Advances in Data Analysis and Classification (2021) 15:159–191
https://doi.org/10.1007/s11634-020-00390-y

REGULAR ARTICLE

A novel semi-supervised support vector machine with asymmetric squared loss

Huimin Pei[1] · Qiang Lin[1] · Liran Yang[2] · Ping Zhong[1]

Received: 1 September 2019 / Revised: 16 January 2020 / Accepted: 2 March 2020 /
Published online: 10 March 2020

Abstract
Laplacian support vector machine (LapSVM), which is based on the semi-supervised manifold regularization learning framework, performs better than the standard SVM, especially for the case where the supervised information is insufficient. However, the use of hinge loss leads to the sensitivity of LapSVM to noise around the decision boundary. To enhance the performance of LapSVM, we present a novel semi-supervised SVM with the asymmetric squared loss (asy-LapSVM) which deals with the expectile distance and is less sensitive to noise-corrupted data. We further present a simple and efficient functional iterative method to solve the proposed asy-LapSVM, in addition, we prove the convergence of the functional iterative method from two aspects of theory and experiment. Numerical experiments performed on a number of commonly used datasets with noise of different variances demonstrate the validity of the proposed asy-LapSVM and the feasibility of the presented functional iterative method.

Keywords Semi-supervised learning · Support vector machine · Noise sensitivity · Asymmetric squared loss

Mathematics Subject Classification 90C20 · 90C90

1 Introduction

As a powerful supervised learning algorithm based on solid theoretical foundations, support vector machine (SVM) (Cristianini and Shawe-Taylor 2000; Vapnik 1995) has gained substantial attention in many research areas. By simultaneously minimizing the regularization term and the hinge loss function, SVM implements the structural risk

✉ Ping Zhong
zping@cau.edu.cn

[1] College of Science, China Agricultural University, Beijing 100083, China

[2] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

minimization principle rather than the traditional empirical risk minimization principle. Due to its remarkable characteristics such as the good generalization performance, the absence of local minima and sparse representation of the solution, SVM and its variants (Zhao et al. 2019; Melki et al. 2018) have been successfully applied to a wide range of applications, including classification, regression, clustering, representational learning and so on. However, there still exists a great space for improvements on the traditional SVM.

On the one hand, the use of hinge loss function leads to the sensitivity of SVM to re-sampling and noise around the decision boundary. To alleviate this disadvantage, various methods have been proposed so far (Bi and Zhang 2004; Shivaswamy et al. 2006; Xu et al. 2009; Zhong 2012; Wang et al. 2015; Wang and Zhong 2014). Recently, motivated by the link between the hinge loss and the shortest distance, Huang et al. (2014) proposed a new SVM classifier with the pinball loss (Pin-SVM). The pinball loss, which shares many good properties and brings noise insensitivity for classification, is related to the quantile distance (Koenker 2005; Jumutc et al. 2013) between two classes. The theoretical analysis and experimental results show that Pin-SVM is more stable to noise-corrupted data compared with the traditional SVM. Lately, to speed up the training process for Pin-SVM, Huang et al. (2014) further exploited the expectile distance as a surrogate of the quantile distance and propose a new SVM classifier with the asymmetric squared loss (aLS-SVM). This is motivated by the fact that the expectile value, which is related to minimizing the asymmetric squared loss (Lu et al. 2018), has similar statistical properties to the quantile value. In other words, aLS-SVM is an approximation of Pin-SVM which can be effectively solved.

On the other hand, a main challenge for the standard SVM is its dependence on sufficient supervised information. In many real-world applications, such as natural language parsing (Tur et al. 2005), spam filtering (Guzella and Caminhas 2009), and video surveillance (Zhang et al. 2011), the acquisition of enough labeled data is usually difficult while unlabeled data are available in large quantity. In such situations, the performance of SVM usually deteriorates since a lot of information carried by the unlabeled data is simply ignored. To handle the problem, semi-supervised learning (SSL) is proposed. It has become an efficient paradigm (Chapelle et al. 2006; Scardapane et al. 2016; Li et al. 2017; Calma et al. 2018), especially that with manifold regularization (MR) (Belkin et al. 2006; Chen et al. 2014) which tries to capture the geometric information from both labeled and unlabeled data and makes the smoothness of classifiers along the intrinsic manifold via an additional regularization term. With the addition of the MR into the conventional SVM, Belkin et al. (2006) presented the classical Laplacian SVM (LapSVM), in which the geometric information embedded in the abundant unlabeled data is fully considered to build more reasonable classifiers. After that, under the semi-supervised MR learning framework, a number of SVM-based SSL algorithms have emerged (Sun 2013; Khemchandani and Pal 2016; Pei et al. 2017a, b). However, similar to SVM, LapSVM is sensitive to noise-corrupted data, since it also employs the minimal distance which is related to the hinge loss to measure the margin between two classes.

In this paper, inspired by the studies above, we propose a novel semi-supervised support vector machine with the asymmetric squared loss (asy-LapSVM). Our motivation mainly depends on the facts that the MR has the ability to encode the geometric information embedded in the unlabeled data and the expectile distance is stable to noise-corrupted data. In other words, we hope that the proposed asy-LapSVM possesses the ability to make full use of the abundant unlabeled data, and moreover, it is stable to noise-corrupted data. Moreover, to speed up the training process, we present a simple and efficient functional iterative method to replace the traditional quadratic programming (QP) to solve the involved optimization problems. The convergence property of the iterative method is proved by both of the theory and experiments. Experimental results on a number of commonly used datasets show that the proposed asy-LapSVM achieves a significant performance in comparison with several popular supervised learning (SL) and SSL algorithms.

In summary, by incorporating the properties of the SSL and the asymmetric squared loss, the advantages of the proposed algorithm are as follows:

- We construct a robust LapSVM framework by adopting an asymmetric squared loss, and it can be effectively solved with the help of a simple functional iterative method.
- The proposed model belongs to inductive learning and is natural for out-of-sample data, which can avoid expensive graph computation.
- The extensive comparison experiments with several related methods on widely used benchmark datasets demonstrate the effectiveness of the proposed method.

Moreover, compared with the very recently published algorithms, such as the pure SSL algorithms in references (Huang et al. 2014; Ma et al. 2019), the proposed asy-LapSVM not only maintains their primary advantages such as the ability to make use of unlabeled data and handle unseen data in the testing phase directly, but also makes them less sensitive to noise-corrupted data (feature noises). As for the robust SSL algorithms algorithms in references (Du et al. 2019; Pei et al. 2018; Gu et al. 2019), they are all proposed for outliers among labeled data (label noises).

The remainder of this paper is organized as follows. Section 2 briefly outlines the background. The novel asy-LapSVM is proposed in Sect. 3, which includes both the linear and nonlinear cases. Moreover, the proof of convergence of the iterative algorithm is given. After presenting the experimental results on multiple datasets in Sect. 4, we conclude this paper in Sect. 5.

2 Background

In this section, some background knowledge of the proposed algorithm including the asymmetric squared loss function and the semi-supervised manifold regularization learning framework is briefly reviewed.

2.1 Asymmetric squared loss function

In the field of machine learning, loss function is usually one of the key issues in designing learning algorithms since most problems require it to describe the cost of the discrepancy between the prediction and the observation. In fact, the use of the loss function can be traced back to a long time ago. For example, the least-square loss function for regression was already employed by Legendre, Gauss, and Adrain in the early 19th century (Steinwart and Christmann 2008). At present, various margin-based loss functions, such as hinge loss, squared loss, exponential loss, logistic loss, and brown boost loss have been exploited to search for the optimal classification and regression functions.

In a binary classification problem, a large margin between two classes plays an important role to obtain a good classifier. Traditionally, the margin is measured by the minimal distance between two sets, which is related to the hinge loss (1) or the squared hinge loss (2):

$$v_{\mathrm{hin}}(r) = \max\{0, r\}, \tag{1}$$

$$v_{\mathrm{shin}}(r) = \max\{0, r\}^2, \tag{2}$$

where $r = 1 - yf(\boldsymbol{x})$, in which $\boldsymbol{x} \in R^d$ is the input sample, $y \in \{+1, -1\}$ is the corresponding output, and $f(\boldsymbol{x})$ is the prediction. However, measuring the margin by the minimal value leads to the sensitivity of the corresponding classifiers to re-sampling and noise around the decision boundary. To overcome this weak point, Huang et al. (2014) employed the quantile value which has been deeply studied and widely applied in regression problems, as a surrogate of the minimal value to measure the margin between classes, and proposed the following pinball loss (3) which brings noise insensitivity:

$$v_{\mathrm{pin}}(r) = \begin{cases} ur, & r \geq 0, \\ -(1-u)r, & r < 0, \end{cases} \tag{3}$$

where $0 < u < 1$. Different from the hinge loss, the pinball loss gives an additional penalty on the correctly classified samples. So, the pinball loss can be regarded as an extension to the hinge loss. However, the pinball loss is non-smooth and its minimization is more difficult than that of smooth loss functions. Hence, Huang et al. (2014) modified the measurement of margin by taking the expectile value, which is related to the following smooth asymmetric squared loss:

$$v_{\mathrm{asy}}(r) = \begin{cases} ur^2, & r \geq 0, \\ (1-u)r^2, & r < 0. \end{cases} \tag{4}$$

The property of asymmetric squared loss is similar to that of the pinball loss. From the definition of $v_{\mathrm{asy}}(\cdot)$, one observes that when the value of u is 1, it becomes the squared hinge loss $v_{\mathrm{shin}}(\cdot)$. Similar to the difference between the pinball loss and the hinge loss, the asymmetric squared loss, which gives an asymmetric penalty for negative and positive losses, can be seen as a generalized squared hinge loss. The plots of $v_{\mathrm{hin}}(r)$, $v_{\mathrm{shin}}(r)$, $v_{\mathrm{pin}}(r)$ and $v_{\mathrm{asy}}(r)$ with $u = 0.83$ are shown in Fig. 1.

2.2 Semi-supervised manifold regularization learning framework

The idea of regularization, which is widely used in machine learning, has its root in mathematics to solve ill-posed problems (Tikhonov 1963). A number of popular learn-

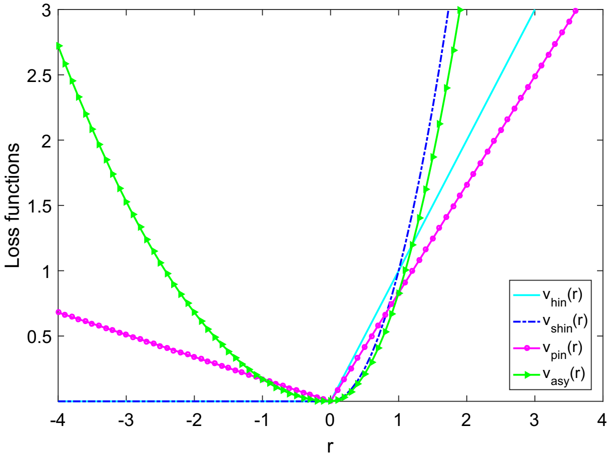

Fig. 1 Plots of loss functions

ing algorithms can be interpreted as a supervised regularization learning framework that consists of different loss functions and complexity measures in an appropriately chosen Reproducing Kernel Hilbert Space (RKHS).

Given a set of labeled samples $\mathcal{S}_l = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ generated according to a probability distribution, the standard supervised regularization learning framework can be written as:

$$\min_{f \in \mathcal{H}_k} \sum_{i=1}^{l} v(r_i) + \lambda_1 \|f\|_k^2, \tag{5}$$

where $r_i = 1 - y_i f(\boldsymbol{x}_i)$ for $i = 1, \ldots, l$, v stands for some loss function on the labeled samples, λ_1 is the weight of $\|f\|_k^2$ that controls the complexity of the unknown function f, in which $\|\cdot\|_k$ is a norm related to a Mercer kernel k in the RKHS $\mathcal{H}_k$. For the choice of k, it can be the linear kernel function, the polynomial kernel function (POLY) and the radial basis function kernel function (RBF) and so on. In the numerical experiments, the RBF kernel (6) which gains an advantages over others (Lu et al. 2018) will be employed:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\sigma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), \tag{6}$$

where σ is the kernel parameter.

It is clear that the excellent performance of standard supervised regularization learning framework (5) is in premise of sufficient labeled samples. However, in many real-world applications, the acquisition of labeled data is generally more difficult than the collection of unlabeled ones. In such situations, a learning framework that is able to make full use of unlabeled data to improve recognition performance is of potentially great significance. In the light of this idea, by adding a manifold regularization term $\|f\|_{\mathcal{I}}^2$ in formulation (5), Belkin et al. (2006) proposed the following semi-supervised manifold regularization learning framework:

$$\min_{f \in \mathcal{H}_k} \sum_{i=1}^{l} v(r_i) + \lambda_1 \|f\|_k^2 + \lambda_2 \|f\|_{\mathcal{I}}^2, \tag{7}$$

where λ_2 is the weight of $\|f\|_{\mathcal{I}}^2$ in the low dimensional manifold (or intrinsic norm), which enforces f smoothness along the intrinsic manifold.

