Applied Intelligence (2019) 49:303–318
https://doi.org/10.1007/s10489-018-1273-4

Lagrangian supervised and semi-supervised extreme learning machine

Jun Ma[1] · Yakun Wen[1] · Liming Yang[2]

Published online: 25 August 2018

Abstract
Two extreme learning machine (ELM) frameworks are proposed to handle supervised and semi-supervised classifications. The first is called lagrangian extreme learning machine (LELM), which is based on optimality conditions and dual theory. Then LELM is extended to semi-supervised setting to obtain a semi-supervised extreme learning machine (called Lap-LELM), which incorporates the manifold regularization into LELM to improve performance when insufficient training information is available. In order to avoid the inconvenience caused by matrix inversion, Sherman-Morrison-Woodbury (SMW) identity is used in LELM and Lap-LELM, which leads to two smaller sized unconstrained minimization problems. The proposed models are solvable in a space of dimensionality equal to the number of sample points. The resulting iteration algorithms converge globally and have low computational burden. So as to verify the feasibility and effectiveness of the proposed method, we perform a series of experiments on a synthetic dataset, near-infrared (NIR) spectroscopy datasets and benchmark datasets. Compared with the traditional methods, experimental results demonstrate that the proposed methods achieve better performances than the traditional supervised and semi-supervised methods in most of considered datasets.

Keywords Optimality conditions · Lagrangian function · Extreme learning machine · Semi-supervised learning · Classification

1 Introduction

Due to its simple structure, low computational complexity and good generalization, extreme learning machine (ELM) [1–6] has been successfully applied in many fields [7–10]. Compared with traditional neural networks, the main advantages of ELM are that it runs fast with global optimal solution and is easy to implement. Its hidden nodes and input weights are randomly generated and the output weights are expressed analytically. Recently, many researchers have done in-depth researches on ELM. For example, Wang et al. [7] proposed a self-adaptive extreme learning machine (SaELM). It can select the best neuron number in hidden layers to construct the optimal networks. This method trains fast and can obtain the global optimal solution, with good generalization. Zhang et al. [8] proposed a ELM$_+$ which introduces the privileged information to the traditional ELM method. Zhang et al. [9] proposed a memetic algorithm based extreme learning machine (M-ELM). It embeds the local search strategy into the global optimization framework to obtain optimal network parameters. Huang et al. [2] proposed a ELM based on optimization theory (OPT-ELM) by introducing hinge loss into the ELM framework. It minimizes the norm of the output weights to find a separating hyperplane with the maximal margin between two classes of data, which is similar to the idea employed in support vector machine (SVM) [11]. Compared to ELM, the minimization norm of output weights enables OPT-ELM to get better generalization performance. OPT-ELM solves a quadratic programming (QP) problem, which assures that a global optimal solution can be found.

The manifold regularization method has been widely used for semi-supervised learning tasks. One of the most popular manifold regularization is the Laplacian regularization [12–18], which utilizes graph Laplacian to determine the geometry information of data. ELMs are very popular in many fields and are mainly used to supervised learning

✉ Liming Yang
cauyanglm@163.com

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

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

tasks, which greatly limits their applicability. Many researchers introduced Laplacian regularization into the ELM framework for semi-supervised learning tasks [19–23].

Lagrangian support vector machine (LSVM) [24] is a computationally powerful machine learning tool. It minimizes an unconstrained differentiable convex function in a space of dimensionality equal to the number of classified points. In recent years, some researchers have extended the idea of LSVM to twin support vector machines [25–33].

In this paper, inspired by the studies above, we propose two novel ELM formulations for supervised and semi-supervised classifications. The main contributions of this paper are summarized as follows:

(1) The first is called lagrangian extreme learning machine (LELM), which is based on optimality conditions and dual theory. Then LELM is generalized to semi-supervised setting to obtain a semi-supervised lagrangian extreme learning machine (Lap-LELM), which incorporates the manifold regularization into LELM to improve performance when insufficient training information is available.

(2) In order to avoid the inconvenience caused by matrix inversion, Sherman-Morrison-Woodbury (SMW) [24] identity is used in LELM and Lap-LELM optimization problems, which leads to two smaller sized unconstrained minimization problems. The proposed models are solvable in a space of dimensionality equal to the number of samples.

(3) Two fast and simple algorithms are designed to optimize the proposed LELM and Lap-LELM, which requires only iteratively solving equations rather than quadratic programming like OPT-ELM. The resulting iteration algorithms converge globally and have low computational burden.

(4) Difference from the lagrangian SVM (LSVM) which has difficulty in dealing with nonlinear problems, the proposed LELM and Lap-LELM have the explicit kernel function form, and are convenient to use in nonlinear classifications.

The rest of this paper is organized as follows. Section 2, briefly dwells on the OPT-ELM, LSVM and MR. We propose LELM and Lap-LELM in Sections 3 and 4, respectively. The experimental results, and discussions are presented in Sections 5. Finally, the conclusion is drawn in Section 6.

2 Related work

In order to propose an improved version of OPT-ELM, we review OPT-ELM in Section 2.1, and introduce LSVM and MR in Sections 2.2 and 2.3, respectively.

2.1 Optimization method based ELM

Consider a supervised learning problem with training data $T_l = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l}, i = 1, \ldots, l$, where $x_i \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$, T_l denotes a set of l labeled samples.

Huang et al. [2] proposed ELM framework based on optimization theory (called OPT-ELM), in which the hinge loss function was introduced. This leads to the following optimization:

$$\begin{array}{ll} \min\limits_{\beta,\xi} & \frac{1}{2} \parallel \beta \parallel^2 + C \sum_{i=1}^{l} \xi_i \\ \text{s.t.} & y_i h(x_i)^T \beta \geq 1 - \xi_i,\ \xi_i \geq 0,\ \ i = 1, \ldots, l \end{array} \tag{1}$$

where $h(x) = (g(w_1^T x + b_1), \ldots, g(w_L^T x + b_L))^T$ actually maps the data from the n-dimensional input space to ELM feature space; L is the number of hidden layer nodes; $\xi = (\xi_1, \ldots, \xi_l)$ is a slack variable; C is a positive penalty parameter. This is a quadratic programming with global solution.

According to the optimization theory, the dual problem of optimization problem (1) is

$$\begin{array}{ll} \min\limits_{\alpha} & \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j h(\mathbf{x}_i) h(\mathbf{x}_j) - \sum_{i=1}^{l} \alpha_i \\ \text{s.t.} & 0 \leq \alpha_i \leq C,\ \ i = 1, ..., l. \end{array} \tag{2}$$

We can define the ELM kernel function as:

$$\begin{aligned} K_{ELM} &= h(x_i)h(x_j) \\ &= [G(a_1, b_1, x_i), \ldots, G(a_L, b_L, x_i)]^T \cdot [G(a_1, b_1, x_j), \ldots, \\ &\quad G(a_L, b_L, x_j)]^T \end{aligned} \tag{3}$$

where $G(a, b, x)$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [1–6] and $\{(a_i, b_i)\}_{i=1}^{L}$ are randomly generated according to any continuous probability distribution.

Thus, we can get the following optimization problem:

$$\begin{array}{ll} \min\limits_{\alpha} & \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j K_{ELM}(\mathbf{x}_i, \mathbf{x}_j) \alpha_i \alpha_j - \sum_{i=1}^{l} \alpha_i \\ \text{s.t.} & 0 \leq \alpha_i \leq C,\ \ i = 1, ..., l. \end{array} \tag{4}$$

The decision function of OPT-ELM is

$$f(x) = sign \left(\sum_{i=1}^{l} \alpha_i y_i K_{ELM}(x, x_i) \right) \tag{5}$$

2.2 Lagrangian support vector machine

Mangasarian and Musicant modified the standard SVM and proposed Lagrangian Support Vector Machine [24]:

$$\begin{array}{ll} \min\limits_{\mathbf{w},\gamma,y} & \nu \frac{\xi^T \xi}{2} + \frac{1}{2}(\mathbf{w}^T \mathbf{w} + b^2) \\ \text{s.t.} & D(A\mathbf{w} - eb) + \xi \geq e \end{array} \tag{6}$$

where parameter $\nu > 0$, $\mathbf{w}$ is the normal to the bounding planes and b determines their location relative to the origin. According to duality theory, the dual of this problem is:

$$\min_{0\leq u\in\mathbb{R}^n} f(u) = \frac{1}{2}u^T Qu - e^T u. \tag{7}$$

where $Q = \left(\frac{I}{\nu} + D(AA^T + ee^T)D\right)$.

The LSVM Algorithm is based directly on the Karush-Kuhn-Tucker necessary and sufficient optimality conditions of the dual problem (7):

$$0 \leq u \perp Qu - e \geq 0$$

These optimality conditions lead to the following very simple iterative scheme which constitutes LSVM Algorithm:

$$u^{i+1} = Q^{-1}(e + ((Qu^i - e) - \alpha u^i)_+);\ i = 0, 1, \ldots,$$

for which we will establish global linear convergence from any starting point under the easily satisfiable condition:

$$0 < \alpha < \frac{2}{\nu}$$

More details can refer to [24].

