	0.8997	0.8981	**0.9137**
	F1_−	0.9412	0.9451	0.9274	0.9421	0.9361	0.9318	0.9439	0.8483	**0.9527**
monk2	F1_+	0.4374	0.1453	**0.5183**	0.515	0.4817	0.2975	0.452	0.3748	0.4508
	F1_−	0.7594	0.7829	0.7548	0.7703	0.7462	0.782	0.7547	0.7641	**0.7973**
breast	F1_+	0.9706	0.9701	0.9655	0.9717	0.9609	0.9601	0.966	0.9692	**0.9737**
	F1_−	0.9465	0.9444	0.9359	0.9484	0.9233	0.9195	0.9365	0.9431	**0.9523**
diabetes	F1_+	0.7937	0.8151	0.7714	0.7997	0.7898	0.8034	0.8011	0.8172	**0.8178**
	F1_−	0.6	0.528	0.554	0.5835	**0.6175**	0.475	0.6099	0.5808	0.5782
vehicle	F1_+	0.1809	0.2309	0.3701	0.3	**0.4373**	0.1125	0.4181	0.2245	0.2409
	F1_−	0.85	0.8484	0.8461	0.8555	0.8426	0.8592	0.8492	0.8536	**0.864**
german	F1_+	0.7954	0.8203	0.8066	0.8112	0.8043	0.8189	0.8035	0.8211	**0.8243**
	F1_−	0.359	0.0013	0.3157	0.2923	0.3431	0.2098	**0.3725**	0.256	0.2564

Table A.4
F1 scores of all nine algorithms on partial real-world data sets with the RBF kernel.

Dataset		SVM	TSVM	LapRLS	LapSVM	MeanS3VM-*iter*	MeanS3VM-*mkl*	CS4VM	WeLISVM	SSC-EKE
comp VS rec	F1_+	0.7468	**0.7773**	0.7378	0.7578	0.7498	0.7424	0.7616	0.7674	0.7757
	F1_−	0.7218	0.7593	0.7334	0.732	0.7331	0.7158	0.7383	0.7091	**0.7726**
comp VS sci	F1_+	0.685	0.6737	0.674	0.6905	0.6801	0.6389	0.6869	0.6525	**0.6948**
	F1_−	0.7205	0.7337	0.7093	0.7355	0.7097	0.7179	0.7223	0.7371	**0.7462**
comp VS talk	F1_+	0.7194	0.7385	0.7213	0.7461	0.741	0.7308	0.7441	0.7457	**0.7564**
	F1_−	0.7514	**0.7835**	0.7688	0.7681	0.745	0.7367	0.7536	0.7808	0.7824
rec VS sci	F1_+	0.6648	0.6767	0.6659	0.6866	0.665	0.6394	0.6665	0.6709	**0.692**
	F1_−	0.6831	0.6806	0.6743	0.7032	0.6613	0.6529	0.6679	**0.7169**	0.7075
rec VS talk	F1_+	0.6725	**0.6878**	0.6611	0.6865	**0.6878**	0.6701	0.6873	0.6764	0.6819
	F1_−	0.7332	0.7196	0.7237	0.7354	0.7121	0.7275	0.7297	**0.7641**	0.7608
sci VS talk	F1_+	0.6634	0.6833	0.6442	0.6819	0.6761	0.6822	0.6702	**0.7129**	0.6847
	F1_−	0.6633	0.6593	0.674	0.6821	0.6665	0.6062	0.6724	0.6287	**0.6957**
gisette_4 VS 9	F1_+	0.9083	0.9119	0.9128	0.9175	0.9111	0.9101	0.9134	0.9137	**0.9274**
	F1_−	0.9112	0.9141	0.9155	0.9208	0.9123	0.913	0.9159	0.9159	**0.9304**
mnist_3 VS 8	F1_+	0.9444	0.9311	0.9487	0.9479	0.9343	0.9351	0.9483	0.9391	**0.9568**
	F1_−	0.9463	0.9333	0.9525	0.9518	0.9382	0.9381	0.9505	0.9426	**0.9596**

B. Proofs

B.1. Proof of Theorem 1

Any function $f \in H_K$ can be uniquely decomposed into a component f_s in the subspace spanned by the kernel functions $\{K(\mathbf{x}_i, \cdot), 1 \le i \le l\}$ and a component $f_\perp$ perpendicular to this subspace. That is, $f = f_s + f_\perp$ and $f_s = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \cdot)$.