Given l labeled and m unlabeled data in the training set

$$\mathcal{S} = \mathcal{S}_l \cup \mathcal{S}_u = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l} \cup \{\boldsymbol{x}_i\}_{i=l+1}^{n}, \tag{8}$$

where $\boldsymbol{x}_i \in R^d, i = 1, \ldots, n$ ($n = l + m$ is the number of training samples), and $y_i \in \{-1, 1\}$ for $i = 1, \ldots, l$. $\mathcal{S}_u$ denotes a set of m unlabeled data which are drawn according to a marginal distribution. The manifold regularization term $\|f\|_{\mathcal{I}}^2$ can be reexpressed as:

$$\|f\|_{\mathcal{I}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} (f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j))^2 = \mathbf{f}^T L \mathbf{f}, \tag{9}$$

where $\mathbf{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]^T$ is the vector of the n values of f on the training data, L is the graph Laplacian associated to $\mathcal{S}$, given by $L = G - W$, where W is the adjacency matrix which can be defined by the p nearest neighbors, and its non-negative edge weight W_{ij} represents the similarity of every pair of input instances. G is a diagonal matrix with its i-th diagonal element $G_{ii} = \sum_{j=1}^{n} W_{ij}$ represents the weight degree of vertex i. When the manifold regularization term (9) is used in the semi-supervised regularization learning framework (7), we can understand it by the means: if the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ has higher similarity (W_{ij} is larger), the difference of $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ will obtain a big punishment.

By applying the hinge loss function to the semi-supervised manifold regularization learning framework (7), Belkin et al. (2006) proposed the classical LapSVM which can exploit the geometric information of the marginal distribution embedded in unlabeled data to construct a more reasonable classifier. Due to its excellent performance, LapSVM becomes a powerful choice for SSL. More details about LapSVM can be found in Belkin et al. (2006).

3 Semi-supervised SVM with asymmetric squared loss

In this section, we elaborate the formulation of the proposed semi-supervised SVM with the asymmetric squared loss (asy-LapSVM) for semi-supervised binary classification problem. After giving the detailed derivations of the proposed asy-LapSVM in the linear and nonlinear cases, we present a simple and efficient functional iterative method to optimize them. Then we prove the convergence of the proposed functional iterative method in theory. And in the end, we summary the proposed asy-LapSVM.

3.1 Linear asy-LapSVM

Although LapSVM has shown good generalization, we notice that hinge loss function is noise sensibility. To further improve its generalization performance, we provide a stable asy-LapSVM, especially for noise-corrupted data. In specific, unlike LapSVM, we exploit the asymmetric squared loss (4) as a surrogate of the hinge loss for the proposed linear asy-LapSVM. Meanwhile, we maximize the margin measured by the expectile distance between two classes by optimizing with respect to both the weight vector $\boldsymbol{w}$ and the bias term b of the linear asy-LapSVM classifier $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b$. Thus, the regularization term $\|f\|_k^2$ can be expressed as:

$$\|f\|_k^2 = \frac{1}{2}(\|\boldsymbol{w}\|^2 + b^2). \tag{10}$$

As for the manifold regularization term $\|f\|_{\mathcal{I}}^2$, it has the following form:

$$\|f\|_{\mathcal{I}}^2 = \mathbf{f}^T L\mathbf{f} = \boldsymbol{w}^T D^T L D\boldsymbol{w}, \tag{11}$$

where $D = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]^T$ and $\mathbf{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]^T = [\boldsymbol{w}^T\boldsymbol{x}_1, \ldots, \boldsymbol{w}^T\boldsymbol{x}_n]^T = D\boldsymbol{w}$, in which we deliberately reduce the bias term b for convenience.

By substituting the asymmetric squared loss (4), the regularization term (10) and the manifold regularization term (11) into the semi-supervised manifold regularization learning framework (7), the primal problem of the linear asy-LapSVM can be formulated as

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi}\ & \tfrac{1}{2}(\|\boldsymbol{w}\|^2 + b^2) + \tfrac{c}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + \tfrac{\lambda}{2}\boldsymbol{w}^T D^T L D\boldsymbol{w} \\ \text{s.t.}\ \ & Y(D\boldsymbol{w} + \boldsymbol{e}b) \geq \boldsymbol{e}_l - \tfrac{1}{u}\boldsymbol{\xi}, \\ & Y(D\boldsymbol{w} + \boldsymbol{e}b) \leq \boldsymbol{e}_l + \tfrac{1}{1-u}\boldsymbol{\xi}, \end{aligned} \tag{12}$$

where $0 < u < 1$, c and λ are the regularization parameters, $\boldsymbol{\xi}$ is the error variable vector, $\boldsymbol{e}_l$ is the vector of ones of l dimension, $\boldsymbol{e}$ is the vector of ones of n dimension, $Y \in R^{l\times n}$ is a matrix with elements $Y_{ii} = y_i$ and other elements are zeros.

Introducing the nonnegative Lagrange parameter vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, the Lagrangian function for the problem (12) can be expressed as

$$\begin{aligned} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) = & \frac{1}{2}(\|\boldsymbol{w}\|^2 + b^2) + \frac{c}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + \frac{\lambda}{2}\boldsymbol{w}^T D^T L D\boldsymbol{w} \\ & - \boldsymbol{\alpha}_1^T\left[Y(D\boldsymbol{w} + \boldsymbol{e}b) - \boldsymbol{e}_l + \frac{1}{u}\boldsymbol{\xi}\right] \\ & - \boldsymbol{\alpha}_2^T\left[-Y(D\boldsymbol{w} + \boldsymbol{e}b) + \boldsymbol{e}_l + \frac{1}{1-u}\boldsymbol{\xi}\right]. \end{aligned} \tag{13}$$

According to the following Karush–Kuhn–Tucker (KKT) conditions

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = (I + \lambda D^T L D)\boldsymbol{w} - D^T Y^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) = 0, \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial b} = b - \boldsymbol{e}^T Y^T (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) = 0, \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = c\boldsymbol{\xi} - \left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right) = 0, \tag{16}$$

we get

$$\boldsymbol{w} = (I + \lambda D^T L D)^{-1} D^T Y^T (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2), \tag{17}$$

$$b = \boldsymbol{e}^T Y^T (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2), \tag{18}$$

$$\boldsymbol{\xi} = \frac{1}{c}\left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right). \tag{19}$$

Then the following dual problem of (12) can be derived by substituting the above Eqs. (17)–(19) into the Lagrangian function (13)

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2} & \frac{1}{2}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)^T F (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \frac{1}{2c}\left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right)^T \left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right) - e_l^T (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) \\ \text{s.t.} & \boldsymbol{\alpha}_1 \geq 0\boldsymbol{e}_l, \\ & \boldsymbol{\alpha}_2 \geq 0\boldsymbol{e}_l, \end{array} \tag{20}$$

where $F = Y(D(I + \lambda D^T L D)^{-1} D^T + \boldsymbol{e}\boldsymbol{e}^T)Y^T$.

Although the proposed asy-LapSVM can be solved by the classical quadratic programming, inspired by the idea in Balasundaram and Benipal (2016), we adopt a simple functional iterative method, which leads to the minimization of a differentiable convex function in a space of dimensionality equal to the number of classified points and in some cases is dramatically faster than a standard quadratic programming SVM solver, to solve the dual problem (20). Specifically, based on the Karush–Kuhn–Tucker (KKT) necessary and sufficient optimality conditions for the dual problem (20), we have

$$\begin{array}{l} 0\boldsymbol{e}_l \leq \boldsymbol{\alpha}_1 \perp \left(F + \dfrac{I}{cu^2}\right)\boldsymbol{\alpha}_1 - \left(F - \dfrac{I}{cu(1-u)}\right)\boldsymbol{\alpha}_2 - e_l \geq 0\boldsymbol{e}_l, \\ 0\boldsymbol{e}_l \leq \boldsymbol{\alpha}_2 \perp \left(F + \dfrac{I}{c(1-u)^2}\right)\boldsymbol{\alpha}_2 - \left(F - \dfrac{I}{cu(1-u)}\right)\boldsymbol{\alpha}_1 + e_l \geq 0\boldsymbol{e}_l, \end{array} \tag{21}$$

$\Leftrightarrow$

$$\begin{array}{l} 0\boldsymbol{e}_l \leq \dfrac{\boldsymbol{\alpha}_1}{cu^2(1-u)} \perp \dfrac{\boldsymbol{\alpha}_1}{cu^2(1-u)} + \left(F - \dfrac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \\ 0\boldsymbol{e}_l \leq \dfrac{\boldsymbol{\alpha}_2}{cu(1-u)^2} \perp \dfrac{\boldsymbol{\alpha}_2}{cu(1-u)^2} - \left(F - \dfrac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \end{array} \tag{22}$$

$\Leftrightarrow$

$$\begin{aligned}
0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} \perp \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} \\
&\quad + \frac{1}{(1-u)}\left(\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right) \geq 0\boldsymbol{e}_l, \\
0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} \perp \frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} - \frac{1}{u}\left(\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right) \\
&\geq 0\boldsymbol{e}_l,
\end{aligned} \tag{23}$$

where the symbol "⊥" represents two vectors are orthogonal. By exploiting the easily established identity between any two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ (Fung and Mangasarian 2004):

$$0 \leq \boldsymbol{a} \perp (\boldsymbol{a} - \boldsymbol{b}) \geq 0 \Leftrightarrow \boldsymbol{a} = \boldsymbol{b}_+ = \frac{\boldsymbol{b} + |\boldsymbol{b}|}{2}, \tag{24}$$

where $\boldsymbol{b}_+ = \max\{\boldsymbol{0}, \boldsymbol{b}\}$ and $|\boldsymbol{b}|$ denotes the vector with all of its components set to absolute values, the optimality conditions (23) can be written in the following equivalent form:

$$\begin{aligned}
\frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} &= \left(\frac{1}{(1-u)}\left(-\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \boldsymbol{e}_l\right)\right)_+ \\
&= \frac{1}{2(1-u)}\left(-\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \boldsymbol{e}_l\right. \\
&\quad \left. + \left|-\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \boldsymbol{e}_l\right|\right), \\
\frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} &= \left(\frac{1}{u}\left(\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right)\right)_+ \\
&= \frac{1}{2u}\left(\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right. \\
&\quad \left. + \left|\left(F - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right|\right).
\end{aligned} \tag{25}$$

Next, let $\boldsymbol{\alpha} = \boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2$, we obtain the following absolute value equation problem:

$$\begin{aligned}
\frac{\boldsymbol{\alpha}}{cu^2(1-u)^2} &= \frac{1}{2(1-u)}\left(-\left(F - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} + \boldsymbol{e}_l\right. \\
&\quad \left. + \left|-\left(F - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} + \boldsymbol{e}_l\right|\right) \\
&\quad - \frac{1}{2u}\left(\left(F - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} - \boldsymbol{e}_l\right. \\
&\quad \left. + \left|\left(F - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} - \boldsymbol{e}_l\right|\right)
\end{aligned}$$

$$= \frac{1}{2u(1-u)}\left(-\left(F-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}+\boldsymbol{e}_l\right.$$
$$\left.+(2u-1)\left|\left(F-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right|\right), \tag{26}$$

$\Leftrightarrow$

$$\left(F+\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}=\boldsymbol{e}_l+(2u-1)\left|\left(F-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right|. \tag{27}$$

The problem (27) can be solved by a simple functional iterative method given by

$$\boldsymbol{\alpha}^{i+1}=\left(F+\frac{I}{cu(1-u)}\right)^{-1}\left(\boldsymbol{e}_l+(2u-1)\left|\left(F-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}^i-\boldsymbol{e}_l\right|\right),$$
$$i=0,1,2,\ldots, \tag{28}$$

After the optimal solution $\boldsymbol{\alpha}^*$ is obtained, we get the following linear asy-LapSVM classifier which is represented by the dual variables

$$f(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}+b=\boldsymbol{\alpha}^{*T}Y(D(I+\lambda D^TLD)^{-1}\boldsymbol{x}+\boldsymbol{e}). \tag{29}$$

3.2 Nonlinear asy-LapSVM

The same as the linear case, the asymmetric squared loss (4) is exploited in the proposed nonlinear asy-LapSVM, and the margin measured by the expectile distance is maximized by optimizing with respect to both the weight vector $\boldsymbol{w}$ and the bias term b of the nonlinear asy-LapSVM classifier $f(\boldsymbol{x})=\boldsymbol{w}^T\phi(\boldsymbol{x})+b$, where $\phi(\cdot)$ is a nonlinear mapping function from a low-dimensional input space to a high-dimensional Hilbert space $\mathcal{H}$. Thus, the regularization term $\|f\|_k^2$ can be expressed as:

$$\|f\|_k^2=\frac{1}{2}(\|\boldsymbol{w}\|^2+b^2). \tag{30}$$

As for the manifold regularization term $\|f\|_{\mathcal{I}}^2$, on the basis of the constructed graph laplacian matrix L, the manifold regularization term is defined as:

$$\|f\|_{\mathcal{I}}^2=\mathbf{f}^TL\mathbf{f}=\boldsymbol{w}^TD_1^TLD_1\boldsymbol{w}, \tag{31}$$

where $D_1=[\phi(\boldsymbol{x}_1),\ldots,\phi(\boldsymbol{x}_n)]^T$ includes all of the labeled and unlabeled samples in $\mathcal{S}$ and $\mathbf{f}=[f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_n)]^T=[\boldsymbol{w}^T\phi(\boldsymbol{x}_1),\ldots,\boldsymbol{w}^T\phi(\boldsymbol{x}_n)]^T=D_1\boldsymbol{w}$, in which we deliberately reduce the bias term b for convenience.