2.3 Manifold regularization

Consider a semi-supervised learning problem with training data $T = T_l \cup T_u = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l} \cup \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, $i = 1, \ldots, l$, where $x_i \in \mathbb{R}^n$, $x_j \in \mathbb{R}^n$, $y_i \in \{-1, +1\}$, T_l denotes a set of l labeled samples, T_u denotes a set of u unlabeled samples. The manifold regularization approach [13] takes advantage of the geometry of the marginal distribution $\mathcal{P}_X$. We assume that the support of data probability distribution has the geometry of the Riemannian manifold $\mathcal{M}$. The label of the two closest samples in the $\mathcal{P}_X$ intrinsic geometry should be the same or similar, which means that the conditional probability distribution $\mathcal{P}(y \mid x)$ should vary little between two such points. As a result, we have

$$\| f \|_I^2 = \sum_{i=1}^{l+u}\sum_{j=1}^{l+u} W_{ij}(f(x_i) - f(x_j))^2 = f^T \mathbf{L}_p f \tag{8}$$

where $\mathbf{L}_p = \mathbf{D} - \mathbf{W}$ is the graph Laplacian; $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ given by $\mathbf{D}_{ii} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$, and $\mathbf{D}_{ij} = 0$ for $i \neq j$; the normalizing coefficient $\frac{1}{(l+u)^2}$ is the natural scale factor for the empirical estimate of the Laplace operator. If kernel function $k(\cdot, \cdot)$ is given, we estimate the target function by minimizing

$$f^* = \arg \min_{f\in\mathcal{H}_k} = \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 + \gamma_I \| f \|_I^2 \tag{9}$$

where V is loss function, γ_A controls the complexity of functions in the ambient space and γ_I controls the complexity of functions in the intrinsic geometry of sample probability distribution.

3 Lagrangian extreme learning machine

In this section, based on the optimization method theory, we propose a new lagrangian extreme learning machine (LELM). Then, we compare LELM with other related algorithms.

3.1 Lagrangian extreme learning machine

We change slightly the OPT-ELM with hinge loss-function. First, we change the l_1-norm of ξ to l_2-norm squared which makes the constraint $\xi \geq 0$ redundant and guarantees the strictly convexity of the object function. Therefore, based on the optimization theory we can get the following optimization problem:

$$\begin{array}{ll} \min\limits_{\beta,\xi} & \frac{1}{2} \| \beta \|^2 + \frac{C}{2} \| \xi \|^2 \\ \text{s.t.} & D(H\beta) + \xi \geq e \end{array} \tag{10}$$

where C is a tradeoff parameter, $D_{l\times l}$ is diagonal matrix with $y_i (i = 1, 2, \ldots, l)$ along its diagonal, $H = [h(x_1), \ldots, h(x_1)]^T$ is output matrix, e is a column vector of any dimension. The object function above is strictly convex, which guarantees that (10) has a unique solution.

The Lagrange function of the primal LELM optimization (10) is

$$L(\beta, \xi, \alpha) = \frac{1}{2} \| \beta \|^2 + \frac{C}{2} \| \xi \|^2 - \alpha^T (DH\beta + \xi - e) \tag{11}$$

where $\alpha = (\alpha_1, \ldots, \alpha_l)^T$ are the Lagrange multipliers with non-negative values.

Based on Karush-Kuhn-Tucker (KKT) condition we have:

$$\frac{\partial L}{\partial \beta} = \beta - H^T D\alpha = 0 \Rightarrow \beta = H^T D\alpha \tag{12}$$

$$\frac{\partial L}{\partial \xi} = C\xi - \alpha = 0 \Rightarrow \xi = \frac{\alpha}{C} \tag{13}$$

Substituting (12) and (13) into (11) we can get the dual form of LELM:

$$\min_{\alpha\geq 0} \frac{1}{2}\alpha^T \left(\frac{I}{C} + DHH^T D\right)\alpha - e^T\alpha \tag{14}$$

Before stating our algorithm we define two matrices to simplify notation as follows:

$$M = [DH, \mathbf{0}], \quad Q = \frac{I}{C} + MM^T. \tag{15}$$

With these definitions, the dual problem (14) can be written as follows:

$$\min_{\alpha\geq 0} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{16}$$

Similarly, the Lagrange function of the optimization problem (16) is

$$L(\alpha,\gamma) = \frac{1}{2}\alpha^T Q\alpha - e^T\alpha - \gamma\alpha$$

Based on the Karush-Kuhn-Tucker necessary and sufficient optimality conditions we can get

$$\frac{\partial L(\alpha,\gamma)}{\partial\alpha} = Q\alpha - e - \gamma = 0 \Rightarrow \gamma = Q\alpha - e \geq 0,$$

$$\gamma\alpha = 0,$$

$$\alpha \geq 0.$$

Thus, we have

$$0 \leq \alpha \perp Q\alpha - e \geq 0. \tag{17}$$

For any two real numbers or vectors x and y, the following identity can be established

$$0 \leq x \perp y \geq 0 \Leftrightarrow x = (x - ay)_+,\ a > 0.$$

Thus, the optimality condition (17) can be written in the following equivalent form for any positive θ:

$$Q\alpha - e = ((Q\alpha - e) - \theta\alpha)_+ \tag{18}$$

These optimality conditions lead to the following very simple iterative scheme which constitutes our LELM Algorithm:

$$\alpha^{i+1} = Q^{-1}[e + ((Q\alpha^i - e) - \theta\alpha^i)_+],\quad i = 0, 1, ..., \tag{19}$$

where θ satisfies the condition

$$0 < \theta < \frac{2}{C}. \tag{20}$$

Next, we will introduce the projection theorem [34] to prepare for the subsequent proof of the global convergence of our algorithm.

Theorem 1 (**Projection Theorem** [34]) *Let X be a nonempty, close, and convex subset of $\mathbb{R}^n$.*

(a) *For every $z \in \mathbb{R}^n$, there exist a unique $x^* \in X$ that minimizes $\| z - x \|$ over all $x \in X$. This vector is called the projection of z on X and denoted by $[z]_+$.*

(b) *Give some $z \in \mathbb{R}^n$, a vector $x^* \in X$ is equal to projection $[z]_+$ if and only if*

$$(z - x^*)^T(x - x^*) \leq 0 \tag{21}$$

(c) *The mapping $f : \mathbb{R}^n \mapsto X$ defined by $f(x) = [x]_+$ is continuous and nonexpansive, that is*

$$\| [x]_+ - [y]_+ \| \leq \| x - y \|,\quad \forall x, y \in \mathbb{R}^n. \tag{22}$$

(d) *In the case where X is a subset, a vector $x^* \in \mathbb{R}^n$ is equal to the projection $[z]_+$ if and only if $z - x^*$ is orthogonal to X, that is,*

$$(z - x^*)^T x = 0. \tag{23}$$

Theorem 2 (**Global Convergence of LELM**) *Let Q be the symmetric positive definite matrix defined by (15) and let (19) hold. Starting with an arbitrary α^0, the iterates α^i of (20) converge to the unique solution $\bar{\alpha}$ of (16) at the linear rate:*

$$\| Q\alpha^{i+1} - Q\bar{\alpha} \| \leq \| I - \theta Q^{-1} \| \cdot \| Q\alpha^i - Q\bar{\alpha} \|. \tag{24}$$

Proof Suppose $\bar{\alpha}$ is the solution of (16), it must satisfy the optimality condition (18) for any $\theta > 0$. So, we have

$$Q\alpha^{i+1} - e = ((Q\alpha^i - e) - \theta\alpha)_+ \tag{25}$$

and

$$Q\bar{\alpha} - e = ((Q\bar{\alpha} - e) - \theta\bar{\alpha})_+, \tag{26}$$

From (25) and (26) we can get:

$$\| Q\alpha^{i+1} - Q\bar{\alpha} \| = \| (Q\alpha^i - \theta\alpha^i)_+ - (Q\bar{\alpha} - e - \theta\bar{\alpha})_+ \|. \tag{27}$$

The Projection Theorem 1 states that the distance between any two points in $\mathbb{R}^n$ is not less than the distance between their projections on any convex set in $\mathbb{R}^n$. We can obtain the following inequality:

$$\begin{aligned} \| Q\alpha^{i+1} - Q\bar{\alpha} \| &\leq \| (Q - \theta I)(\alpha^i - \bar{\alpha}) \| \\ &\leq \| I - \theta Q^{-1} \| \cdot \| Q(\alpha^i - \bar{\alpha}) \| \end{aligned} \tag{28}$$

Now we only need to prove $\| I - \theta Q^{-1} \| < 1$. This follows (20) as follows. Noting the definition (15) of Q and letting $\lambda_i, i = 1, ..., m$, denote the nonnegative eigenvalues of MM^T, all we need is:

$$0 < \theta(\frac{1}{C} + \lambda_i)^{-1} < 2, \tag{29}$$

which is satisfied under the assumption (20). □

Based on the above discussion, the LELM is summarized as Algorithm 1.

```
Algorithm 1 LELM Algorithm
Input:
  Training set $T_l = \{x_i, y_i\}_{i=1}^{l}$

  Step 1: Set parameter $C$.
  Step 2: Randomly assign input weight $w_i$ biases $b_i$.
  Step 3: Calculate the hidden layer output matrix , $H$ .
  Step 4: Via (19) calculate $\alpha$
  Step 5: Calculate the output weight

  $\beta = H^T D\alpha.$

  return The decision function $f(x) = sign(\beta \cdot h(x))$.
```

3.2 Compared with other related algorithms

We compared the proposed Lagrangian extreme learning machine (LELM) with other related algorithms: Lagrangian support vector machine (LSVM) [24], Unconstrained Lagrangian twin support vector machine (ULTWSVM) [28], Optimization method based extreme learning machine (OPT-ELM) [2] and L_2-Regularized Extreme Learning Machine (ELM) [1].