Since the kernel K has the reproducing property, i.e., $f(\mathbf{x}_j) = \langle f, K(\mathbf{x}_j, \cdot)\rangle = \langle f_s, K(\mathbf{x}_j, \cdot)\rangle + \langle f_\perp, K(\mathbf{x}_j, \cdot)\rangle = \langle f_s, K(\mathbf{x}_j, \cdot)\rangle = \langle \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \cdot), K(\mathbf{x}_j, \cdot)\rangle = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x}_j)$, we know that the term related to the loss function in (1) depends only on the values of the coefficients $\{\alpha_i, 1 \le i \le l\}$ and the gram matrix of the kernel K. Furthermore, because $||f||_K^2 = ||\sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \cdot)||_K^2 + ||f_\perp||_K^2$, we can deduce that the minimizer of (1) must have $f_\perp = 0$. Combining these analyses, it is clear that the minimizer of (1) is $f^*(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}_i, \mathbf{x})$. □

B.2. Proof of Theorem 2

Using the Lagrange multipliers $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_l)$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_l)$, we can generate the Lagrange function:

$$L(\boldsymbol{\alpha}, b, \xi, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{1}{l}\sum_{i=1}^{l}\xi_i + \frac{1}{2}\alpha^{\mathrm{T}} 2\gamma \boldsymbol{K}\boldsymbol{\alpha} - \sum_{\boldsymbol{i}=1}^{\boldsymbol{l}} \beta_i \left(y_i \left(\sum_{j=1}^{l} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) - 1 + \xi_i \right) - \sum_{\boldsymbol{i}=1}^{\boldsymbol{l}} \gamma_i \xi_i. \tag{B.1}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions, we obtain

$$\frac{\partial L}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{l} \beta_i y_i = 0 \tag{B.2}$$

$$\begin{aligned} &\tfrac{\partial L}{\partial \xi_i} = 0 \Leftrightarrow \tfrac{1}{l} - \beta_i - \gamma_i = 0 \\ &\Leftrightarrow 0 \le \beta_i \le \tfrac{1}{l} \quad (\xi_i, \gamma_i \text{ arenon} - \text{negative}) \end{aligned} \tag{B.3}$$

Substituting (B.2)-(B.3) into (B.1), we can formulate a reduced Lagrange function:

$$\begin{aligned} L^R(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \tfrac{1}{2}\alpha^{\mathrm{T}} 2\gamma \boldsymbol{K}\boldsymbol{\alpha} - \sum_{\boldsymbol{i}=1}^{\boldsymbol{l}} \beta_i \left(y_i \sum_{j=1}^{l} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - 1 \right) \\ &= \tfrac{1}{2}\alpha^{\mathrm{T}} 2\gamma \boldsymbol{K}\boldsymbol{\alpha} - \alpha^{\mathrm{T}} \boldsymbol{K}\mathbf{Y}\beta + \sum_{\boldsymbol{i}=1}^{\boldsymbol{l}} \beta_i, \end{aligned} \tag{B.4}$$

where $\mathbf{Y} = \mathrm{diag}(y_1, y_2, \ldots, y_l)$ and $\boldsymbol{K}$ is the $l \times l$ kernel matrix.

Setting the derivative of (B.4) to zero with respect to $\boldsymbol{\alpha}$, we get

$$\frac{\partial L^R}{\partial \alpha} = 2\gamma \boldsymbol{K}\boldsymbol{\alpha} - \boldsymbol{K}\mathbf{Y}\beta = 0 \Leftrightarrow \alpha = \frac{\mathbf{Y}\beta}{2\gamma} \tag{B.5}$$

Substituting (B.5) into (B.4) and combining (B.2) and (B.3), we can eventually obtain (4). □

B.3. Proof of Theorem 3

Likewise, the function $f \in H_K$ can be uniquely *decomposed* into a component f_s in the subspace spanned by the kernel functions $\{K(\mathbf{x}_i, \cdot),\ 1 \le i \le l+u\}$ and a component $f_\perp$ orthogonal to this subspace. That is, $f = f_s + f_\perp$ and $f_s = \sum_{i=1}^{l+u} \alpha_i K(\mathbf{x}_i, \cdot)$.