By substituting the asymmetric squared loss (4), the regularization term (30) and the manifold regularization term (31) into the the semi-supervised manifold regularization learning framework (7), the primal problem of the nonlinear asy-LapSVM can be formulated as

$$
\begin{aligned}
\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}(\|\boldsymbol{w}\|^2 + b^2) + \tfrac{c}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} + \tfrac{\lambda}{2}\boldsymbol{w}^T D_1^T L D_1 \boldsymbol{w} \\
\text{s.t.} \quad & Y(D_1\boldsymbol{w} + \boldsymbol{e}b) \geq \boldsymbol{e}_l - \tfrac{1}{u}\boldsymbol{\xi}, \\
& Y(D_1\boldsymbol{w} + \boldsymbol{e}b) \leq \boldsymbol{e}_l + \tfrac{1}{1-u}\boldsymbol{\xi},
\end{aligned} \tag{32}
$$

where the constant u, the parameters c and λ, the vectors $\boldsymbol{\xi}$, $\boldsymbol{e}_l$ and $\boldsymbol{e}$, the matrix Y have the same meanings as the ones in problems (12).

According to the Representer Theorem (Belkin et al. 2006), $\boldsymbol{w}$ can be expressed as $\boldsymbol{w} = \sum_{i=1}^{n} \rho_i \phi(\boldsymbol{x}_i) = D_1^T \boldsymbol{\rho}$, where $\boldsymbol{\rho} \in R^n$ is a parameter vector. Then the terms containing $\boldsymbol{w}$ in the optimization problem (32) can be rewritten as

$$
\frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}(D_1^T\boldsymbol{\rho})^T(D_1^T\boldsymbol{\rho}) = \frac{1}{2}\boldsymbol{\rho}^T K \boldsymbol{\rho}, \tag{33}
$$

$$
\frac{\lambda}{2}\boldsymbol{w}^T D_1^T L D_1 \boldsymbol{w} = \frac{\lambda}{2}(D_1^T\boldsymbol{\rho})^T D_1^T L D_1 (D_1^T\boldsymbol{\rho}) = \frac{\lambda}{2}\boldsymbol{\rho}^T K L K \boldsymbol{\rho}, \tag{34}
$$

where $K \in R^{n \times n}$ is a Gram matrix with elements $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Based on the above analysis, the nonlinear optimization problem (32) can be converted into the following form:

$$
\begin{aligned}
\min_{\boldsymbol{\rho},b,\boldsymbol{\xi}} \quad & \tfrac{1}{2}\boldsymbol{\rho}^T F_1 \boldsymbol{\rho} + \tfrac{1}{2}b^2 + \tfrac{c}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} \\
\text{s.t.} \quad & Y(K\boldsymbol{\rho} + \boldsymbol{e}b) \geq \boldsymbol{e}_l - \tfrac{1}{u}\boldsymbol{\xi}, \\
& Y(K\boldsymbol{\rho} + \boldsymbol{e}b) \leq \boldsymbol{e}_l + \tfrac{1}{1-u}\boldsymbol{\xi},
\end{aligned} \tag{35}
$$

where $F_1 = K + \lambda K L K$ is a symmetric positive semi-definite matrix since the Gram matrix K and the graph Laplacian L are two symmetric positive semi-definite matrices.

The Lagrange function of the optimization problem (35) can be written as

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\rho}, \boldsymbol{\xi}, \boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2) = {} & \frac{1}{2}\boldsymbol{\rho}^T F_1 \boldsymbol{\rho} + \frac{1}{2}b^2 + \frac{c}{2}\boldsymbol{\xi}^T\boldsymbol{\xi} - \boldsymbol{\alpha}_1^T\left[Y(K\boldsymbol{\rho} + \boldsymbol{e}b) - \boldsymbol{e}_l + \frac{1}{u}\boldsymbol{\xi}\right] \\
& - \boldsymbol{\alpha}_2^T\left[-Y(K\boldsymbol{\rho} + \boldsymbol{e}b) + \boldsymbol{e}_l + \frac{1}{1-u}\boldsymbol{\xi}\right],
\end{aligned} \tag{36}
$$

where $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are Lagrange parameter vectors.

Differentiating the Lagrange function (36) with respect to $\boldsymbol{\rho}$, b, $\boldsymbol{\xi}$ and setting them equal to zero, we can obtain

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} = F_1\boldsymbol{\rho} - KY^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) = 0 \Rightarrow \boldsymbol{\rho} = F_1^{-1}KY^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2), \tag{37}
$$

$$
\frac{\partial \mathcal{L}}{\partial b} = b - \boldsymbol{e}^T Y^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) = 0 \Rightarrow b = \boldsymbol{e}^T Y^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2), \tag{38}
$$

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{\xi}} = c\boldsymbol{\xi} - \left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right) = 0 \Rightarrow \boldsymbol{\xi} = \frac{1}{c}\left(\frac{\boldsymbol{\alpha}_1}{u} + \frac{\boldsymbol{\alpha}_2}{1-u}\right). \tag{39}
$$

It is worth noting that although the matrix F_1 in (37) is always positive semi-definite, it may not be well conditioned in some situations. In the light of the idea of regularization, F_1^{-1} can be revised by $(\epsilon I + F_1)^{-1}$, where ϵI $(\epsilon > 0)$ is a regularization term. In the following, we shall continue to use F_1^{-1} with the understanding that, if need be, $(\epsilon I + F_1)^{-1}$ is to be used.

Then the following dual problem of (35) can be derived by substituting the above Eqs. (37)–(39) into the Lagrangian function (36)

$$\begin{aligned} \min_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2} \ & \tfrac{1}{2}(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)^T F_2(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) + \tfrac{1}{2c}\left(\tfrac{\boldsymbol{\alpha}_1}{u}+\tfrac{\boldsymbol{\alpha}_2}{1-u}\right)^T\left(\tfrac{\boldsymbol{\alpha}_1}{u}+\tfrac{\boldsymbol{\alpha}_2}{1-u}\right) - \boldsymbol{e}_l^T(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) \\ \text{s.t.} \ \ & \boldsymbol{\alpha}_1 \geq 0\boldsymbol{e}_l, \\ & \boldsymbol{\alpha}_2 \geq 0\boldsymbol{e}_l. \end{aligned} \tag{40}$$

where $F_2 = Y(KF_1^{-1}K + ee^T)Y^T$.

We solve the dual problem (40) by a functional iterative method. Specifically, based on the Karush–Kuhn–Tucker (KKT) necessary and sufficient optimality conditions for the dual problem (40) , we have

$$\begin{aligned} 0\boldsymbol{e}_l &\leq \boldsymbol{\alpha}_1 \perp \left(F_2 + \frac{I}{cu^2}\right)\boldsymbol{\alpha}_1 - \left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}_2 - \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \\ 0\boldsymbol{e}_l &\leq \boldsymbol{\alpha}_2 \perp \left(F_2 + \frac{I}{c(1-u)^2}\right)\boldsymbol{\alpha}_2 - \left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}_1 + \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \end{aligned} \tag{41}$$

$\Leftrightarrow$

$$\begin{aligned} 0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)} \perp \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)} + \left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) - \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \\ 0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_2}{cu(1-u)^2} \perp \frac{\boldsymbol{\alpha}_2}{cu(1-u)^2} - \left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) + \boldsymbol{e}_l \geq 0\boldsymbol{e}_l, \end{aligned} \tag{42}$$

$\Leftrightarrow$

$$\begin{aligned} 0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} \perp \frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} + \frac{1}{(1-u)}\left(\left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right) \\ &\quad \geq 0\boldsymbol{e}_l, \\ 0\boldsymbol{e}_l &\leq \frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} \perp \frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} - \frac{1}{u}\left(\left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2) - \boldsymbol{e}_l\right) \geq 0\boldsymbol{e}_l. \end{aligned} \tag{43}$$

By exploiting the identity (24), the optimality conditions (43) can be written in the following equivalent form:

$$
\begin{aligned}
\frac{\boldsymbol{\alpha}_1}{cu^2(1-u)^2} &= \left(\frac{1}{(1-u)}\left(-\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)+\boldsymbol{e}_l\right)\right)_+\\
&= \frac{1}{2(1-u)}\left(-\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)+\boldsymbol{e}_l\right.\\
&\quad \left.+\left|-\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)+\boldsymbol{e}_l\right|\right),\\
\frac{\boldsymbol{\alpha}_2}{cu^2(1-u)^2} &= \left(\frac{1}{u}\left(\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)-\boldsymbol{e}_l\right)\right)_+\\
&= \frac{1}{2u}\left(\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)-\boldsymbol{e}_l\right.\\
&\quad \left.+\left|\left(F_2-\frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2)-\boldsymbol{e}_l\right|\right).
\end{aligned}
\tag{44}
$$

Let $\boldsymbol{\alpha}=\boldsymbol{\alpha}_1-\boldsymbol{\alpha}_2$, we obtain the following absolute value equation problem:

$$
\begin{aligned}
\frac{\boldsymbol{\alpha}}{cu^2(1-u)^2} &= \frac{1}{2(1-u)}\left(-\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}+\boldsymbol{e}_l\right.\\
&\quad \left.+\left|-\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}+\boldsymbol{e}_l\right|\right)\\
&\quad -\frac{1}{2u}\left(\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right.\\
&\quad \left.+\left|\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right|\right)\\
&= \frac{1}{2u(1-u)}\left(-\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}+\boldsymbol{e}_l\right.\\
&\quad \left.+(2u-1)\left|\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right|\right),
\end{aligned}
\tag{45}
$$

$\Leftrightarrow$

$$
\left(F_2+\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}=\boldsymbol{e}_l+(2u-1)\left|\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}-\boldsymbol{e}_l\right|. \tag{46}
$$

The problem (46) can be solved by the simple functional iterative method given by

$$
\begin{aligned}
\boldsymbol{\alpha}^{i+1} &= \left(F_2+\frac{I}{cu(1-u)}\right)^{-1}\left(\boldsymbol{e}_l+(2u-1)\left|\left(F_2-\frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}^i-\boldsymbol{e}_l\right|\right),\\
&\quad i=0,1,2,\ldots.
\end{aligned}
\tag{47}
$$

After the optimal solution $\boldsymbol{\alpha}^*$ is obtained, we get the following nonlinear asy-LapSVM classifier which is represented by the dual variables

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b = \sum_{i=1}^{n} \rho_i^* k(\boldsymbol{x}_i, \boldsymbol{x}) + b^* \tag{48}$$

where $\boldsymbol{\rho}^* = F_1^{-1} K Y^T \boldsymbol{\alpha}^*$ and $b^* = \boldsymbol{e}^T Y^T \boldsymbol{\alpha}^*$.

Next, we prove the convergence of the proposed functional iterative method in the following Theorem.