Compared with LSVM and ULTWSVM:

(1) Obviously, the objective is different. The bias b is not required in LELM since the separating hyperplane $\beta^T h(x) = 0$ passes through the origin in LELM feature space, while LSVM needs bias b to determine the hyperplane. The goal of ULTWSVM is to look for two non-parallel classification hyperplanes, but the goal of our proposed LELM is to look for classification hyperplanes that cross the origin.

(2) Difference from the LSVM and ULTWSVM which are difficult to optimize in dealing with nonlinear problems because of the unknown implication mapping and the kernel parameters, kernel function of proposed LELM (10) has the explicit form: $\boldsymbol{K}_{ELM}(\boldsymbol{x}_i, \boldsymbol{x}_j) = h(\boldsymbol{x}_i)^T h(\boldsymbol{x}_j)$, and its network parameters are randomly generated without tuning.

Compared with OPT-ELM and ELM:

(1) Compared with the OPT-ELM, we change slightly the OPT-ELM with hinge loss-function. First, we replace the l_1-norm with the l_2-norm of the slack variable ξ by weighing $\frac{C}{2}$, which guarantees the strict convexity of the object function. This leads to the LELM optimization problem with unique solution.

(2) The traditional ELM tends to reach zero training errors, however, in LELM training errors are generally not equal to zero, which leads to more better generation performance on testing data.

4 Laplacian lagrangian extreme learning machine

In this section, we propose a semi-supervised lagrangian extreme learning machine (Lap-LELM) via extending LELM to a semi-supervised learning framework. The proposed Lap-LELM incorporates the manifold regularization to leverage unlabeled data to improve the classification accuracy when labeled data are scarce. Then, we compare Lap-LELM with other related algorithms.

4.1 Laplacian lagrangian extreme learning machine

For the binary classification applications, the decision function of ELM is $f(\mathbf{x}) = sign(\beta \cdot h(\mathbf{x}))$, where $h(\mathbf{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden layer ELM feature space. By means of the Representer Theorem [2], output weights β can be expressed in the dual problem as the expansion over labeled and unlabeled samples

$$\beta = \sum_{i=1}^{l+u} \alpha_i h(x_i),$$

where $h(x) = [h(x_1), \ldots, h(x_{l+u})]^T$ and $\alpha = [a_1, \ldots, \alpha_{l+u}]$. Then, the decision function is

$$f(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x) \quad (30)$$

and $\mathbf{K}$ is the kernel matrix formed by kernel functions $K(x_i, x_j) = h(x_i) \cdot h(x_j)$. Therefore, the regularization term can be expressed by the kernel matrix and the expansion coefficient:

$$\| f \|_{\mathcal{H}}^2 = \| \beta \|^2 = (h(x)\alpha)^T (h(x)\alpha) = \alpha^T \mathbf{K}\alpha. \quad (31)$$

The geometry of the data is represented by a graph, where nodes represent labeled and unlabeled samples connected by weights W_{ij}. Regularizing the graph follows from the manifold assumption. We can get the manifold term via spectral graph theory [13],

$$\| f \|_I^2 = \frac{1}{(l+u)^2} \sum_{i,j=1}^{l+u} \mathbf{W}_{ij} (f(x_i) - f(x_j))^2 = \frac{1}{(l+u)^2} f^T \mathbf{L}_p f, \quad (32)$$

where $\mathbf{L}_p = \mathbf{D} - \mathbf{W}$ is the graph Laplacian; $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{W}$ given by $\mathbf{D}_{ii} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$, and $\mathbf{D}_{ij} = 0$ for $i \neq j$; the normalizing coefficient $\frac{1}{(l+u)^2}$ is the natural scale factor for the empirical estimate of the Laplace operator; and $\mathbf{f} = \mathbf{K}\alpha$.

Therefore, based on the above (9), (31) and (32) we propose the following Lap-LELM:

$$\begin{array}{ll} \min\limits_{\xi\in\mathbb{R}^l,\alpha\in\mathbb{R}^{l+u}} & \frac{1}{l}\sum_{i=1}^{l}\xi_i + \gamma_A\alpha^T\mathbf{K}\alpha + \frac{\gamma_I}{(l+u)^2}\alpha^T\mathbf{K}^T\mathbf{L}_p\mathbf{K}\alpha \\ \text{s.t.} & y_i\sum_{j=1}^{l+u}\alpha_i\mathbf{K}(x_i,x_j) \geq 1-\xi_i,\ \ i=1,\ldots,l. \\ & \xi_i > 0,\ \ i=1,\ldots,l. \end{array} \tag{33}$$

where γ_A and γ_I is regularization parameters, $\mathbf{K}$ is the kernel matrix formed by kernel functions $K(x_i, x_j) = h(x_i) \cdot h(x_j)$.

The Lagrange function of the primal Lap-LELM optimization (33) is

$$\begin{aligned} L(\alpha,\xi,\lambda) = &\ \frac{1}{l}\sum_{i=1}^{l}\xi_i + \gamma_A\alpha^T\mathbf{K}\alpha + \frac{\gamma_I}{(l+u)^2}\alpha^T\mathbf{K}^T\mathbf{L}_p\mathbf{K}\alpha \\ & -\lambda_i\left(y_i\sum_{i=1}^{l+u}\alpha_i\mathbf{K}(x_i,x_j\right) - 1 + \xi_i) - \theta_i\sum_{i=1}^{l}\xi_i \end{aligned} \tag{34}$$

where $\lambda = [\lambda_1, \ldots, \lambda_l]^T$ are the Lagrange multipliers. In order to find the optimal solutions of (34) we should have

$$\frac{\partial L}{\partial \xi_i} = \frac{1}{l} - \lambda_i - \theta_i = 0 \Rightarrow \theta_i = \frac{1}{l} - \lambda_i \tag{35}$$

Substitute (35) into (34) we have

$$\min_{\alpha,\lambda} \frac{1}{2}\alpha^T(2\gamma_A\mathbf{K}+\frac{2\gamma_I}{(l+u)^2}\mathbf{K}^T\mathbf{L}_p\mathbf{K})\alpha - \alpha^T\mathbf{K}\mathbf{J}^T\mathbf{Y}\lambda + \sum_{i=1}^{l}\lambda_i \tag{36}$$

where $\mathbf{J} = [\mathbf{I}, \mathbf{0}]_{l\times(l+u)}$, $\mathbf{I}_{l\times l}$, $\mathbf{0}_{l\times u}$, $\mathbf{Y} = diag(y_i, \ldots, y_l)$. Taking derivatives again with respect to α, we obtain the solution

$$\alpha = (2\gamma_A\mathbf{I} + \frac{2\gamma_I}{(l+u)^2}\mathbf{L}_p\mathbf{K})^{-1}\mathbf{J}^T\mathbf{Y}\lambda \tag{37}$$

Thusly, we obtain the following quadratic-programming problem:

$$\begin{array}{ll} \min\limits_{\lambda} & \frac{1}{2}\lambda^T\mathbf{S}\lambda - \sum_{i=1}^{l}\lambda_i \\ \text{s.t.} & 0 \leq \lambda_i \leq \frac{1}{l} \end{array} \tag{38}$$

where $\mathbf{S} = \mathbf{YJK}(2\gamma_A\mathbf{I} + \frac{2\gamma_I}{(l+u)^2}\mathbf{L}_p\mathbf{K})^{-1}\mathbf{J}^T\mathbf{Y}$. The Lagrange function of the optimization (38) is

$$L(\lambda,\eta) = \frac{1}{2}\lambda^T\mathbf{S}\lambda - e^T\lambda - \eta^T\lambda \tag{39}$$

where $\eta = [\eta_1, \ldots, \eta_l]^T$ are the Lagrange multipliers. According to the Karush-Kuhn-Tucker Conditions (KKT conditions), we can get

$$\frac{\partial L}{\partial \lambda} = \mathbf{S}\lambda - e - \eta = 0 \tag{40}$$

$$\lambda^T\eta = 0 \tag{41}$$

$$\lambda > 0 \tag{42}$$

From (40)–(42), we have

$$0 \leq \lambda \perp \mathbf{S}\lambda - e \geq 0 \tag{43}$$

The optimality condition (43) can be written in the following equivalent form for any positive ϑ:

$$\mathbf{S}\lambda - e = (\mathbf{S}\lambda - e - \vartheta\lambda)_+ \tag{44}$$

As mentioned above, these optimality conditions will result in a very simple iteration scheme for our Lap-LELM algorithm:

$$\lambda^{i+1} = S^{-1}[e + (S\lambda^i - e - \theta\lambda^i)_+], i = 0, 1, \ldots \tag{45}$$

Our algorithm has the property of globally linear convergence when the initial point of iteration satisfies the following conditions:

$$0 < \vartheta < \frac{2}{C} \tag{46}$$

Thusly, we can get output weight

$$\beta = h(x)\alpha = h(x)(2\gamma_A\mathbf{I} + \frac{2\gamma_I}{(l+u)^2}\mathbf{L}_p\mathbf{K})^{-1}\mathbf{J}^T\mathbf{Y}\lambda. \tag{47}$$

Based on the above discussion, the Lap-LELM algorithm is summarized as Algorithm 2.