Based on the reproducing *property* of the kernel K, $f(\mathbf{x}_j) = \langle f, K(\mathbf{x}_j,\cdot)\rangle = \langle f_s, K(\mathbf{x}_j,\cdot)\rangle + \langle f_\perp, K(\mathbf{x}_j,\cdot)\rangle = \langle f_s, K(\mathbf{x}_j,\cdot)\rangle = \langle \sum_{i=1}^{l+u}\alpha_i K(\mathbf{x}_i,\cdot), K(\mathbf{x}_j,\cdot)\rangle = \sum_{i=1}^{l+u}\alpha_i K(\mathbf{x}_i,\mathbf{x}_j)$. We know that the terms related to the loss function and the intrinsic norm $||f||_I^2$ in (7) rely only on the values of the coefficients $\{\alpha_i,\ 1\le i\le l+u\}$ and the gram matrix of the kernel K. Furthermore, because $||f||_K^2 = ||\sum_{i=1}^{l+u}\alpha_i K(\mathbf{x}_i,\cdot)||_K^2 + ||f_\perp||_K^2$, we can deduce that the minimizer of (7) must have $f_\perp = 0$. Combining these analyses, we know that the minimizer of (7) must be $f^*(\mathbf{x}) = \sum_{i=1}^{l+u}\alpha_i K(\mathbf{x}_i,\mathbf{x})$. □

B.4. Proof of Theorem 4

Using the Lagrange multipliers $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_l)$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_l)$, we can generate the Lagrange function:

$$\begin{aligned} L(\boldsymbol{\alpha}, b, \xi, \boldsymbol{\beta}, \boldsymbol{\gamma}) &= \tfrac{1}{l}\sum_{i=1}^{l}\xi_i + \tfrac{1}{2}\alpha^{\mathrm{T}}\left(2\gamma_A\boldsymbol{K} + 2\tfrac{\gamma_I}{(u+l)^2}\boldsymbol{K}\boldsymbol{L}\boldsymbol{K}\right)\boldsymbol{\alpha} \\ &- \sum_{i=1}^{l}\beta_i\left(y_i\left(\sum_{j=1}^{l+u}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) + b\right) - 1 + \xi_i\right) - \sum_{i=1}^{l}\gamma_i\xi_i. \end{aligned} \tag{B.6}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions, we get

$$\frac{\partial L}{\partial b} = 0 \Leftrightarrow \sum_{i=1}^{l}\beta_i y_i = 0 \tag{B.7}$$

$$\begin{aligned} &\tfrac{\partial L}{\partial \xi_i} = 0 \Leftrightarrow \tfrac{1}{l} - \beta_i - \gamma_i = 0 \\ &\Leftrightarrow 0 \le \beta_i \le \tfrac{1}{l}\quad (\xi_i, \gamma_i \text{ arenon} - \text{negative}) \end{aligned} \tag{B.8}$$

Substituting (B.7) and (B.8) into (B.6), we can formulate a reduced Lagrange function:

$$\begin{aligned} L^R(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= \tfrac{1}{2}\alpha^{\mathrm{T}}\left(2\gamma_A\boldsymbol{K} + 2\tfrac{\gamma_I}{(u+l)^2}\boldsymbol{K}\boldsymbol{L}\boldsymbol{K}\right)\boldsymbol{\alpha} - \sum_{i=1}^{l}\beta_i\left(y_i\sum_{j=1}^{l+u}\alpha_j K(\mathbf{x}_i,\mathbf{x}_j) - 1\right) \\ &= \tfrac{1}{2}\alpha^{\mathrm{T}}\left(2\gamma_A\boldsymbol{K} + 2\tfrac{\gamma_I}{(u+l)^2}\boldsymbol{K}\boldsymbol{L}\boldsymbol{K}\right)\boldsymbol{\alpha} - \alpha^{\mathrm{T}}\boldsymbol{K}\mathbf{J}^{\mathrm{T}}\mathbf{Y}\beta + \sum_{i=1}^{l}\beta_i, \end{aligned} \tag{B.9}$$

where $\mathbf{J} = [\mathbf{I}\ \mathbf{0}]$ is a $l\times(l+u)$ matrix, with $\boldsymbol{I}$ being the $l\times l$ identity matrix, $\mathbf{Y} = \mathrm{diag}(y_1, y_2, \ldots, y_l)$, and $\boldsymbol{K}$ is the $(l+u)\times(l+u)$ kernel matrix.