Theorem 1 *Assume that* $\boldsymbol{\alpha} \in R^l$ *is the solution of the absolute value Eq.* (46) *and* $0.5 < u < 1$, *then, for any starting vector* $\boldsymbol{\alpha}^0 \in R^l$, *the sequence of iterates generated by* (47) *will always converge to* $\boldsymbol{\alpha}$ *with linear rate of convergence.*

Proof According to the conditions given in the above theorem, by using (46) and (47), we get

$$\begin{aligned} &\left(F_2 + \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^{i+1} - \boldsymbol{\alpha}) \\ &\quad = (2u-1)\left(\left|\left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}^i - \boldsymbol{e}_l\right| - \left|\left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} - \boldsymbol{e}_l\right|\right), \\ &\qquad i = 0, 1, 2, \ldots. \end{aligned}$$

Since

$$\begin{aligned} &\left|\left(\left|\left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha}^i - \boldsymbol{e}_l\right| - \left|\left(F_2 - \frac{I}{cu(1-u)}\right)\boldsymbol{\alpha} - \boldsymbol{e}_l\right|\right)\right| \\ &\quad \leq \left|\left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^i - \boldsymbol{\alpha})\right|, \end{aligned}$$

we have

$$\begin{aligned} &\left\|\left(F_2 + \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^{i+1} - \boldsymbol{\alpha})\right\| \\ &\quad \leq (2u-1)\left\|\left(F_2 - \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^i - \boldsymbol{\alpha})\right\| \\ &\quad = (2u-1) \\ &\qquad \left\|\left(F_2 - \frac{I}{cu(1-u)}\right)\left(F_2 + \frac{I}{cu(1-u)}\right)^{-1}\left(F_2 + \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^i - \boldsymbol{\alpha})\right\| \\ &\quad \leq (2u-1)\left\|\left(F_2 - \frac{I}{cu(1-u)}\right)\left(F_2 + \frac{I}{cu(1-u)}\right)^{-1}\right\| \\ &\qquad \left\|\left(F_2 + \frac{I}{cu(1-u)}\right)(\boldsymbol{\alpha}^i - \boldsymbol{\alpha})\right\|. \end{aligned}$$

Let $\{q_1, \ldots, q_l\}$ be the set of the nonnegative eigenvalues of the positive semi-definite matrix F_2. Clearly, the eigenvalues of $(F_2 - \frac{I}{cu(1-u)})(F_2 + \frac{I}{cu(1-u)})^{-1}$ will become:

$$\left\{\frac{q_1-\frac{1}{cu(1-u)}}{q_1+\frac{1}{cu(1-u)}},\ldots,\frac{q_l-\frac{1}{cu(1-u)}}{q_l+\frac{1}{cu(1-u)}}\right\},$$

So, $(2u-1)||(F_2-\frac{1}{cu(1-u)})(F_2+\frac{1}{cu(1-u)})^{-1}|| = (2u-1)\max\{|\frac{q_1-\frac{1}{cu(1-u)}}{q_1+\frac{1}{cu(1-u)}}|,\ldots,$ $|\frac{q_l-\frac{1}{cu(1-u)}}{q_l+\frac{1}{cu(1-u)}}|\} < 1$ is always true. This shows the **Theorem** 1 holds. □

Although we only prove the convergence of the nonlinear case, the convergence of the linear case also can be easily gotten in the similar way. In the end, we summary the proposed nonlinear asy-LapSVM as follows:

Input. A set of l labeled data and m unlabeled data $\mathcal{S} = \mathcal{S}_l \cup \mathcal{S}_u = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l} \cup \{\boldsymbol{x}_i\}_{i=l+1}^{n}$, the constant u, the positive parameters c and λ, the kernel parameter σ, the number of the nearest neighbors p, the vector of ones of l dimension $\boldsymbol{e}_l$, the vector of ones of n dimension e, the matrix $Y \in R^{l\times n}$ with elements $Y_{ii} = y_i$ according to the class of each sample, the maximum number of iterations $maxI = 15$, the tolerance $tol = 10^{-4}$, and initial iteration number $t = 0$.

Step 1. Compute the graph Laplacian matrix $L = G - W$, in which W denotes the adjacency matrix which is constructed by p nearest neighbors with n nodes in $\mathcal{S}$ and edge weights W_{ij} are calculated by binary weights. G denotes the diagonal matrix with its diagonal elements $G_{ii} = \sum_{j=1}^{n} W_{ij}$ representing the weight degree of the vertex i.

Step 2. Choose a proper kernel function $k(\cdot,\cdot)$, and compute the Gram matrix K.

Step 3. Compute the matrices

$$F_2 = Y(K(K+\lambda KLK)^{-1}K + \boldsymbol{e}\boldsymbol{e}^T)Y^T,$$
$$Q = \left(F_2 + \frac{I}{cu(1-u)}\right)^{-1},$$
$$Q_1 = F_2 - \frac{I}{cu(1-u)},$$

where $I \in \mathcal{R}^{l\times l}$ is an identity matrix.

Step 4. Compute the initial vector

$$\boldsymbol{\alpha}^0 = Q\boldsymbol{e}_l.$$

Step 5. Compute $\boldsymbol{\alpha}^{t+1}$ via

$$\boldsymbol{\alpha}^{t+1} = Q(\boldsymbol{e}_l + (2u-1)|Q_1\boldsymbol{\alpha}^t - \boldsymbol{e}_l|), \quad i = 0, 1, 2, \ldots.$$

Step 6. If $\|w^{t+1} - w^t\| < tol$ or $t > maxI$, stop; else, let $t = t+1$ and goto **Step 5**.

Step 7. Compute the bias term $b = \boldsymbol{e}^T Y^T \boldsymbol{\alpha}^t$.

Step 8. Derive the nonlinear asy-LapSVM classifier

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) + b = \sum_{i=1}^{n} \rho_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b,$$

where $\boldsymbol{\rho} = (K + \lambda K L K)^{-1} K Y^T \boldsymbol{\alpha}^t$.

4 Numerical experiments

In this section, we will verify the validity of the proposed asy-LapSVM by comparing it with several closely relevant algorithms on different kinds of datasets with noise of different variances. After the experimental setup and data are described in Sects. 4.1 and 4.2, we carefully analyze the performance of our proposed asy-LapSVM in the following subsections.

4.1 Experimental setup

We compare the proposed asy-LapSVM with a group of popular supervised learning (SL) algorithms [SVM (Cristianini and Shawe-Taylor 2000), regularized least squares (RLS) (Belkin et al. 2006), extreme learning machine (ELM) (Huang et al. 2006), aLS-SVM (Huang et al. 2014)] and semi-supervised learning (SSL) algorithms (LapSVM (Belkin et al. 2006), Laplacian RLS (LapRLS) (Belkin et al. 2006), semi-supervised ELM (SSELM) (Huang et al. 2014)). In the experiments, the sigmoid activation function is used for the two algorithms with ELM, and the number of hidden neurons is set to 1000. For the remaining algorithms and the proposed asy-LapSVM, the RBF kernel (6) which meets the Mercer's theorem is employed. For the choice of parameters, a set of possible values are first predefined, and 5-fold cross validation is used for all the compared methods. In order to avoid the bias caused by different sample partition, 5-fold cross validation with different partition repeats 10 times at random, and the average testing accuracies are computed to obtain the final evaluation results. The ranges of the five parameters are listed as follows:

- $c : \{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$,
- $\lambda : \{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$,
- $p : \{5, 10, 15, 20\}$,
- $\sigma : \{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$,
- $u : \{0.55, 0.65, 0.75, 0.83, 0.95, 0.99\}$.

In the model training, we randomly select 10% training samples as labeled samples, and the remaining training samples are regarded as unlabeled samples. For supervised algorithms, we use only the selected labeled samples to train the classifier, while for semi-supervised algorithms, we use all the training set with both labeled and unlabeled samples to train the classifier. The simulations of all the algorithms are carried out in MATLAB R2016a on a personal computer with system configuration: Intel Core i7 (3.6 GHz) and 8 GB random access memory. We use the quadratic programming

(QP) solver embedded in MATLAB to solve all the QP problems and the MATLAB operation "\" to realize the matrix inverse involved in the algorithms.

4.2 Data specification

In this subsection, several publicly available datasets including one artificial dataset, six UCI[1] datasets and five image datasets are employed to study the performance of the proposed asy-LapSVM. The G50C[2] artificial dataset is generated from two unit covariance normal distributions with equal probabilities. The class means are adjusted so that the Bayes error is 5%. For the image datasets, the Coil-20[3] dataset includes 1440 grey-scale images sampled from 20 objects, and each object has 72 images with the size of 32 × 32. The USPS[4] handwritten digit dataset consists of 7,291 training images and 2,007 test images with size of 16 × 16. The YaleB[5] face recognition dataset admittedly contains manifold structures includes 2414 grey human facial images sampled from 38 persons, and each person has about 64 different images with size of 32×32. The Multiple Features (MF)[6] handwritten numeral dataset consists of 10 classes and 2,000 samples. The NUS-WIDE-OBJECT (NWO)[7] dataset contains 31 categories and 30,000 network images (17,927 for training and 12,703 for testing) created by the media search laboratory of National University of Singapore.

In order to fit into the binary-class environment, we choose two pairwise digits in the USPS dataset, three pairwise objects in the Coil-20 dataset, three pairwise facial images in the YaleB face recognition dataset, eight pairwise subsets in the MF dataset, and seven pairwise subsets in the NWO dataset to constitute 23 binary-class image datasets, namely, USPS1, USPS2, Coil-201, Coil-202, Coil-203, YaleB1, YaleB2, YaleB3, MF1, MF2, MF3, MF4, MF5, MF6, MF7, MF8, NWO1, NWO2, NWO3, NWO4, NWO5, NWO6, and NWO7. To facilitate the calculation, the well-known principal component analysis (PCA) is adopted to preprocess these image datasets, where the dimension and accumulative contribution rate after processing are displayed in Table 1. The features of each dataset are corrupted by zero-mean Gaussian noise, and the ratio of the variance of noise to that of the features, denoted as z, is set to be 0 (i.e., noise-free), 0.05, and 0.1. The training and test data are aggravated by the same noise. The important statistics of these employed datasets are summarized in Table 1, where "No." denotes the ordinal numbers of the datasets. "Selected class" and "Selected size" denote the classes and sizes selected from the original datasets, respectively. "Dimension" denotes the number of the original features.

[1] http://archive.ics.uci.edu/ml/index.php.

[2] http://people.cs.uchicago.edu/~vikass/manifoldregularization.html.

[3] http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[4] http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html.

[5] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html.

[6] http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

[7] http://lms.comp.nus.edu.sg/research/NUS-WIDE.html.

Table 1 Information of the employed datasets

No.	Datasets	Selected classes	Selected size	Dimension	Accumulative contribution rate	Reduced dimension
1	G50C	–	(275, 275)	50	–	–
2	Blood	–	(570, 178)	4	–	–
3	Spect	–	(55, 212)	22	–	–
4	Ionosphere	–	(126, 225)	34	–	–
5	Fourclass	–	(307, 307)	2	–	–
6	Monks2	–	(395, 206)	6	–	–
7	Pima	–	(500, 268)	8	–	–
8	Coil-201	(1, 2)	(72, 72)	1024	95.15%	30
9	Coil-202	(2, 4)	(72, 72)	1024	95.12%	32
10	Coil-203	(2, 8)	(72, 72)	1024	95.20%	26
11	USPS1	(3, 8)	(731, 645)	256	95.09%	38
12	USPS2	(1, 2)	(1194, 1005)	256	95.05%	22
13	YaleB1	(1, 2)	(64, 64)	1024	93.04%	8
14	YaleB2	(5, 10)	(64, 64)	1024	92.34%	8
15	YaleB3	(21, 36)	(64, 64)	1024	89.16%	8
16	MF1	(1, 2)	(200, 200)	649	95.05%	104
17	MF2	(1, 3)	(200, 200)	649	95.11%	109
18	MF3	(1, 4)	(200, 200)	649	95.07%	110
19	MF4	(2, 3)	(200, 200)	649	95.05%	113
20	MF5	(2, 4)	(200, 200)	649	95.02%	117
21	MF6	(3, 4)	(200, 200)	649	95.01%	120
22	MF7	(4, 5)	(200, 200)	649	95.10%	112
23	MF8	(5, 6)	(200, 200)	649	95.06%	114
24	NWO1	(1, 2)	(351, 964)	1134	92.03%	380
25	NWO2	(1, 3)	(351, 1336)	1134	92.02%	414
26	NWO3	(1, 4)	(351, 90)	1134	92.01%	232
27	NWO4	(2, 3)	(964, 1336)	1134	92.02%	445
28	NWO5	(2, 5)	(964, 674)	1134	92.05%	413
29	NWO6	(3, 5)	(1336, 674)	1134	92.02%	450
30	NWO7	(5, 6)	(674, 606)	1134	92.25%	390

4.3 Experimental results and discussions

The results of the eight algorithms on the aforementioned datasets with different proportion noise are shown in Table 2, where the best ones are presented in bold. In addition, the learning time (second) taken by each method under the optimal parameters obtained by one 5-fold cross validation is listed in Table 3. For better interpretation, the comparison results of the time among the proposed asy-LapSVM and other three

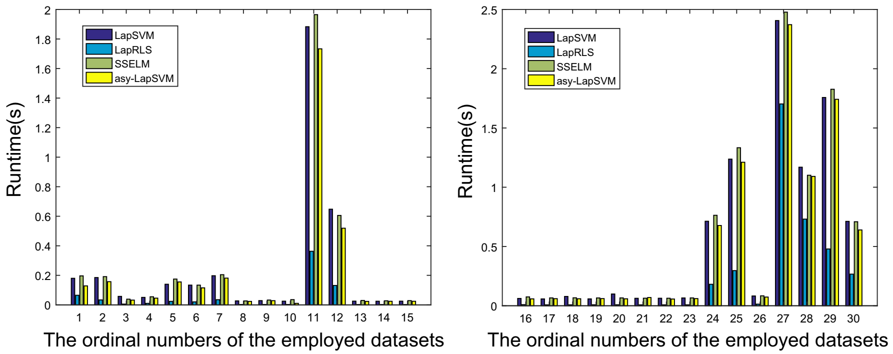

Fig. 2 The learning time (s) of the proposed asy-LapSVM and the three SSL algorithms on the employed datasets with $z = 0$

SSL algorithms in the absence of noise are visualized in Fig. 2, where indexes on the horizontal axis denote the ordinal numbers of the employed datasets.