Algorithm 2 Lap-LELM Algorithm

```
Input:
  Labeled samples $T_l = \{x_i, y_i\}_{i=1}^{l}$
  Unlabeled samples $T_u = \{x_j\}_{j=l+1}^{l+u}$;

  Step 1: Construct the graph Laplacian $\mathbf{L}_p$ from both $T_l$ and $T_u$.
  Step 2: Random input weights and biases, and calculate the output matrix of the hidden neurons $H$.
  Step 3: Choose the $\gamma_A$, $\gamma_I$ and parameter $\vartheta$.
  Step 4:
  Compute the output weights $\beta$ using (47)

  return The decision function $f(x) = sign(\beta \cdot h(x))$.
```

4.2 Compared with other related algorithms

We compare Laplacian lagrangian extreme learning machine (Lap-LELM) with other related algorithms: Laplacian support vector machine (Lap-SVM) [13], Manifold proximal support vector machine (MPSVM) [18], Semi-superviesd extreme learning machine (SS-ELM) [21], and Manifold regularized extreme learning machine (MR-ELM) [23].

Compared with Lap-SVM and MPSVM:

(1) The bias b is not required in Lap-LELM since the separating hyperplane $\boldsymbol{\beta}^T h(x) = 0$ passes through the origin in ELM feature space, while Lap-SVM and MPSVM need threshold to determine the hyperplane.

(2) Different from the Lap-SVM and MPSVM which is difficult to optimize in dealing with nonlinear problems because of the unknown implication mapping, however, the proposed Lap-LELM (33) kernel function has the explicit form: $\boldsymbol{K}_{ELM}(\boldsymbol{x}_i, \boldsymbol{x}_j) = h(\boldsymbol{x}_i)^T h(\boldsymbol{x}_j)$. Thus the proposed Lap-LELM with fast running speed has less decision variables than the existing Lap-LSVM and MPSVM.

Compared with SS-ELM and MR-ELM:

(1) The SS-ELM and MR-ELM are based on the traditional ELM framework, our proposed Lap-LELM is based on the OPT-ELM framework. Relative to SS-ELM and MR-ELM our proposed Lap-LELM is robust, since the quadratic loss function is used in SS-ELM and MR-ELM, but the hinge loss is used in the proposed Lap-LELM.

(2) Compared with SS-ELM and MR-ELM, Lap-LELM has the advantages of fast operation, global convergence and low computational burden. The Lap-LELM is solvable in a space of dimensionality equal to the number of sample points, however, the SS-ELM and MR-ELM models are solved in a space with a dimension greater than the number of sampling points, so the calculation is complicated and the computational burden is high.

5 Numerical results

To evaluate the accuracy and efficiency of the LELM and Lap-LELM algorithm, we performed experiments on benchmark datasets and four NIR spectroscopy datasets. All algorithms were implemented using MATLAB R2014b on a 3.40 GHz machine with 8 GB of memory.

5.1 Experimental design

The evaluation criteria and datasets description should be specified before presenting the experimental results.

5.1.1 Algorithm evaluation criteria

In order to evaluate the effectiveness of the proposed method, the evaluation criteria used in this paper are ACC (accuracy), F_1-measure and MCC (Matthews correlation coefficient), where ACC is the recognition rate of two samples, F_1 is the precision and recall rate of the two indicators of the harmonic average, MCC is a comprehensive evaluation criteria. The CPU-time also serves as an indicator of algorithm evaluation. They are defined as:

$$ACC = \frac{TP + TN}{TP + FN + TN + FP},$$

$$MCC = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}},$$

$$F_1 = \frac{2TP}{2TP + FP + FN}.$$

5.1.2 Datasets description

Our experiments performed on eleven UCI datasets[1] and four NIR spectroscopy datasets [5], respectively. The eleven UCI datasets are listed in Table 1.

The two classes of data are generated from different 2-dimensional normal distributions $N(\mu_1, \sigma_1)$ and $N(\mu_2, \sigma_2)$, where $\mu_1 = [4.5, 1.5]$, $\mu_2 = [1.5, -0.5]$ and $\sigma_1 = \sigma_2 = [0.5, 0.5]$ and each class contains 1000 sample points.

Licorice is a traditional Chinese herbal medicine. We utilize 244 licorice seeds including 122 hard seeds and 122 soft seeds in our experiment. Near-infrared (NIR) spectroscopic datasets of licorice seeds was obtained via an MPA spectrometer. The NIR spectral range of 5000-10000 cm^{-1} is recorded with a resolution of 4 cm^{-1}. The initial spectra are digitized by OPUS 5.5 software. To comprehension validation, numerical experiments are carried out in four different spectral regions: 5000-6000 cm^{-1}, 6000-7000 cm^{-1}, 7000-8000 cm^{-1} and 8000-10000 cm^{-1}. The corresponding spectral regions are denoted regions A, B, C and D, respectively. Information in them is summarized in Table 2.

5.2 Supervised learning results

Before experiment, all datasets are normalized to the range of [0, 1]. We choose the LSVM [24], OPT-ELM [2],

[1] http://archive.ics.uci.edu/ml/datasets.html

Table 1 Description of UCI datasets

Datasets	♯ of samples	Attribute	Datasets	♯ of samples	Attribute
Balance	576	4	Wholesale	440	7
Breast Cancer	699	9	Australian	690	14
Vote	432	16	Diabetes	1151	19
German	1000	24	WDBC	569	30
Ionosphere	350	34	QSAR	1055	41
Pima	768	8			

NLTWSVM [28], SLTWSVM [28] and SVM [11] as the baseline methods. We conduct numerical experiments on UCI datasets and NIR spectroscopy datasets, respectively. We perform ten-fold cross validation in all considered datasets. In other words, the dataset is split randomly into ten subsets, and one of those sets is reserved as a test set.

In our experiment, the RBF kernel $K(x_i, x_j) = e^{-\sigma||x_i - x_j||_2^2}$ is considered in SVM, LSVM, NLTWSVM and SLTWSVM. In the LELM and OPT-ELM model we use Sigmoid function $1/(1 + exp(-(\mathbf{w} \cdot \mathbf{x} + b)))$ ($\mathbf{w}$ and b are randomly generated) as the activation function. For SVM, LSVM, NLTWSVM, SLTWSVM and LELM, we carry out grid search and 10-fold cross-validation on the training sets to get the optimal parameters (C, ν, L, σ) with highest accuracy. All of the following experimental results are performed on these optimal parameters. The parameter C is selected from $\{10^{-5}, \cdots, 10^{5}\}$. The parameter ν is selected from $\{2^0, 2^1, \cdots, 2^8\}$. The RBF kernel parameter σ is selected from $\{2^{-5}, \cdots, 2^5\}$. The Hidden layer nodes L is selected from $\{200, 300, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000\}$. Theoretically, the larger L is and the better the generalization performance of LELM [5]. For the algorithm LSVM and LELM we set the parameter $0 < \delta < \frac{1.9}{\nu}$ and $0 < \theta < \frac{1.7}{C}$.

5.2.1 Experiments on artificial dataset

In order to verify the performance of the proposed LELM, we compared LELM with SVM, LSVM, OPT-ELM, NLTWSVM and SLTWSVM on artificial datasets. The results of this experiment are shown in Table 3. We can see that LELM outperforms SVM, LSVM, OPT-ELM, NLTWSVM and SLTWSVM in the analysis of ACC and CPU-time analysis.

Table 2 Near-infrared spectral sample regions of licorice seeds

Regions	Spectral range (cm^{-1})	Number of samples	Number of variables
Region A	5000-6000	244	520
Region B	6000-7000	244	518
Region C	7000-8000	244	778
Region D	8000-1000	244	1555

5.2.2 Experiments on UCI datasets

We compare the proposed LELM with OPT-ELM and LSVM, on eleven UCI datasets. All experimental results are shown in Figs. 1, 2, 3 and 4. Compared with the OPT-ELM and LSVM, Fig. 1 illustrate that the LELM improves generalization ability and has higher ACC values on most data sets. In Fig. 2, we compare the MCC values of the proposed LELM with OPT-ELM and LSVM on each datasets. As can be seen from Fig. 2, in addition to Prima, the MCC values of LELM is higher than that of OPT-ELM and LSVM in most cases. Further, we can also find that OPT-ELM outperforms LSVM on most datasets except Australian and Vote. Figure 3 presents a comparison of the F_1 values of the proposed LELM with LSVM and OPT-ELM on each datasets. From the experimental results in Fig. 3, we can find that in the comparison of F_1 values, LELM has better performance than OPT-ELM and LSVM in most cases. Figure 4 compare the number of support vectors for the proposed LELM with OPT-ELM and LSVM on each datasets. From Fig. 4, we find that LELM has fewer support vectors (SVs) than OPT-ELM and LSVM on most datasets. It can be further found that the number of support vectors of LELM and OPT-ELM is similar on some data sets. More precisely, the former has better performance.