Setting the derivative of (B.9) to zero with respect to $\boldsymbol{\alpha}$, we obtain

$$\begin{aligned} &\tfrac{\partial L^R}{\partial \alpha} = \left(2\gamma_A\boldsymbol{K} + 2\tfrac{\gamma_I}{(u+l)^2}\boldsymbol{K}\boldsymbol{L}\boldsymbol{K}\right)\boldsymbol{\alpha} - \boldsymbol{K}\mathbf{J}^{\mathrm{T}}\mathbf{Y}\beta = 0 \\ &\Leftrightarrow \alpha = \left(2\gamma_A\mathbf{I} + \left(2\gamma_I/(u+l)^2\right)\mathbf{L}\boldsymbol{K}\right)^{-1}\mathbf{J}^{\mathrm{T}}\mathbf{Y}\beta \end{aligned} \tag{B.10}$$

Substituting (B.10) into (B.9) and combining (B.7) and (B.8), we can eventually obtain (10). □

References

[1] M. Aly. (2005). Survey on Multiclass Classification Methods [Online], Available: https://www.cs.utah.edu/~piyush/teaching/aly05multiclass.pdf.
[2] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. vol.4 (2010) 40–79.
[3] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labelled and unlabelled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[4] K. Bennett, A. Demiriz, Semi-Supervised Support Vector Machines, Proc. Adv. Neural Inf. Process. Syst. 11 (1999) 368–374.
[5] G.C. Cawley, Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs, in: International Joint Conference on Neural Networks, Vancouver, BC, 2006, pp. 1661–1668.
[6] V. Christianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, 2002.
[7] R.G. Congalton, R.A. Mead, A review of three discrete multivariate analysis techniques used in assessing the accuracy of remotely sensed data from error matrices, IEEE Trans. Geosci. Remote Sens. GE-24 (January(1)) (1986) 169–174.
[8] W.Y. Dai, G.R. Xue, Q. Yang, Y. Yu, Co-clustering based classification for out-of-domain documents, in: Proceedings of the Thirteenth ACM SIGKDD, 2007, pp. 210–219.
[9] K.-B. Duan1, S.S. Keerthi, Which is the best multiclass SVM method? an empirical study, in: *Proceedings of the 6th International Workshop on MCS*, June, Seaside, CA, USA, 2005, pp. 278–285.
[10] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (March(2)) (2002) 415–425.
[11] G.-B. Huang, P. Saratchandran, N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Trans. Neural Netw. 16 (January(1)) (2005) 57–67.
[12] G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Trans. Neural Netw. 17 (July(4)) (2006) 879–892.
[13] T. Joachims, Transductive inference for text classification using support vector machines, in: Proc. 16th Int. Conf. Mach. Learn., Bled, Slovenia, 1999, pp. 200–209.
[14] K.I. Kim, K. Jung, S. Park, H.J. Kim, Support vector machines for texture classification, IEEE Trans. Pattern Anal. Mach. Intell. 24 (November(11)) (2002) 1542–1550.
[15] Y.-F. Li, J. Kwok, Z.-H. Zhou, Semi-supervised learning using label mean, in: Proc. 26th Int. Conf. Mach. Learn., 2009, pp. 633–640.
[16] Y.-F. Li, J. Kwok, Z.-H. Zhou, Cost-sensitive semi-supervised support vector machine, in: Proceedings of the 24th AAAI Conference on Artificial Intelligences (AAAI'10), 2010, pp. 500–505.
[17] Y.-F. Li, I. Tsang, J. Kwok, Z.-H. Zhou, Convex and scalable weakly labelled SVMs, J. Mach. Learn. Res. 14 (1) (2013) 2151–2188.
[18] Y.-F. Li, Z.-H. Zhou, Towards making unlabelled data never hurt, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 175–188.
[19] G. Li, K. Chang, S.C.H. Hoi, Multi-view semi-supervised learning with consensus, IEEE Trans. Knowl. Data Eng. 24 (11) (2012) 2040–2051.