From the experimental results and experimental runtime reported in Tables 2 and 3 and Fig. 2, we can get the following conclusions:

(1) The SSL algorithms including the proposed asy-LapSVM perform better than the corresponding SL ones in most cases, which might be attribute to the manifold regularization which helps classifiers gain more geometric information embedded in unlabeled data to achieve better performance.

(2) The proposed asy-LapSVM stands out the other three SSL algorithms (LapSVM, LapRLS and SSELM) on 54 cases, which implies that adopting the asymmetric squared loss can improve the prediction accuracy, especially for noise-corrupted data. For example, the accuracy of asy-LapSVM in YaleB3 dataset with $z = 0.10$ is 92.55%, and nearly increases by 2.55% compared with the second best method LapSVM.

(3) From Table 3, it can be easy to know that the SSL algorithms generally consume longer time than the SL algorithms. The major reason is that, in addition to the few labeled data which is considered in SL algorithms, the SSL algorithms need to consider lots of unlabeled data.

(4) From Fig. 2, we can see that although the presented functional iterative method for the proposed asy-LapSVM costs more time than LapRLS, it shows faster learning speed than the other two SSL algorithms in most cases, which implies the feasibility of the presented functional iterative method.

To sum up, the proposed method is always the best in terms of classification accuracy and also preferable in terms of learning time among all the SSL algorithms, which implies that the asy-LapSVM is a powerful SSL algorithm for semi-supervised binary-class classification problems in the presence of noise.

Table 2 Comparison of accuracies (%) of eight algorithms on the employed datasets

Dataset	z	SVM *Accuracy*	RLS *Accuracy*	ELM *Accuracy*	ALSSVM *Accuracy*	LapSVM *Accuracy*	LapRLS *Accuracy*	SSELM *Accuracy*	asy-LapSVM *Accuracy*
G50C	0.00	**91.50 ± 2.40**	91.33 ± 2.38	90.10 ± 2.01	91.27 ± 2.09	91.33 ± 1.88	90.14 ± 2.51	88.31 ± 1.56	90.54 ± 7.66
	0.05	89.56 ± 2.49	89.36 ± 2.20	89.31 ± 1.89	**90.54 ± 2.29**	**90.54 ± 2.22**	**90.54 ± 2.29**	86.87 ± 2.08	89.53 ± 8.58
	0.10	90.14 ± 2.45	90.51 ± 2.38	89.51 ± 2.44	**90.96 ± 2.62**	89.67 ± 2.95	88.36 ± 2.33	89.20 ± 2.76	89.96 ± 3.18
Blood	0.00	78.37 ± 1.67	79.24 ± 1.53	78.74 ± 1.99	78.59 ± 1.33	79.14 ± 1.45	78.74 ± 2.12	78.74 ± 1.994	**79.47 ± 0.96**
	0.05	77.33 ± 1.12	78.49 ± 1.29	77.71 ± 1.64	78.43 ± 1.99	78.24 ± 1.92	78.45 ± 1.37	78.66 ± 1.83	**79.20 ± 0.92**
	0.10	77.38 ± 0.92	77.82 ± 1.28	77.43 ± 1.03	**79.43 ± 1.99**	78.74 ± 1.26	78.52 ± 1.99	78.66 ± 1.43	78.22 ± 0.47
Spect	0.00	80.44 ± 0.31	80.56 ± 0.31	81.41 ± 3.05	82.58 ± 3.29	80.44 ± 2.67	83.24 ± 3.10	**83.58 ± 3.05**	**83.58 ± 3.29**
	0.05	81.50 ± 0.45	80.89 ± 0.22	81.31 ± 2.49	82.33 ± 2.98	80.18 ± 2.13	83.33 ± 2.42	83.42 ± 2.85	**84.20 ± 3.26**
	0.10	80.50 ± 0.27	80.50 ± 0.35	80.35 ± 1.80	82.09 ± 3.46	80.44 ± 2.31	**83.33 ± 3.63**	83.26 ± 3.76	83.18 ± 3.12
Ionosphere	0.00	**90.57 ± 0.47**	84.25 ± 2.27	82.26 ± 2.58	90.19 ± 2.82	89.58 ± 3.29	90.24 ± 3.14	89.08 ± 4.65	89.43 ± 3.37
	0.05	88.52 ± 0.49	82.26 ± 4.68	81.40 ± 3.56	88.77 ± 3.76	88 ± 2.98	89.10 ± 3.24	88.58 ± 3.54	**89.23 ± 2.12**
	0.10	87.83 ± 0.74	83.210 ± 3.97	81.36 ± 3.46	88.96 ± 2.98	88.58 ± 2.52	**89.33 ± 4.23**	88.25 ± 3.16	89.05 ± 3.25
Fourclss	0.00	95.25 ± 2.19	94.84 ± 2.39	89.10 ± 4.21	95.25 ± 1.55	95.25 ± 1.42	94.56 ± 1.36	92.85 ± 2.55	**97.42 ± 1.22**
	0.05	82.93 ± 1.55	83.87 ± 3.66	81.63 ± 3.43	84.25 ± 3.44	84.25 ± 2.67	84.25 ± 3.22	83.71 ± 3.15	**84.95 ± 1.78**
	0.10	76.97 ± 1.57	77.47 ± 2.54	76.01 ± 1.84	78.33 ± 2.22	74.25 ± 2.54	77.25 ± 2.78	73.55 ± 2.82	**78.92 ± 1.29**
Monks2	0.00	72.18 ± 0.42	67.29 ± 2.00	64.39 ± 2.23	69.78 ± 1.38	67.39 ± 2.56	69.39 ± 2.35	70.39 ± 3.76	**72.55 ± 0.51**
	0.05	70.42 ± 0.32	66.91 ± 1.72	64.55 ± 2.47	69.23 ± 1.77	69.23 ± 1.58	69.23 ± 1.77	69.23 ± 1.36	**70.61 ± 1.07**
	0.10	69.21 ± 0.33	66.52 ± 0.91	64.31 ± 3.51	67.33 ± 2.65	64.45 ± 2.85	69.07 ± 2.01	**69.82 ± 1.31**	69.50 ± 1.92
Pima	0.00	75.06 ± 2.35	76.15 ± 2.45	74.94 ± 2.37	74.26 ± 2.33	75.08 ± 2.72	75.44 ± 2.37	74.94 ± 2.37	**76.94 ± 2.37**
	0.05	74.85 ± 3.35	**76.06 ± 3.48**	74.42 ± 3.13	73.38 ± 3.46	75.57 ± 3.28	75.22 ± 3.28	74.42 ± 3.28	74.42 ± 3.28
	0.10	73.64 ± 3.38	**74.72 ± 3.41**	73.39 ± 3.37	73.92 ± 2.65	73.39 ± 3.24	73.39 ± 3.65	73.39 ± 3.74	74.22 ± 2.12
Coil-201	0.00	73.30 ± 1.04	73.64 ± 4.82	70.18 ± 3.65	70.55 ± 4.25	73.55 ± 6.87	73.30 ± 2.47	75.23 ± 3.13	**75.98 ± 2.98**
	0.05	73.00 ± 1.08	72.27 ± 3.52	70.45 ± 4.83	70.76 ± 4.23	73.45 ± 3.95	70.45 ± 4.83	**75.23 ± 2.54**	71.85 ± 2.83

Table 2 continued

Dataset	z	SVM *Accuracy*	RLS *Accuracy*	ELM *Accuracy*	ALSSVM *Accuracy*	LapSVM *Accuracy*	LapRLS *Accuracy*	SSELM *Accuracy*	asy-LapSVM *Accuracy*
	0.10	73.40 ± 0.90	73.64 ± 4.69	70.59 ± 5.53	70.17 ± 4.46	75.59 ± 5.35	**75.73 ± 2.16**	72.86 ± 1.81	73.85 ± 2.83
Coil-202	0.00	75.60 ± 7.89	71.14 ± 8.16	67.86 ± 6.35	**75.86 ± 7.54**	74.23 ± 7.18	71.96 ± 7.44	74.23 ± 4.85	72.55 ± 1.02
	0.05	**79.60 ± 5.36**	71.82 ± 7.74	69.91 ± 6.20	70.15 ± 6.15	70.22 ± 6.76	71.56 ± 6.33	72.27 ± 4.48	70.61 ± 2.14
	0.10	**72.10 ± 8.13**	70.68 ± 6.73	68.09 ± 6.24	69.46 ± 7.33	69.86 ± 7.54	69.77 ± 6.65	70.95 ± 2.41	69.50 ± 1.82
Coil-203	0.00	70.00 ± 8.30	64.32 ± 4.63	67.18 ± 4.45	**79.00 ± 7.15**	77.27 ± 6.41	77.55 ± 6.65	70.23 ± 4.37	70.60 ± 5.30
	0.05	68.60 ± 8.30	64.55 ± 4.31	69.09 ± 4.07	**74.73 ± 3.95**	64.37 ± 4.45	68.18 ± 4.37	70.68 ± 3.88	70.50 ± 2.30
	0.10	68.80 ± 8.89	68.64 ± 6.59	71.18 ± 5.99	68.18 ± 4.55	**72.64 ± 4.73**	68.18 ± 4.37	69.77 ± 7.58	68.50 ± 3.30
USPS1	0.00	97.85 ± 1.11	97.56 ± 1.22	95.67 ± 1.16	97.15 ± 1.67	97.15 ± 1.27	97.35 ± 1.12	98.80 ± 1.35	**98.84 ± 0.56**
	0.05	96.71 ± 1.82	96.71 ± 1.82	94.71 ± 0.88	95.43 ± 1.43	95.88 ± 1.08	97.33 ± 1.31	96.50 ± 1.72	**98.97 ± 0.65**
	0.10	96.21 ± 1.10	96.21 ± 1.10	92.55 ± 1.75	96.15 ± 1.52	94.95 ± 1.41	96.25 ± 1.65	96.50 ± 1.84	**98.64 ± 0.84**
USPS2	0.00	**99.92 ± 0.01**	98.85 ± 0.44	99.92 ± 0.01	99.85 ± 0.01	99.21 ± 0.12	99.85 ± 0.01	99.88 ± 0.16	99.85 ± 0.15
	0.05	**99.88 ± 0.01**	98.24 ± 0.22	99.87 ± 0.01	99.85 ± 0.01	98.80 ± 0.45	99.09 ± 0.31	99.85 ± 0.01	99.83 ± 0.22
	0.10	99.67 ± 0.19	99.03 ± 0.15	99.50 ± 0.31	**99.85 ± 0.22**	98.74 ± 0.18	98.80 ± 0.45	**99.85 ± 0.01**	**99.85 ± 0.03**
YaleB1	0.00	81.50 ± 7.19	83.25 ± 6.35	79.60 ± 6.55	83.25 ± 6.40	82.50 ± 7.15	87.50 ± 7.32	92.50 ± 2.76	**95.50 ± 3.07**
	0.05	80.25 ± 7.12	81.75 ± 6.78	78.45 ± 6.65	81.50 ± 6.33	82.50 ± 6.65	82.50 ± 8.35	92.50 ± 3.14	**95.50 ± 2.30**
	0.10	80.25 ± 9.31	81.75 ± 9.26	78.50 ± 7.30	81.50 ± 7.05	87.50 ± 7.12	87.50 ± 9.25	90.75 ± 4.35	**93.30 ± 4.47**
YaleB2	0.00	86.25 ± 6.15	86.25 ± 7.10	83.15 ± 10.08	86.25 ± 6.89	95.50 ± 3.41	92.50 ± 2.55	92.50 ± 2.08	**97.00 ± 1.53**
	0.05	85.00 ± 9.28	85.75 ± 8.34	83.35 ± 9.61	86.25 ± 6.56	95.50 ± 3.74	**97.50 ± 2.15**	90.75 ± 4.54	**97.50 ± 2.32**
	0.10	86.00 ± 7.75	87.00 ± 6.95	79.65 ± 9.08	87.00 ± 7.85	86.25 ± 6.32	90.00 ± 4.33	90.75 ± 5.17	**95.50 ± 3.51**
YaleB3	0.00	85.50 ± 4.97	86.05 ± 4.59	87.50 ± 7.69	90.00 ± 1.25	87.50 ± 2.76	90.00 ± 2.31	88.00 ± 2.42	**95.50 ± 2.33**
	0.05	82.50 ± 7.19	83.85 ± 6.88	77.80 ± 9.04	82.50 ± 1.87	87.50 ± 2.44	90.00 ± 2.33	88.50 ± 2.43	**92.55 ± 2.56**
	0.10	87.25 ± 5.06	87.25 ± 5.20	81.54 ± 5.71	82.50 ± 2.02	90.00 ± 3.67	87.50 ± 4.21	88.50 ± 2.72	**92.55 ± 3.16**
MF1	0.00	99.33 ± 0.66	99.58 ± 0.59	99.32 ± 0.67	99.42 ± 0.56	**100.0 ± 0.00**	99.42 ± 0.56	99.83 ± 0.35	**100.0 ± 0.00**