To further verify the performance of our proposed LELM, we compared the proposed LELM with the traditional methods OPT-ELM and LSVM on seven UCI datasets. The experimental results are presented in Table 4. The classification accuracy and CPU-times presented in Table 4 are the average of five-times experiments. From Table 4, we can find that the LELM classification accuracy and time analysis are superior to other algorithms on the seven datasets. It is further found that the performance of

Table 3 Performance comparison of SVM, LSVM, OPT-ELM, NLTWSVM, SLTWSVM and LELM on on artificial datasets

Datasets	SVM
	ACC(%)
	(C^*, σ^*)
	CPU-time(s)
		LSVM
		ACC(%)
		(C^*, σ^*)
		CPU-time(s)
			OPT-ELM
			ACC(%)
			(C^*, L^*)
			CPU-time(s)
				NLTWSVM
				ACC(%)
				$(C_1^* = C_2^*, \sigma^*)$
				CPU-time(s)
					SLTWSVM
					ACC(%)
					$(C_1^* = C_2^*, \sigma^*)$
					CPU-time(s)
						LELM
						ACC(%)
						(C^*, L^*)
						CPU-time(s)

Artificial dataset $(1000 * 2)$	99.48 $(10^2, 2^{-1})$ 53.371	99.75 $(10^3, 2^{-3})$ 46.815	99.81 $(10^2, 2000)$ 39.646	99.73 $(10^3, 2^{-1})$ 11.253	99.73 $(10^1, 2^{-2})$ 11.247	99.85 $(10^2, 2000)$ 9.736

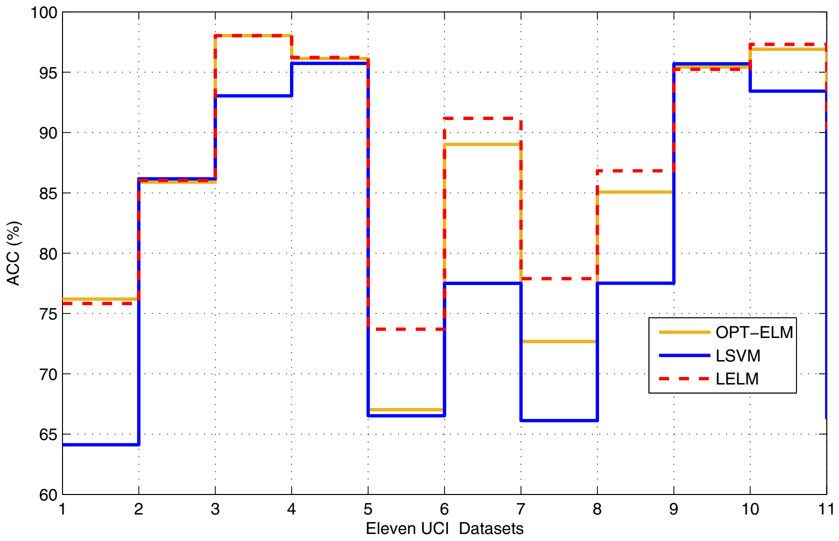

Fig. 1 Performance comparison of ACC (%) of OPT-ELM, LSVM and LELM on eleven UCI datasets, where 1 denote Diabetes, 2 denote Australian, 3 denote Balance, 4 denote Breast Cancer, 5 denote German, 6 denote Ionosphere, 7 denote Pima, 8 denote QSAR, 9 denote Vote, 10 denote WDBC, 11 denote Wholeasle

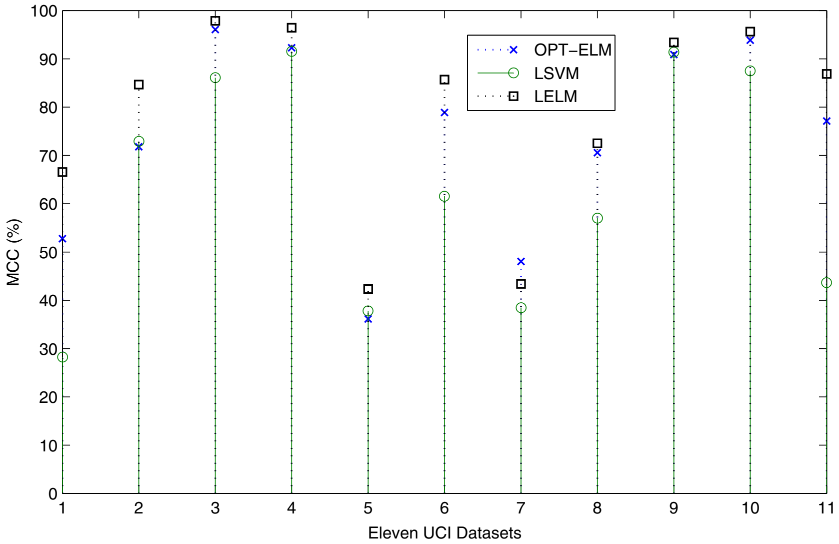

Fig. 2 Performance comparison of MCC (%) of OPT-ELM, LSVM and LELM on eleven UCI datasets, where 1 denote Diabetes, 2 denote Australian, 3 denote Balance, 4 denote Breast Cancer, 5 denote German, 6 denote Ionosphere, 7 denote Pima, 8 denote QSAR, 9 denote Vote, 10 denote WDBC, 11 denote Wholeasle

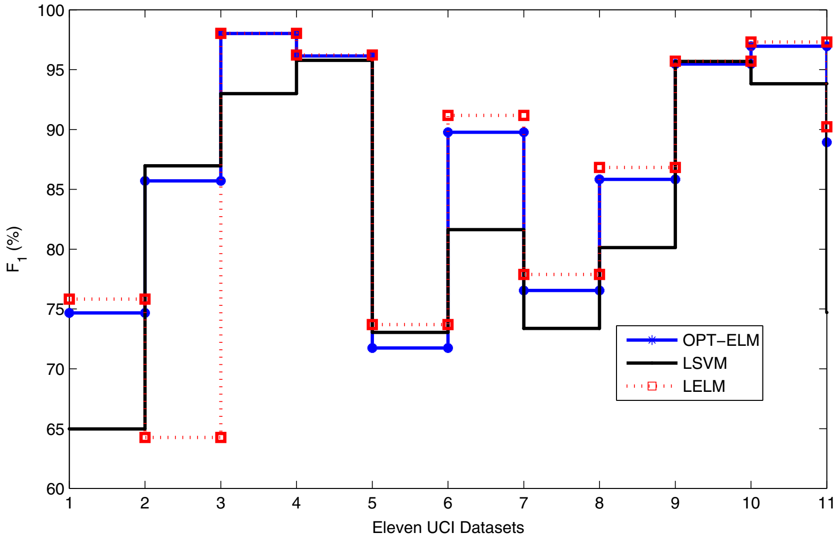

Fig. 3 Performance comparison of F_1 (%) of OPT-ELM, LSVM and LELM on eleven UCI datasets, where 1 denote Diabetes, 2 denote Australian, 3 denote Balance, 4 denote Breast Cancer, 5 denote German, 6 denote Ionosphere, 7 denote Pima, 8 denote QSAR, 9 denote Vote, 10 denote WDBC, 11 denote Wholeasle

the LELM we proposed on most of the data sets is similar to that of the NLTWSVM and SLTWSVM. To be precise, LELM is better than NLTWSVM and SLTWSVM. The above analysis of experimental results further validates that our proposed algorithm is effective and reliable, which fully demonstrates the correctness of our theory.

5.2.3 Experiments on NIR spectroscopy datasets

We compared the proposed LELM with the SVM, LSVM, OPT-ELM NLTWSVM and SLTWSVM on NIR spectroscopy datasets. Our experimental results are presented in Table 5. We find from Table 5 that the LELM achieves better performance than SVM, LSVM, OPT-ELM NLTWSVM and SLTWSVM with respect to the ACC and CPU-time analysis. Further find that the performance of our proposed LELM are very similar to that of NLTWSVM and SLTWSVM on the NIR spectroscopy datasets. Accurately our algorithm is better than NLTWSVM and SLTWSVM.