[20] C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 464–471.
[21] S.-T. Liu, X.-L. Luo, A method based on Rayleigh quotient gradient flow for extreme and interior eigenvalue problems, Linear Algebra Appl. 432 (7) (2010) 1851–1863.
[22] K. Lu, Q. Wang, J. Xue, W. Pan, 3D model retrieval and classification by semi-supervised learning with content-based similarity, Inf. Sci. 281 (2014) 703–713.
[23] A.K. McCallum, Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, 1996. http://www.cs.cmu.edu/mccallum/bow.
[24] P. Melin, J. Amezcua, F. Valdez, O. Castillo, A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias, Inf. Sci. 279 (2014) 483–497.
[25] N. Nedjah, F.P. da Silva, A.O. de Sá, L.M. Mourelle, D.A. Bonilla, A massively parallel pipelined reconfigurable design for M-PLN based neural networks for efficient image classification, Neurocomputing 183 (2016) 39–55.
[26] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, IJCV 42 (3) (2001) 145–175.
[27] P. Qian, F.-L. Chung, S. Wang, Z. Deng, Fast graph-based relaxed clustering for large data sets using minimal enclosing ball, IEEE Trans. SMC-B 42 (3) (2012) 672–687.
[28] H. Qu, Y. Oussar, G. Dreyfus, W. Xu, Regularized recurrent least squares support vector machines, in: International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Shanghai, 2009, pp. 508–511.
[29] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (January(1)) (1998) 23–38.
[30] M. Roy, S. Ghosh, A. Ghosh, A novel approach for change detection of remotely sensed images using semi-supervised multiple classifier system, Inf. Sci. 269 (2014) 35–47.
[31] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (August(8)) (2000) 888–905.
[32] L.M. Surhone, M.T. Timpledon, S.F. Marseken (Eds.), Rayleigh Quotient: Mathematics, Hermitian Matrix, Conjugate Transpose, Eigenvalue, Eigenvector and Eigenspace, Min-Max Theorem, Rayleigh Quotient Iteration, Numerical Range, Euclidean Vector, Betascript Press, Saarland, Germany, 2010.
[33] D.M.J. Tax, R.P.W. Duin, Using two-class classifiers for multiclass classification, in: 16th International Conference on Pattern Recognition, vol. 2, 2002, pp. 124–127.
[34] I. Tsang, J. Kwok, J. Zurada, Generalized core vector machines, IEEE Trans. Neural Netw. 17 (5) (2006) 1126–1139.
[35] I. Tsang, J. Kwok, P. Cheung, Core vector machines: fast SVM training on very large data sets, J. Mach. Learn. Res. 6 (2005) 363–392.
[36] V.N. Vapnik, The Natural of Statistical Learning Theory, Springer, New York, 1995.
[37] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[38] X. Wang, F.-L. Chung, S. Wang, On minimum class locality preserving variance support vector machine, Pattern Recognit. 43 (8) (2010) 2753–2762.
[39] Y. Wang, S. Chen, Z.H. Zhou, New semi-supervised classification method based on modified cluster assumption, IEEE Trans. Neural Netw. Learn. Syst. 23 (May(5)) (2012) 689–702.
[40] Y. Wang, S. Chen, Safety-aware semi-supervised classification, IEEE Trans. Neural Netw. Learn. Syst. 24 (November(11)) (2013) 1763–1772.
[41] B. Wang, J. Tsotsos, Dynamic label propagation for semi-supervised multi-class multi-label classification, Pattern Recognit. 52 (2016) 75–84.
[42] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.
[43] S. Xiang, F. Nie, C. Zhang, Semi-supervised classification via local spline regression, IEEE Trans. Pattern Anal. Mach. Intell. 32 (November(11)) (2010) 2039–2053.
[44] G. Yu, G. Zhang, C. Domeniconi, Z. Yu, J. You, Semi-supervised classification based on random subspace dimensionality reduction, Pattern Recognit. 45 (3) (2012) 1119–1135.
[45] Y. Zhang, L. Wu, Classification of fruits using computer vision and a multiclass support vector machine, Sensors 12 (2012) 12489–12505.
[46] D. Zhang, J. Wang, X. Zhao, X. Wang, A bayesian hierarchical model for comparing average F1 scores, in: IEEE International Conference on Data Mining, 2015, pp. 589–598.