Table 2 continued

Dataset	z	SVM *Accuracy*	RLS *Accuracy*	ELM *Accuracy*	ALSSVM *Accuracy*	LapSVM *Accuracy*	LapRLS *Accuracy*	SSELM *Accuracy*	asy-LapSVM *Accuracy*
	0.05	99.33 ± 0.66	99.58 ± 0.59	99.33 ± 0.68	99.50 ± 0.58	**100.0 ± 0.00**	99.42 ± 0.56	99.83 ± 0.35	99.83 ± 0.01
	0.10	99.33 ± 0.66	99.58 ± 0.59	99.37 ± 0.67	99.33 ± 0.66	**100.0 ± 0.00**	99.33 ± 0.66	99.83 ± 0.35	99.83 ± 0.01
MF2	0.00	99.67 ± 0.43	99.92 ± 0.26	99.63 ± 0.41	99.67 ± 0.43	**100.0 ± 0.00**	99.67 ± 0.41	99.83 ± 0.35	**100.0 ± 0.00**
	0.05	99.67 ± 0.43	99.92 ± 0.26	99.65 ± 0.40	99.67 ± 0.43	**100.0 ± 0.00**	99.67 ± 0.41	99.83 ± 0.01	**100.0 ± 0.00**
	0.10	99.67 ± 0.43	99.92 ± 0.26	99.60 ± 0.42	99.67 ± 0.43	**100.0 ± 0.00**	99.60 ± 0.42	99.83 ± 0.01	**100.0 ± 0.00**
MF3	0.00	99.17 ± 0.68	99.25 ± 0.61	99.20 ± 0.58	99.25 ± 0.61	**100.0 ± 0.00**	99.33 ± 0.61	99.42 ± 0.56	**100.0 ± 0.00**
	0.05	99.17 ± 0.68	99.25 ± 0.61	99.18 ± 0.61	99.25 ± 0.61	**100.0 ± 0.00**	99.42 ± 0.56	99.58 ± 0.31	99.58 ± 0.43
	0.10	99.25 ± 0.73	99.42 ± 0.69	99.20 ± 0.66	99.33 ± 0.66	**100.0 ± 0.00**	99.42 ± 0.56	99.67 ± 0.20	99.67 ± 0.31
MF4	0.00	99.58 ± 0.44	99.67 ± 0.58	99.50 ± 0.49	99.58 ± 0.44	**100.0 ± 0.00**	99.92 ± 0.01	**100.0 ± 0.00**	**100.0 ± 0.00**
	0.05	99.58 ± 0.44	99.75 ± 0.40	99.53 ± 0.43	99.58 ± 0.44	**100.0 ± 0.00**	99.92 ± 0.01	99.67 ± 0.33	99.92 ± 0.01
	0.10	99.58 ± 0.44	99.83 ± 0.35	99.48 ± 0.47	99.67 ± 0.33	**100.0 ± 0.00**	99.92 ± 0.01	99.67 ± 0.33	99.92 ± 0.01
MF5	0.00	98.50 ± 1.41	99.17 ± 0.88	98.22 ± 1.52	98.58 ± 1.25	99.10 ± 0.01	99.17 ± 0.43	99.17 ± 0.61	**99.20 ± 0.35**
	0.05	98.58 ± 1.25	**99.25 ± 0.73**	98.27 ± 1.48	98.58 ± 1.25	99.10 ± 0.01	99.17 ± 0.43	99.10 ± 0.35	99.07 ± 0.31
	0.10	98.67 ± 1.05	99.33 ± 0.66	98.25 ± 1.52	98.83 ± 0.98	**100.0 ± 0.00**	99.10 ± 0.43	99.17 ± 0.43	99.10 ± 0.27
MF6	0.00	99.42 ± 0.56	99.58 ± 0.31	99.42 ± 0.67	99.42 ± 0.56	**99.75 ± 0.01**	99.58 ± 0.28	**99.75 ± 0.16**	**99.75 ± 0.16**
	0.05	99.42 ± 0.56	99.58 ± 0.31	99.42 ± 0.73	99.42 ± 0.56	**99.75 ± 0.01**	99.58 ± 0.28	**99.75 ± 0.16**	99.42 ± 0.33
	0.10	99.50 ± 0.48	**99.67 ± 0.28**	99.33 ± 0.64	99.50 ± 0.58	99.58 ± 0.01	99.33 ± 0.64	99.17 ± 0.43	99.42 ± 0.33
MF7	0.00	99.33 ± 0.86	**99.58 ± 0.59**	99.22 ± 0.88	99.50 ± 0.58	99.25 ± 0.56	99.25 ± 0.73	99.25 ± 0.73	**99.58 ± 0.28**
	0.05	99.33 ± 0.86	**99.58 ± 0.59**	99.25 ± 0.89	99.50 ± 0.58	99.17 ± 0.48	99.17 ± 0.64	99.25 ± 0.88	**99.58 ± 0.28**
	0.10	99.25 ± 0.92	**99.58 ± 0.59**	99.25 ± 0.89	99.50 ± 0.58	99.17 ± 0.48	99.33 ± 0.31	99.33 ± 0.31	99.25 ± 0.31
MF8	0.00	98.75 ± 1.48	99.08 ± 1.07	98.55 ± 1.56	99.17 ± 1.11	99.17 ± 0.16	99.33 ± 0.29	**99.58 ± 0.29**	99.33 ± 0.86
	0.05	98.75 ± 1.58	99.08 ± 1.07	98.55 ± 1.55	99.17 ± 1.11	99.33 ± 0.58	99.17 ± 0.58	**99.58 ± 0.64**	99.17 ± 0.43
	0.10	98.67 ± 1.53	99.08 ± 1.07	98.58 ± 1.43	99.08 ± 1.14	99.17 ± 0.43	99.17 ± 0.43	**99.33 ± 0.64**	99.17 ± 0.43

Table 2 continued

Dataset	z	SVM *Accuracy*	RLS *Accuracy*	ELM *Accuracy*	ALSSVM *Accuracy*	LapSVM *Accuracy*	LapRLS *Accuracy*	SSELM *Accuracy*	asy-LapSVM *Accuracy*
NWO1	0.00	80.83 ± 1.85	80.76 ± 2.15	81.11 ± 1.80	81.02 ± 1.55	81.11 ± 1.80	81.26 ± 1.10	81.35 ± 2.02	**81.55 ± 1.22**
	0.05	80.83 ± 1.87	80.81 ± 2.04	81.09 ± 1.67	81.03 ± 1.24	81.17 ± 1.25	**81.33 ± 1.36**	80.29 ± 2.55	81.21 ± 1.06
	0.10	80.45 ± 1.77	81.01 ± 2.33	80.93 ± 1.73	80.22 ± 1.89	81.04 ± 1.42	**81.17 ± 1.47**	80.79 ± 2.29	81.01 ± 1.33
NWO2	0.00	90.18 ± 1.30	90.57 ± 1.29	90.22 ± 0.97	90.02 ± 1.35	90.18 ± 0.87	**90.65 ± 0.93**	90.42 ± 0.93	90.02 ± 0.93
	0.05	90.22 ± 1.20	**90.57 ± 1.34**	90.17 ± 0.88	90.10 ± 1.31	90.12 ± 0.93	90.53 ± 1.12	90.21 ± 0.89	90.27 ± 1.01
	0.10	90.26 ± 1.09	**90.57 ± 1.50**	90.18 ± 0.99	90.12 ± 1.30	90.15 ± 1.10	90.11 ± 1.33	90.33 ± 1.03	89.22 ± 0.89
NWO3	0.00	81.27 ± 1.08	89.78 ± 2.88	89.54 ± 2.67	81.34 ± 1.17	89.68 ± 2.52	89.78 ± 2.88	**90.44 ± 2.43**	89.22 ± 2.51
	0.05	81.27 ± 1.08	**89.93 ± 2.84**	89.34 ± 2.64	81.27 ± 1.14	89.64 ± 2.76	88.78 ± 2.01	89.15 ± 3.13	89.28 ± 1.45
	0.10	81.27 ± 1.08	89.70 ± 2.79	89.22 ± 2.38	81.12 ± 1.12	89.25 ± 3.14	88.56 ± 3.15	**89.76 ± 2.98**	89.12 ± 1.77
NWO4	0.00	80.56 ± 1.88	81.77 ± 1.79	79.83 ± 1.44	80.77 ± 1.93	81.12 ± 2.13	81.22 ± 1.63	81.77 ± 1.72	**82.52 ± 1.31**
	0.05	80.56 ± 1.91	81.87 ± 1.72	79.85 ± 1.36	80.69 ± 1.92	80.71 ± 1.56	81.35 ± 1.55	81.72 ± 1.76	**82.33 ± 1.99**
	0.10	80.61 ± 1.74	**81.72 ± 1.88**	79.91 ± 1.46	80.69 ± 1.82	80.71 ± 1.56	80.81 ± 1.31	81.65 ± 1.82	81.67 ± 1.29
NWO5	0.00	80.24 ± 2.30	81.81 ± 2.44	80.41 ± 2.28	**83.02 ± 2.57**	82.59 ± 2.57	82.66 ± 2.57	**83.02 ± 2.51**	81.97 ± 2.05
	0.05	**82.64 ± 2.50**	81.89 ± 2.37	80.38 ± 2.27	80.90 ± 2.51	80.55 ± 3.55	81.91 ± 2.51	82.33 ± 2.51	82.55 ± 1.92
	0.10	82.66 ± 2.41	81.52 ± 2.45	80.37 ± 2.26	82.07 ± 2.53	82.11 ± 2.17	82.97 ± 2.67	82.97 ± 2.53	**83.55 ± 2.35**
NWO6	0.00	72.83 ± 1.66	73.59 ± 1.89	72.44 ± 1.21	73.08 ± 1.80	73.85 ± 1.54	73.33 ± 1.54	73.12 ± 1.35	**74.45 ± 1.58**
	0.05	72.83 ± 1.81	73.64 ± 1.82	72.76 ± 1.09	73.06 ± 1.79	**73.85 ± 1.54**	73.40 ± 1.15	73.49 ± 1.22	73.73 ± 1.72
	0.10	72.76 ± 1.66	73.59 ± 1.92	72.61 ± 1.52	72.98 ± 1.75	**74.59 ± 1.36**	73.69 ± 1.76	73.59 ± 1.44	73.59 ± 1.54
NWO7	0.00	82.83 ± 1.72	84.23 ± 1.35	83.40 ± 1.56	83.27 ± 1.75	84.33 ± 1.78	84.36 ± 1.82	84.26 ± 1.78	**85.45 ± 2.64**
	0.05	82.83 ± 1.72	84.21 ± 1.35	83.42 ± 1.28	83.25 ± 1.82	**84.26 ± 1.78**	**84.26 ± 1.82**	84.23 ± 1.58	84.11 ± 1.22
	0.10	82.75 ± 1.82	84.26 ± 1.40	83.54 ± 1.42	83.19 ± 1.75	84.54 ± 2.22	**85.23 ± 2.31**	84.12 ± 1.45	84.26 ± 1.78

Table 3 The comparison of learning time (s) of eight algorithms on the employed datasets