5.3 Semi-supervised learning results

In order to verify the generalization performance of the proposed semi-supervised LELM algorithm, we conducted

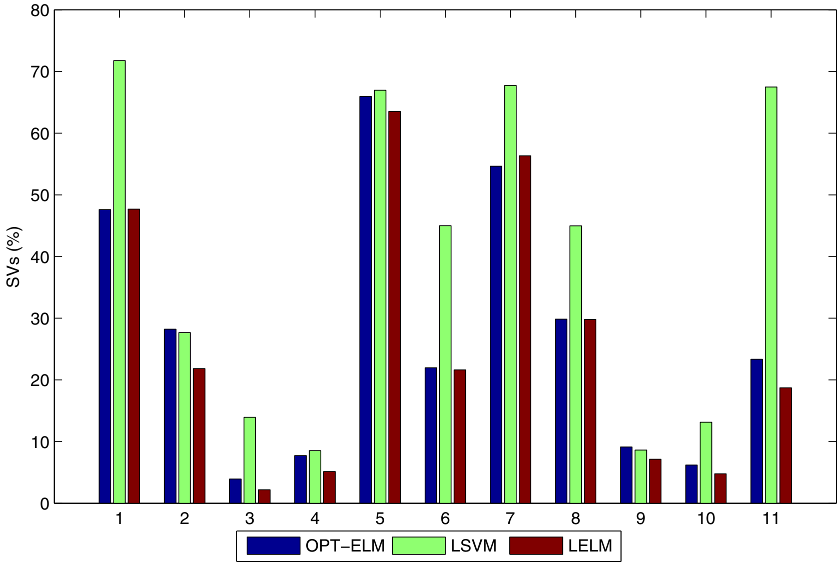

Fig. 4 Performance comparison of SVs (%) of OPT-ELM, LSVM and LELM on eleven UCI datasets, where 1 denote Diabetes, 2 denote Australian, 3 denote Balance, 4 denote Breast Cancer, 5 denote German, 6 denote Ionosphere, 7 denote Pima, 8 denote QSAR, 9 denote Vote, 10 denote WDBC, 11 denote Wholeasle

Table 4 Performance comparison of SVM, LSVM, OPT-ELM, NLTWSVM, SLTWSVM and LELM on UCI datasets

Datasets	SVM	LSVM	OPT-ELM	NLTWSVM	SLTWSVM	LELM
	ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)
	(C^*, σ^*)	(C^*, σ^*)	(C^*, L^*)	$(C_1^* = C_2^*, \sigma^*)$	$(C_1^* = C_2^*, \sigma^*)$	(C^*, L^*)
	CPU-time	CPU-time	CPU-time	CPU-time	CPU-time	CPU-time
Australian	85.65	86.17	85.89	87.25	87.25	89.71
	$(10^{-1}, 2^{2})$	$(10^{-3}, 2^{1})$	$(10^{2}$,1000)	$(10^{-5}, 2^{3})$	$(10^{-5}, 2^{3})$	$(10^{3}$,1000)
	12.111	10.563	8.113	0.989	1.046	0.781
Breast cancer	97.08	95.74	96.14	97.38	97.38	98.55
	$(10^{-1}, 2^{-1})$	$(10^{-2}, 2^{-1})$	$(10^{-1}$,500)	$(10^{-3}, 2^{1})$	$(10^{-3}, 2^{1})$	$(10^{-2}$,500)
	11.940	8.634	7.691	1.008	1.034	1.004
Ionosphere	92.03	89.02	91.50	94.02	93.48	95.16
	$(10^{0}, 2^{0})$	$(10^{1}, 2^{1})$	$(10^{2}$,500)	$(10^{0}, 2^{2})$	$(10^{-4}, 2^{-1})$	$(10^{3}$,500)
	3.131	3.218	1.017	0.277	0.256	0.214
Vote	96.09	95.43	95.70	96.79	96.34	97.62
	$(10^{0}, 2^{0})$	$(10^{2}, 2^{-1})$	$(10^{2}$,500)	$(10^{1}, 2^{5})$	$(10^{1}, 2^{5})$	$(10^{2}$,500)
	4.792	4.801	5.643	0.405	0.509	0.472
WDBC	98.07	96.75	97.41	98.25	98.42	98.51
	$(10^{1}, 2^{-1})$	$(10^{2}, 2^{-2})$	$(10^{2}$,500)	$(10^{1}, 2^{1})$	$(10^{1}, 2^{1})$	$(10^{2}$,1000)
	8.277	7.794	5.931	0.830	1.104	0.672
German	76.90	75.99	75.90	76.50	76.50	78.01
	$(10^{2}, 2^{4})$	$(10^{1}, 2^{3})$	$(10^{3}$,1000)	$(10^{-1}, 2^{4})$	$(10^{-1}, 2^{4})$	$(10^{2}$,1000)
	26.197	19.675	15.587	2.368	2.493	2.237
Pima	77.79	76.75	77.08	77.14	77.14	78.95
	$(10^{3}, 2^{3})$	$(10^{1}, 2^{-2})$	$(10^{3}$,500)	$(10^{0}, 2^{1})$	$(10^{0}, 2^{1})$	$(10^{1}$,1000)
	8.880	9.715	6.631	1.193	1.897	1.156

Table 5 Performance comparison of SVM, LSVM, OPT-ELM, NLTWSVM, SLTWSVM and LELM on NIR spectroscopy datasets

Datasets	SVM	LSVM	OPT-ELM	NLTWSVM	SLTWSVM	LELM
	ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)
	(C^*, σ^*)	(C^*, σ^*)	(C^*, L^*)	$(C_1^* = C_2^*, \sigma^*)$	$(C_1^* = C_2^*, \sigma^*)$	(C^*, L^*)
	CPU-time(s)	CPU-time(s)	CPU-time(s)	CPU-time(s)	CPU-time(s)	CPU-time(s)
Region A	70.48	66.67	71.75	71.13	71.13	72.50
	$(10^{1}, 2^{-3})$	$(10^{2}, 2^{-3})$	$(10^{2}, 300)$	$(10^{3}, 2^{-1})$	$(10^{3}, 2^{-1})$	$(10^{3}, 300)$
	1.712	1.461	1.651	0.663	0.704	0.573
Region B	72.27	72.08	74.37	74.56	74.56	75.83
	$(10^{1}, 2^{-2})$	$(10^{-3}, 2^{-3})$	$(10^{3}, 300)$	$(10^{-3}, 2^{1})$	$(10^{-3}, 2^{1})$	$(10^{1}, 500)$
	3.128	2.802	2.361	1.375	1.411	0.554
Region C	61.67	61.32	62.78	62.42	62.42	63.17
	$(10^{2}, 2^{0})$	$(10^{-3}, 2^{-1})$	$(10^{1}, 300)$	$(10^{-2}, 2^{-3})$	$(10^{-2}, 2^{-3})$	$(10^{3}, 300)$
	1.115	1.137	1.342	0.867	0.732	0.503
Region D	72.06	72.08	72.65	72.33	72.33	73.75
	$(10^{-2}, 2^{-3})$	$(10^{4}, 2^{-1})$	$(10^{-2}, 500)$	$(10^{-1}, 2^{0})$	$(10^{-1}, 2^{0})$	$(10^{-1}, 1000)$
	1.429	1.154	1.367	0.675	0.713	0.547

Table 6 Performance comparison of LapSVM, LELM, SS-ELM, MR-ELM and Lap-LELM on artificial dataset

Datasets	LapSVM ACC(%) $(C^*, \gamma_A^*, \gamma_I^*, \sigma^*)$	LELM ACC(%) (C^*, L^*)	SS-ELM ACC(%) (C^*, λ^*, L^*)	MR-ELM ACC(%) $(C^*, \gamma_A^*, \gamma_I^*, L^*)$	Lap-LELM ACC(%) $(C^*, \gamma_A^*, \gamma_I^*, L^*)$
Artificial dataset $(2000 * 2)$	97.65 $(10^{-1}, 10^1, 10^3, 2^{-2})$	95.37 $(10^{-2}, 2000)$	97.58 $(10^1, 10^{-2}, 2000)$	97.81 $(10^2, 10^1, 10^2, 2000)$	97.91 $(10^3, 10^0, 10^3, 2000)$

experiments on eleven UCI datasets, COIL20(B) and USPST(B)[2] datasets, respectively, and five times 10-fold cross validation. We choose LapSVM [13], SS-ELM [21] and MR-ELM [23] as the benchmark comparison algorithm.

Before experiment, all datasets are normalized to the range of [0, 1]. The LapSVM, SS-ELM, MR-ELM and Lap-LELM constructed data adjacency graphs using k-nearest neighbors. Binary edge weights were chosen, and the neighborhood size k was set to be 9 for all the all datasets. We used the Sigmoid function $1/(1 + exp(-(\mathbf{w} \cdot \mathbf{x} + b)))$ as the activation function for SS-ELM, MR-ELM, LELM and Lap-LELM. The RBF kernel $K(x_i, x_j) = e^{-\sigma||x_i - x_j||_2^2}$ is considered in LapSVM. We carry out grid search and ten-fold cross-validation on the training sets to get the optimal $(C, \gamma_A, \gamma_I, \sigma, \lambda)$ with highest accuracy. γ_A and γ_I are selected from $\{10^0, \cdots, 10^3\}$; C and λ are selected from $\{10^{-5}, \cdots, 10^5\}$; RBF kernel parameter σ is selected from $\{2^{-5}, \cdots, 2^5\}$. The Hidden nodes L is selected from {200, 300, 500, 1000, 1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000}. All the following experimental results are performed in the optimal parameters.

5.3.1 Experiments on artificial datasets

In order to further verify the generalization performance of proposed Lap-LELM, we performed experiments on artificial data. The experimental results are shown in Table 6. We find that Lap-LELM are better than LapSVM, LELM, SS-ELM and MR-ELM in classification accuracy.