Dataset	z	SVM	RLS	ELM	aLS-SVM	LapSVM	LapRLS	SSELM	asy-LapSVM
G50C	0	0.0303	0.0031	0.1127	0.0167	0.1801	0.0652	0.1966	0.1287
	0.05	0.0305	0.0030	0.0607	0.0093	0.1297	0.0519	0.1265	0.1148
	0.10	0.0295	0.0026	0.0531	0.0082	0.1137	0.0470	0.1365	0.1138
Blood	0	0.0511	0.0027	0.0594	0.0120	0.1851	0.0338	0.1913	0.1572
	0.05	0.0400	0.0023	0.0612	0.0108	0.1836	0.0345	0.1848	0.1578
	0.10	0.0385	0.0024	0.0606	0.0107	0.1809	0.0325	0.1833	0.1629
spect	0	0.0134	0.0010	0.0718	0.0035	0.0576	0.0054	0.0388	0.0324
	0.05	0.0131	0.0010	0.0285	0.0033	0.0390	0.0043	0.0379	0.0332
	0.10	0.0128	0.0011	0.0266	0.0030	0.0382	0.0047	0.0383	0.0334
Ionosphere	0	0.0209	0.0017	0.0336	0.0039	0.0508	0.0104	0.0550	0.0457
	0.05	0.0198	0.0015	0.0344	0.0042	0.0494	0.0100	0.0557	0.0480
	0.10	0.0191	0.0017	0.0332	0.0038	0.0520	0.0106	0.0585	0.0505
fourclss	0	0.0255	0.0020	0.0530	0.0086	0.1398	0.0239	0.1748	0.1555
	0.05	0.0255	0.0019	0.0508	0.0084	0.1239	0.0220	0.1604	0.1461
	0.10	0.0267	0.0019	0.0492	0.0088	0.1273	0.0216	0.1550	0.1367
monks2	0	0.0296	0.0021	0.0509	0.0053	0.1339	0.0205	0.1337	0.1154
	0.05	0.0318	0.0023	0.0508	0.0058	0.1514	0.0204	0.1337	0.1186
	0.10	0.0287	0.0020	0.0497	0.0060	0.1219	0.0210	0.1287	0.1180
Pima	0	0.0361	0.0024	0.0626	0.0155	0.1975	0.0352	0.2045	0.1813
	0.05	0.0323	0.0024	0.0625	0.0118	0.1942	0.0361	0.1997	0.1804
	0.10	0.0392	0.0029	0.0623	0.0140	0.1916	0.0346	0.1960	0.1788
COIL201	0	0.0152	0.0010	0.0261	0.0029	0.0272	0.0024	0.0270	0.0236
	0.05	0.0150	0.0008	0.0171	0.0028	0.0277	0.0026	0.0267	0.0242
	0.10	0.0157	0.0008	0.0121	0.0031	0.0259	0.0025	0.0264	0.0236

Table 3 continued

Dataset	z	SVM	RLS	ELM	aLS-SVM	LapSVM	LapRLS	SSELM	asy-LapSVM
COIL202	0	0.0135	0.0008	0.0129	0.0026	0.0289	0.0023	0.0328	0.0285
	0.05	0.0149	0.0008	0.0131	0.0028	0.0253	0.0025	0.0348	0.0271
	0.10	0.0145	0.0009	0.0126	0.0028	0.0286	0.0026	0.0316	0.0259
COIL203	0	0.0138	0.0010	0.0123	0.0032	0.0252	0.0027	0.0356	0.0104
	0.05	0.0135	0.0009	0.0118	0.0032	0.0277	0.0021	0.0233	0.0109
	0.10	0.0130	0.0009	0.0114	0.0029	0.0283	0.0022	0.0143	0.0109
USPS1	0	0.1507	0.0113	0.1977	0.0766	1.8833	0.3630	1.9650	1.7336
	0.05	0.1636	0.0107	0.1965	0.0891	1.9069	0.3628	1.9590	1.7276
	0.10	0.1583	0.0106	0.1978	0.0782	1.8776	0.3608	1.9483	1.7536
USPS2	0	0.0580	0.0062	0.1205	0.0302	0.6479	0.1316	0.6056	0.5191
	0.05	0.0564	0.0058	0.1236	0.0285	0.6593	0.1278	0.5697	0.5427
	0.10	0.0660	0.0057	0.1277	0.0268	0.6568	0.1315	0.5814	0.5198
YaleB1	0	0.0116	0.0006	0.0108	0.0023	0.0254	0.0020	0.0301	0.0236
	0.05	0.0127	0.0009	0.0083	0.0024	0.0250	0.0019	0.0305	0.0254
	0.10	0.0125	0.0009	0.0081	0.0024	0.0247	0.0017	0.0296	0.0249
YaleB2	0	0.0126	0.0008	0.0093	0.0025	0.0248	0.0020	0.0279	0.0243
	0.05	0.0165	0.0008	0.0081	0.0032	0.0270	0.0023	0.0275	0.0306
	0.10	0.0121	0.0008	0.0088	0.0025	0.0247	0.0019	0.0298	0.0247
YaleB3	0	0.0113	0.0007	0.0083	0.0025	0.0252	0.0017	0.0289	0.0244
	0.05	0.0127	0.0006	0.0083	0.0028	0.0256	0.0023	0.0273	0.0254
	0.10	0.0130	0.0006	0.0081	0.0025	0.0264	0.0021	0.0302	0.0251
MF1	0	0.0226	0.0028	0.0525	0.0075	0.0616	0.0094	0.0748	0.0582
	0.05	0.0207	0.0035	0.0496	0.0056	0.0593	0.0172	0.0716	0.0573
	0.10	0.0214	0.0034	0.0508	0.0064	0.0619	0.0116	0.0671	0.0581

Table 3 continued

Dataset	z	SVM	RLS	ELM	aLS-SVM	LapSVM	LapRLS	SSELM	asy-LapSVM
MF2	0	0.0208	0.0033	0.0504	0.0063	0.0577	0.0077	0.0663	0.0598
	0.05	0.0202	0.0036	0.0503	0.0067	0.0581	0.0076	0.0721	0.0645
	0.10	0.0199	0.0032	0.0520	0.0060	0.0594	0.0079	0.0705	0.0612
MF3	0	0.0206	0.0028	0.0506	0.0062	0.0785	0.0078	0.0671	0.0592
	0.05	0.0212	0.0030	0.0511	0.0056	0.0592	0.0079	0.0673	0.0567
	0.10	0.0215	0.0029	0.0490	0.0065	0.0568	0.0081	0.0752	0.0581
MF4	0	0.0191	0.0027	0.0505	0.0059	0.0580	0.0084	0.0660	0.0604
	0.05	0.0223	0.0030	0.0516	0.0060	0.0597	0.0083	0.0667	0.0579
	0.10	0.0209	0.0028	0.0528	0.0058	0.0570	0.0079	0.0669	0.0569
MF5	0	0.0207	0.0028	0.0527	0.0046	0.0991	0.0085	0.0663	0.0587
	0.05	0.0210	0.0030	0.0518	0.0047	0.0645	0.0080	0.0662	0.0607
	0.10	0.0235	0.0025	0.0512	0.0046	0.0648	0.0078	0.0671	0.0618
MF6	0	0.0187	0.0030	0.0575	0.0058	0.0629	0.0076	0.0637	0.0694
	0.05	0.0185	0.0032	0.0523	0.0057	0.0647	0.0080	0.0648	0.0588
	0.10	0.0181	0.0027	0.0550	0.0066	0.0637	0.0079	0.0663	0.0577
MF7	0	0.0197	0.0027	0.0532	0.0045	0.0634	0.0078	0.0634	0.0567
	0.05	0.0197	0.0027	0.0502	0.0053	0.0651	0.0078	0.0731	0.0578
	0.10	0.0219	0.0031	0.0513	0.0047	0.0605	0.0076	0.0657	0.0596
MF8	0	0.0198	0.0032	0.0521	0.0053	0.0659	0.0079	0.0662	0.0604
	0.05	0.0200	0.0029	0.0510	0.0053	0.0714	0.0078	0.0630	0.0584
	0.10	0.0221	0.0027	0.0511	0.0051	0.0747	0.0078	0.0645	0.0584

Table 3 continued

Dataset	z	SVM	RLS	ELM	aLS-SVM	LapSVM	LapRLS	SSELM	asy-LapSVM
NWO1	0	0.0935	0.0434	0.2111	0.0317	0.7131	0.1812	0.7633	0.6772
	0.05	0.0973	0.0430	0.2180	0.0256	0.7048	0.1886	0.7732	0.7216
	0.10	0.0929	0.0412	0.2153	0.0272	0.6903	0.1781	0.7730	0.7245
NWO2	0	0.1321	0.0615	0.2852	0.0428	1.2372	0.2966	1.3334	1.2108
	0.05	0.1326	0.0630	0.2756	0.0441	1.2647	0.3135	1.3400	1.2288
	0.10	0.1358	0.0645	0.2877	0.0433	1.2266	0.3161	1.3100	1.1748
NWO3	0	0.0269	0.0050	0.0788	0.0053	0.0826	0.0136	0.0839	0.0734
	0.05	0.0224	0.0048	0.0742	0.0055	0.0844	0.0147	0.0794	0.0785
	0.10	0.0243	0.0055	0.0748	0.0056	0.0863	0.0136	0.0760	0.0710
NWO4	0	0.2006	0.0935	0.3790	0.0772	2.4063	1.7028	2.4774	2.3718
	0.05	0.2093	0.0949	0.3844	0.0767	2.3864	1.7100	2.4503	2.3169
	0.10	0.2100	0.0951	0.3905	0.0766	2.4217	1.7078	2.4737	2.3300
NWO5	0	0.1257	0.0612	0.2737	0.0408	1.1691	0.7310	1.1011	1.0914
	0.05	0.1266	0.0607	0.2798	0.0435	1.1606	0.7164	1.1423	1.0771
	0.10	0.1303	0.0585	0.2717	0.0413	1.1368	1.4251	1.1093	1.0991
NWO6	0	0.1657	0.0829	0.3381	0.0533	1.4573	0.4792	1.4267	1.2419
	0.05	0.1624	0.0840	0.3619	0.0521	1.4133	0.4566	1.4908	1.1187
	0.10	0.1650	0.0832	0.3276	0.0558	1.4132	0.4729	1.4917	1.2328
NWO7	0	0.0903	0.0444	0.2208	0.0307	0.7121	0.3968	0.7089	0.6397
	0.05	0.0859	0.0426	0.2220	0.0278	0.6758	0.3097	0.6606	0.6471
	0.10	0.0885	0.0444	0.2237	0.0275	0.6695	0.3662	0.6599	0.6502

4.4 Statistical test

To further compare the performance of the proposed asy-LapSVM with the relevant SSL algorithms, we employ the well-known nonparametric Friedman test with the corresponding post hoc tests (Demsar 2006). The average ranks of four methods on 90 employed datasets in terms of accuracy are shown in Table 4. Under the null hypothesis that all the algorithms are equivalent, we compute the Friedman statistics

$$\mathcal{T}_{\chi^2} = \frac{12 \times 90}{4 \times 5}\left[2.63^2 + 2.79^2 + 2.54^2 + 2.03^2 - \frac{4 \times 5^2}{4}\right] = 14.769 \tag{49}$$

and

$$\mathcal{T}_F = \frac{89 \times \mathcal{T}_{\chi^2}}{90 \times 3 - \mathcal{T}_{\chi^2}} = 5.15 \tag{50}$$

which is distributed according to the $\mathcal{F}$-distribution with (3, 87) degrees of freedom.

For $\alpha = 0.05, \mathcal{F}_\alpha(3, 267) = 2.6384 < 5.15$, so we reject the null hypothesis. Next, the Nemenyi post-hoc test is exploited to further compare the four algorithms in pairs. Based on the Studentized range statistic divided by $\sqrt{2}$, we know $q_\alpha = 2.569$ and the critical difference

$$CD = q_\alpha\sqrt{\frac{4 \times 5}{6 \times 90}} = 0.4944. \tag{51}$$

Thus, if the average ranks of two algorithms differ by at least CD, then their performance is significantly different. From Table 4, we can derive the differences between the proposed asy-LapSVM and other three SSL algorithms as follows:

$$\begin{aligned} \mathrm{d(LapSVM{-}asy\text{-}LapSVM)} &= 2.63 - 2.03 = 0.60 > 0.4944, \\ \mathrm{d(LapRLS{-}asy\text{-}LapSVM)} &= 2.79 - 2.03 = 0.76 > 0.4944, \\ \mathrm{d(SSELM{-}asy\text{-}LapSVM)} &= 2.54 - 2.03 = 0.51 > 0.4944, \end{aligned} \tag{52}$$

where $\mathrm{d(a - b)}$ denotes the differences between two algorithms a and b. Then we can summarize that the proposed asy-LapSVM performs significantly better than the other SSL algorithms on the employed datasets with noise of different variances.

4.5 Effect of the parameter *u*

In this subsection, we study the influence of the parameter u in loss function on the performance of the proposed asy-LapSVM. In the experiments, the value of u is tuned in the range $\{0.55, 0.65, 0.75, 0.83, 0.95, 0.99\}$. By changing it in the given range and setting other parameters to be optimal values obtained by one 5-fold cross validation, we estimate its influence on seven employed datasets with $z = 0$ and the results are displayed in Fig. 3. As shown in Fig. 3, all the curves are almost invariable with

Table 4 Average ranks of the accuracies of four SSL algorithms on the employed datasets

z	Dataset	LapSVM	LapRLS	SSELM	asy-LapSVM	Dataset	LapSVM	LapRLS	SSELM	asy-LapSVM
0	G50C	1	3	4	2	MF1	1.5	4	3	1.5
0.05		1.5	1.5	4	3		1	4	2.5	2.5
0.10		2	4	3	1		1	4	2.5	2.5
0	Blood	2	3.5	3.5	1	MF2	1.5	4	3	1.5
0.05		4	3	2	1		1.5	4	3	1.5
0.10		1	3	2	4		1.5	4	.3	1.5
0	Spect	4	3	1.5	1.5	MF3	1.5	4	3	1.5
0.05		4	3	2	1		1	4	2.5	2.5
0.10		4	1	2	3		1	4	2.5	2.5
0	Ionosphere	2	1	4	3	MF4	2	4	2	2
0.05		4	2	3	1		1	2.5	4	2.5
0.10		3	1	4	2		1	2.5	4	2.5
0	Fourclss	2	3	4	1	MF5	4	2.5	2.5	1
0.05		2.5	2.5	4	1		2.5	1	2.5	4
0.10		3	2	4	1		1	3.5	2	3.5
0	Monks2	4	3	2	1	MF6	2	4	2	2
0.05		3	3	3	1		1.5	3	1.5	4
0.10		4	3	1	2		1	3	4	2
0	Pima	3	2	4	1	MF7	3	3	3	1
0.05		1	2	3.5	3.5		3.5	3.5	2	1
0.10		3	3	3	1		4	1.5	1.5	3
0	Coil-201	3	4	2	1	MF8	4	2.5	1	2.5
0.05		2	4	1	3		2	3.5	1	3.5
0.10		2	1	4	3		3	3	1	3

Table 4 continued

z	Dataset	LapSVM	LapRLS	SSELM	asy-LapSVM	Dataset	LapSVM	LapRLS	SSELM	asy-LapSVM
0	Coil-202	1.5	4	1.5	3	NWO1	4	3	2	1
0.05		4	2	1	3		3	1	4	2
0.10		2	3	1	4		2	1	4	3
0	Coil-203	2	1	4	3	NWO2	3	1	2	4
0.05		4	3	1	2		4	1	3	2
0.10		1	3	4	2		2	1	4	3
0	USPS1	4	3	2	1	NWO3	3	2	1	4
0.05		4	2	3	1		1	4	3	2
0.10		4	3	2	1		2	4	1	3
0	USPS2	4	2.5	1	2.5	NWO4	4	3	2	1
0.05		3	4	1	2		4	3	2	1
0.10		4	3	1.5	1.5		4	3	2	1
0	YaleB1	4	3	2	1	NWO5	3	2	1	4
0.05		3.5	3.5	2	1		4	3	2	1
0.10		3.5	3.5	2	1		4	2.5	2.5	1
0	YaleB2	2	3.5	3.5	1	NWO6	2	3	4	1
0.05		3	1.5	4	1.5		1	4	3	2
0.10		4	3	2	1		1	2	3.5	3.5
0	YaleB3	4	2	3	1	NWO7	3	2	4	1
0.05		4	2	3	1		1.5	1.5	3	4
0.10		2	4	3	1		2	1	4	3
	Average		2.63		2.79		2.54		2.03	

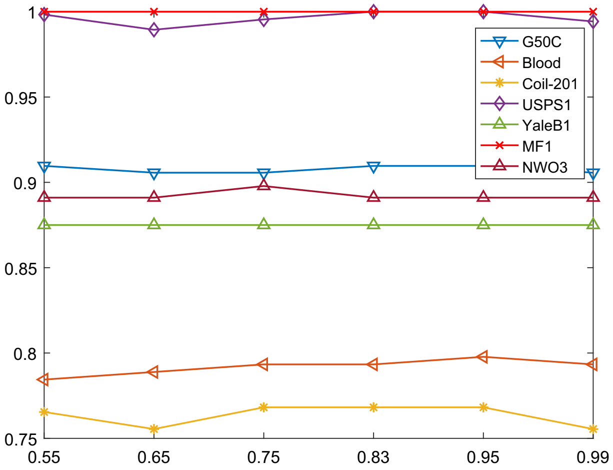

Fig. 3 The classification accuracies of asy-LapSVM with respect to u on seven employed datasets with $z = 0$

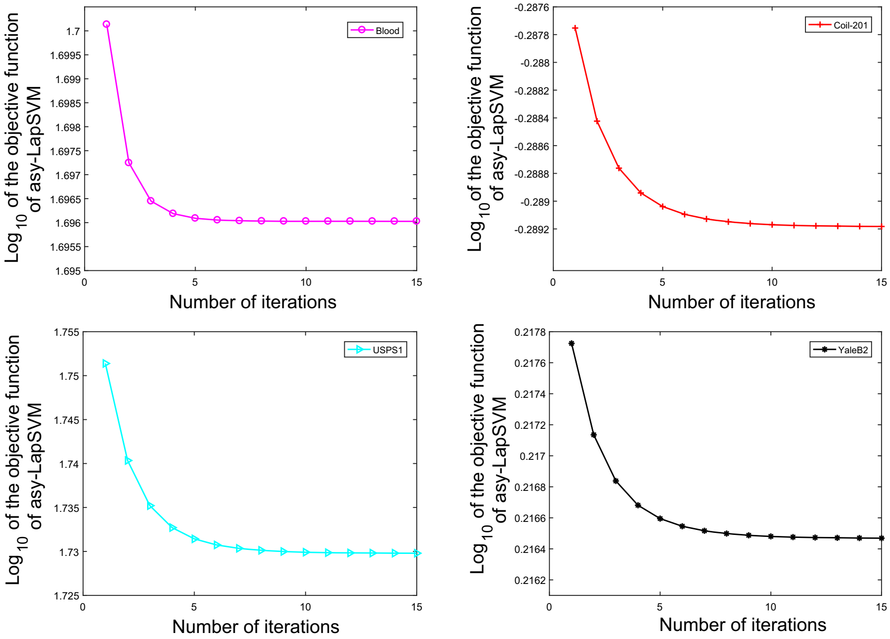

Fig. 4 Analysis for the convergence of asy-LapSVM on four datasets with $z = 0$

the different values of u, which means that the value of u has little effect on the performance of the proposed algorithm. In other words, the proposed asy-LapSVM is insensitive to the parameter u.

4.6 Analysis for the convergence

In this subsection, by exploiting an empirical justification (Ye 2005), we discuss the convergence of the proposed asy-LapSVM on four employed datasets with $z = 0$. The learning results are illustrated in Fig. 4, where the abscissa denotes the number of iterations and the ordinate denotes the logarithm values of the objective function (35). From Fig. 4, it can be clearly seen that the logarithm values of the objective function change with the iterations and go fast steady in less than 10 iterations. Therefore, we can conclude that the proposed algorithm converges in limited iterations.

5 Conclusion

A novel asy-LapSVM is proposed in this paper to enhance the generalization performance of the classical LapSVM. To our knowledge, it is almost the first time to employ the asymmetric squared loss to improve the performance of the SSL algorithms. Moreover, we present a simple and efficient functional iterative method to solve the proposed asy-LapSVM. And we further investigate the convergence of the functional iterative method from both theoretical and experimental aspects. The validity of the proposed asy-LapSVM and the feasibility of the presented functional iterative method are demonstrated by numerical experiments on a series of popular datasets with noise of different variances.

However, limitations of the proposed asy-LapSVM still exist, such as it is not suitable for online learning. The research along this line will be our future work.

Acknowledgements The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

Balasundaram S, Benipal G (2016) On a new approach for Lagrangian support vector regression. Neural Comput Appl. 29(9):533–551

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

Bi J, Zhang T (2004) Support vector classification with input data uncertainty. Neural Inf Process Syst (NIPS) 17:161–168

Calma A, Reitmaier T, Sick B (2018) Semi-supervised active learning for support vector machines: a novel approach that exploits structure information in data. Inform Sci 456:13–33

Chapelle O, Scholkopf B, Zien A (2006) Semi-supervised learning. MIT Press, Cambridge

Chen W, Shao Y, Xu D, Fu Y (2014) Manifold proximal support vector machine for semi-supervised classification. Appl Intell 40:623–638

Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Du B, Tang X, Wang Z, Zhang L, Tao D (2019) Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion. IEEE Trans Cybern 49(4):1440–1453

Fung G, Mangasarian OL (2004) A feature selection Newton method for support vector machine classification. Comput Optim Appl 28(2):185–202

Gu N, Fan P, Fan M, Wang D (2019) Structure regularized self-paced learning for robust semi-supervised pattern classification. Neural Comput Appl 31(10):6559–6574

Guzella TS, Caminhas WM (2009) A review of machine learning approaches to spam filtering. Expert Syst Appl 36(7):10206–10222

Huang G, Zhu Q, Siew C (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501 (Neural Networks Selected Papers from the 7th Brazilian Symposium on Neural Networks, SBRN'04)

Huang X, Shi L, Suykens JAK (2014) Support vector machine classifier with pinball loss. IEEE Trans Pattern Anal Mach Intell 36(5):984–997

Huang X, Shi L, Suykens JAK (2014) Asymmetric least squares support vector machine classifiers. Comput Stat Data Anal 70:395–405

Huang G, Song S, Gupta J, Wu C (2014) Semi-supervised and unsupervised extreme learning machines. IEEE Trans Cybern 44:2405–2417

Jumutc V, Huang X, Suykens JAK (2013) Fixed-size Pegasos for hinge and pinball loss SVM. In: Proceedings of the international joint conference on neural network, Dallas, TX, USA. pp 1122–1128

Khemchandani R, Pal A (2016) Multi-category Laplacian least squares twin support vector machine. Appl Intell 45:458–474

Koenker R (2005) Quantile regression. Cambridge University Press, Cambridge

Li Z, Tian Y, Li K, Zhou F, Yang W (2017) Reject inference in credit scoring using semi-supervised support vector machines. Expert Syst Appl 74:105–114

Lu L, Lin Q, Pei H, Zhong P (2018) The aLS-SVM based multi-task learning classifiers. Appl Intell 48:2393–2407

Ma J, Wen Y, Yang L (2019) Lagrangian supervised and semi-supervised extreme learning machine. Appl Intell 49(2):303–318

Melki G, Kecman V, Ventura S, Cano A (2018) OLLAWV: online learning algorithm using worst-violators. Appl Soft Comput 66:384–393

Pei H, Chen Y, Wu Y, Zhong P (2017) Laplacian total margin support vector machine based on within-class scatter. Knowl-Based Syst 119:152–165

Pei H, Wang K, Zhong P (2017) Semi-supervised matrixized least squares support vector machine. Appl Soft Comput 61:72–87

Pei H, Wang K, Lin Q, Zhong P (2018) Robust semi-supervised extreme learning machine. Knowl-Based Syst 159:203–220

Scardapane S, Fierimonte R, Lorenzo PD, Panella M, Uncini A (2016) Distributed semi-supervised support vector machines. Neural Netw. 80:43–52

Shivaswamy P, Bhattacharyya C, Smola A (2006) Second order cone programming approaches for handling missing and uncertain data. J Mach Learn Res 7:1283–1314

Steinwart I, Christmann A (2008) Support vector machines. Springer, New York

Sun S (2013) Multi-view Laplacian support vector machines. Appl Intell 41(4):209–222

Tikhonov AN (1963) Regularization of incorrectly posed problems. Sov. Math. Dokl 4:1624–1627

Tur G, Hakkani-Tür D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. Speech Commun 45(2):171–186

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Wang K, Zhong P (2014) Robust non-convex least squares loss function for regression with outliers. Knowl-Based Syst 71:290–302

Wang K, Zhu W, Zhong P (2015) Robust support vector regression with generalized loss function and applications. Neural Process Lett 41:89–106

Xu H, Caramanis C, Mannor S (2009) Robustness and regularization of support vector machines. J Mach Learn Res 10:1485–1510

Ye J (2005) Generalized low rank approximations of matrices. Mach Learn 61(1–3):167–191

Zhang T, Liu S, Xu C, Lu H (2011) Boosted multi-class semi-supervised learning for human action recognition. Pattern Recognit 44(10–11):2334–2342

Zhao J, Xu Y, Fujita H (2019) An improved non-parallel Universum support vector machine and its safe sample screening rule. Knowl-Based Syst 170:79–88

Zhong P (2012) Training robust support vector regression with smooth non-convex loss function. Optim Methods Softw 27(6):1039–1058