5.3.2 Experiments on UCI datasets

In this subsection, we perform LELM, LapSVM, SS-ELM, MR-ELM and Lap-LELM on eleven UCI datasets. Most of these datasets have appeared in previous experiments. We conduct the experiments with different proportions of labeled samples, i.e., 10% and 20%. The testing accuracy are computed using standard 10-fold cross validation. For supervised algorithm (LELM), training samples were divided into 10 parts, one (when 20% samples were labeled, this should be two) part for training and the rest for testing, and for semi-supervised algorithm (LapSVM, SS-ELM, MR-ELM Lap-LELM), all samples are involved in training.

We compare the proposed classification accuracy of Lap-LELM with LELM, LapSVM, SS-ELM and MR-ELM. All experimental results are based on the optimal parameters. The classification accuracy was the average of five-time experiments. Results are listed in Table 7. Let's summarize the results in a simple language. We find from Table 7 that the performance of Lap-LELM is better than that of LELM on all datasets. This shows that in the semi-supervised learning problem the use of the graph Laplacian operator can improve the classification accuracy of the model. It is further found that the propose Lap-LELM is superior to the traditional semi-supervised learning algorithm Lap-SVM on the vast majority datasets. We can also find that our algorithm Lap-LELM outperforms ELM-based semi-supervised algorithms SS-ELM and MR-ELM on most datasets. The analysis of experimental results further validates that the introduction of manifold regularization in the LELM framework can effectively improve the performance of LELM when the sample information is insufficient.

To further verify the performance of our proposed Lap-LELM. We compared the proposed Lap-LELM with OPT-ELM and Lap-SVM on six UCI datasets of different proportioned label samples (10%, 30%, 50%, 70%, 90%). The experimental results are presented in Fig. 5. Figure 5 shows the performance of LapSVM, LELM and Lap-LELM on Balance, Breast Cancer, Australian, WDBC, Vote and Ionosphere with different number of labeled samples. In this experiment, all the settings are similar to above, except that we varied the proportion of labeled data in the training set. We can observe that LapLSVM and Lap-LELM outperformed LELM significantly when there is a small amount of labeled data. Further, we can find that the classification accuracy of LELM will grow with the gradually increasing of the labeled samples number.

5.3.3 Experimental results on COIL20(B) and USPST(B) datasets

As we all know, COIL20(B) and USPST(B) are widely used in semi-supervised learning algorithm evaluation datasets.

[2] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html

Table 7 Performance comparison of LapSVM, LELM, SS-ELM, MR-ELM and Lap-LELM on UCI datasets

Datasets	Labeled Ratio	LapSVM	LELM	SS-ELM	MR-ELM	Lap-LELM
		ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)
		$(C^*, \gamma_A^*, \gamma_I^*, \sigma^*)$	(C^*, L^*)	(C^*, λ^*, L^*)	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$
Diabetic	10%	59.62	49.12	59.07	59.07	58.78
		$(10^{-2}, 10^1, 10^0, 2^{-1})$	$(10^{-1}, 1000)$	$(10^1, 10^{-2}, 1000)$	$(10^1, 10^1, 10^0, 500)$	$(10^3, 10^1, 10^3, 1000)$
	20%	59.58	59.62	59.34	59.34	59.65
		$(10^{-2}, 10^1, 10^0, 2^0)$	$(10^{-1}, 1000)$	$(10^1, 10^{-2}, 1000)$	$(10^3, 10^0, 10^0, 1000)$	$(10^3, 10^1, 10^1, 1000)$
Australian	10%	85.28	52.89	84.11	84.11	84.05
		$(10^0, 10^2, 10^1, 2^{-3})$	$(10^1, 500)$	$(10^{-2}, 10^{-1}, 500)$	$(10^3, 10^0, 10^3, 500)$	$(10^3, 10^2, 10^3, 1000)$
	20%	86.31	67.6	85.86	85.86	85.73
		$(10^0, 10^2, 10^2, 2^{-3})$	$(10^1, 500)$	$(10^{-2}, 10^0, 500)$	$(10^3, 10^0, 10^3, 500)$	$(10^3, 10^1, 10^0, 1000)$
Balance						
	10%	82.12	51.79	84.01	84.01	84.55
		$(10^1, 10^2, 10^1, 2^0)$	$(10^3, 500)$	$(10^1, 10^{-1}, 500)$	$(10^{-1}, 10^1, 10^3, 500)$	$(10^{-2}, 10^2, 10^0, 500)$
	20%	87.12	63.75	88.85	88.85	89.36
		$(10^2, 10^2, 10^1, 2^{-3})$	$(10^2, 500)$	$(10^3, 10^{-1}, 500)$	$(10^{-1}, 10^1, 10^3, 500)$	$(10^{-2}, 10^2, 10^0, 500)$
Breast Cancer						
	10%	96.57	51.39	91.23	91.23	92.74
		$(10^{-2}, 10^0, 10^1, 2^{-1})$	$(10^{-1}, 500)$	$(10^1, 10^{-2}, 500)$	$(10^0, 10^1, 10^1, 500)$	$(10^2, 10^3, 10^0, 1000)$
	20%	96.62	61.45	94.74	94.74	94.35
		$(10^{-1}, 10^0, 10^1, 2^0)$	$(10^2, 500)$	$(10^1, 10^{-2}, 500)$	$(10^1, 10^1, 10^1, 500)$	$(10^3, 10^3, 10^0, 1000)$
WDBC						
	10%	93.32	50.33	93.57	93.57	94.09
		$(10^3, 10^0, 10^1, 2^1)$	$(10^2, 500)$	$(10^3, 10^0, 500)$	$(10^{-2}, 10^1, 10^2, 500)$	$(10^{-1}, 10^3, 10^1, 500)$
	20%	93.35	63.33	95.08	95.08	95.15
		$(10^3, 10^0, 10^2, 2^1)$	$(10^2, 500)$	$(10^3, 10^1, 500)$	$(10^{-2}, 10^3, 10^2, 500)$	$(10^{-1}, 10^2, 10^1, 500)$
German						
	10%	66.96	51.13	66.89	66.89	67.66
		$(10^{-3}, 10^1, 10^3, 2^{-2})$	$(10^0, 500)$	$(10^{-1}, 10^{-2}, 500)$	$(10^1, 10^0, 10^0, 500)$	$(10^{-3}, 10^1, 10^0, 500)$
	20%	69.11	68.93	68.91	68.91	69.11
		$(10^{-3}, 10^1, 10^3, 2^{-2})$	$(10^0, 500)$	$(10^{-1}, 10^{-2}, 500)$	$(10^1, 10^0, 10^1, 500)$	$(10^{-3}, 10^1, 10^1, 500)$
Ionosphere						
	10%	82.47	52.19	82.39	82.39	83.54
		$(10^1, 10^1, 10^2, 2^1)$	$(10^2, 500)$	$(10^{-2}, 10^{-3}, 500)$	$(10^{-1}, 10^1, 10^3, 500)$	$(10^2, 10^3, 10^0, 500)$
	20%	86.08	64.19	87.26	87.26	87.43
		$(10^1, 10^1, 10^1, 2^1)$	$(10^2, 500)$	$(10^{-2}, 10^0, 500)$	$(10^{-1}, 10^2, 10^2, 500)$	$(10^2, 10^3, 10^1, 500)$
Pima						
	10%	71.81	51.79	71.18	71.18	70.18
		$(10^0, 10^2, 10^3, 2^{-1})$	$(10^0, 500)$	$(10^{-1}, 10^{-1}, 500)$	$(10^3, 10^2, 10^0, 500)$	$(10^3, 10^1, 10^3, 1000)$
	20%	73.64	63.55	73.38	73.38	73.26
		$(10^1, 10^2, 10^3, 2^{-1})$	$(10^0, 500)$	$(10^{-1}, 10^{-1}, 500)$	$(10^1, 10^2, 10^0, 500)$	$(10^2, 10^1, 10^3, 1000)$
QSAR						
	10%	75.70	51.41	76.24	76.24	77.52
		$(10^3, 10^1, 10^1, 2^{-3})$	$(10^2, 1000)$	$(10^{-3}, 10^{-3}, 1000)$	$(10^1, 10^0, 10^1, 1000)$	$(10^2, 10^1, 10^2, 1000)$
	20%	79.48	62.95	79.66	79.66	81.17
		$(10^2, 10^2, 10^1, 2^{-2})$	$(10^2, 1000)$	$(10^{-3}, 10^2, 1000)$	$(10^2, 10^1, 10^1, 1000)$	$(10^2, 10^2, 10^3, 1000)$

Table 7 (continued)

Datasets	Labeled Ratio	LapSVM	LELM	SS-ELM	MR-ELM	Lap-LELM
		ACC(%)	ACC(%)	ACC(%)	ACC(%)	ACC(%)
		$(C^*, \gamma_A^*, \gamma_I^*, \sigma^*)$	(C^*, L^*)	(C^*, λ^*, L^*)	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$
Vote						
	10%	90.09	51.63	89.97	89.97	90.16
		$(10^2, 10^3, 10^1, 2^0)$	$(10^1, 500)$	$(10^3, 10^0, 500)$	$(10^{-1}, 10^1, 10^1, 500)$	$(10^0, 10^3, 10^1, 500)$
	20%	90.17	59.83	90.31	90.31	91.45
		$(10^1, 10^3, 10^1, 2^0)$	$(10^3, 500)$	$(10^2, 10^0, 500)$	$(10^{-1}, 10^1, 10^1, 500)$	$(10^2, 10^3, 10^1, 500)$
Wholesale						
	10%	71.93	50.85	72.21	72.21	72.65
		$(10^2, 10^2, 10^3, 2^3)$	$(10^1, 500)$	$(10^{-2}, 10^{-2}, 500)$	$(10^3, 10^1, 10^0, 500)$	$(10^1, 10^2, 10^2, 500)$
	20%	76.30	62.3	76.56	76.56	76.89
		$(10^3, 10^3, 10^3, 2^3)$	$(10^0, 500)$	$(10^{-2}, 10^1, 500)$	$(10^2, 10^1, 10^1, 500)$	$(10^1, 10^3, 10^2, 500)$

We compared the proposed Lap-LELM with Lap-SVM, SS-ELM and MR-ELM on COIL20(B) and USPST(B) datasets

The COIL20(B) and USPST(B) datasets are described as follows:

The Columbia object image library (COIL20) is a set of 1440 gray-scale images of 20 different objects. Each sample represents a 32 × 32 gray scale image of an object acquired from a specific view. The COIL20(B) is a binary data set generated by grouping the first 10 objects in COIL20 to class 1 and the remaining objects to class 2.

The USPST data set is a collection of hand-written digits from the USPS postal system. Each digit image is represented by a resolution of 16 × 16 pixels. The USPST(B) is a binary data set which was built by grouping the first 5 digits to class 1 and the remaining digits to class 2.

The proposed Lap-LELM is compared with Lap-SVM, MR-ELM and SS-ELM. We use the classification accuracy on USPST(B)and COIL20(B) datasets to evaluate the performance of these algorithms. The experimental results are shown in Table 8. All experimental results are performed under optimal parameters. From Table 8, we can see that our method is better than Lap-SVM semi-supervised learning algorithms on on both USPST(B)and COIL20(B). It can be further found that the proposed method has good performance compared to other ELM-based semi-supervised algorithms. The above experimental analysis further validates that our proposed Lap-LELM is effective and reliable.

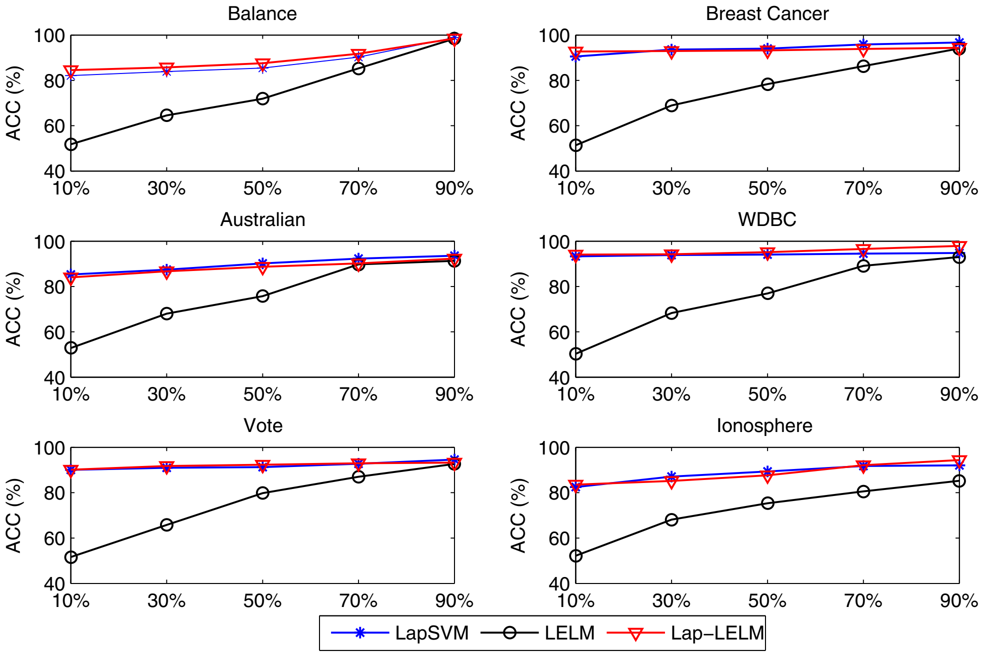

Fig. 5 Testing ACC with respect to different number of labeled data

Table 8 Performance comparison of the OPT-ELM, LapWSC-SVM, Lap-ELM and LapWSC-ELM

	LapSVM	SS-ELM	MR-ELM	Lap-LELM
Datasets	ACC(%)	ACC(%)	ACC(%)	ACC S(%)
	$(C^*, \gamma_A^*, \gamma_I^*, \sigma^*)$	(C^*, λ^*, L^*)	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$	$(C^*, \gamma_A^*, \gamma_I^*, L^*)$
COIL20(B)	92.51	92.61	91.76	93.95
	$(10^3, 10^1, 10^3, 10^{-1})$	$(10^3, 10^{-3}, 1000)$	$(10^2, 10^3, 10^2, 1000)$	$(10^3, 10^3, 10^2, 2000)$
USPST(B)	73.24	90.51	90.92	92.39
	$(10^2, 10^3, 10^0, 10^{-2})$	$(10^{-2}, 10^0,1000)$	$(10^{-3}, 10^1, 10^2, 1000)$	$(10^{-2}, 10^1, 10^3, 2000)$

6 Conclusion

In this paper, we have first proposed a new type of lagrange extreme learning machine (LELM) based on the optimization theory. Then, a semi-supervised lagrangian extreme learning machine (Lap-LELM) is proposed via extending LELM to a semi-supervised learning framework, which incorporates the manifold regularization into LELM to improve performance when insufficient labeled samples are available. Compared to existing supervised and semi-supervised ELM algorithms, the proposed LELM and Lap-LELM maintain almost all the advantages of ELMs, such as the remarkable training efficiency for binary classification problems. In addition, through the SMW identities, LELM and Lap-LELM are transformed into two smaller unconstrained optimizations. At the same time, two very simple iterative algorithms are constructed to solve the two unconstrained optimization problems. Theoretical analysis and numerical experiments show that our iterative algorithms are globally converged, have a low computational burden and a certain degree of generalization performance compared with traditional learning algorithms.

In the near future, we will further optimize our proposed framework and study the sparse regularization problem for our framework. In addition, we will extend our method to multi-class classification and some practical applications.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (No11471010) and Chinese Universities Scientific Fund.

References

1. Huang G, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
2. Huang G, Ding XJ, Zhou HM (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74:155–163
3. Huang G, Huang G, Song S, You KY (2015) Trends in extreme learning machines: a review. Neural Netw 61:32–48
4. Yang L, Zhang S (2017) A smooth extreme learning machine framework. J Intell Fuzzy Syst 33(6):3373–3381
5. Yang L, Zhang S (2016) A sparse extreme learning machine framework by continuous optimization algorithms and its application in pattern recognition. Eng Appl Artif Intel 53(C):176–189
6. Wang Y, Cao F, Yuan Y (2011) A study on effectiveness of extreme learning machine. Neurocomputing 74(16):2483–2490
7. Wang G, Lu M, Dong YQ, Zhao XJ (2016) Self-adaptive extreme learning machine. Neural Comput Appl 27(2):291–303
8. Zhang W, Ji H, Liao G, Zhang Y (2015) A novel extreme learning machine using privileged information. Neurocomputing 168(C):823–828
9. Zhang Y, Wu J, Cai Z, Zhang P, Chen L (2016) Memetic extreme learning machine. Pattern Recogn 58(C):135–148
10. Ding XJ, Lan Y, Zhang ZF, Xu X (2017) Optimization extreme learning machine with ν regularization. Neurocomputing
11. Vapnik, Vladimir N (2002) The nature of statistical learning theory. IEEE Trans Neural Netw 8(6):1564–1564
12. Belkin M, Niyogi P (2004) Semi-supervised learning on riemannian manifolds. Mach Learn 56(1-3):209–239
13. Belkin M, Niyogi P, Sindhwani V (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. JMLR.org
14. Xiaojin Z (2006) Semi-supervised learning literature sur-vey. Semi-Supervised Learning Literature Sur-vey, Technical report, Computer Sciences. University of Wisconsin-Madisoa 37(1):63–77
15. Chapelle O, Sindhwani V, Keerthi SS (2008) Optimization techniques for semi-supervised support vector machines. J Mach Learn Res 9(1):203–233
16. Wang G, Wang F, Chen T, Yeung DY, Lochovsky FH (2012) Solution path for manifold regularized semisupervised classification. IEEE Trans Syst Man Cybern Part B Cybern A Publ IEEE Syst Man Cybern Soc 42(2):308
17. Melacci S, Belkin M (2009) Laplacian support vector machines trained in the primal. J Mach Learn Res 12(5):1149–1184
18. Chen WJ, Shao YH, Xu DK, Fu YF (2014) Manifold proximal support vector machine for semi-supervised classification. Appl Intell 40(4):623–638
19. Deng W, Zheng Q, Chen L (2009) Regularized extreme learning machine. In: IEEE Symposium on computational intelligence and data mining, 2009. CIDM ’09. IEEE, pp 389–395
20. Iosifidis A, Tefas A, Pitas I (2014) Semi-supervised classification of human actions based on neural networks. In: International conference on pattern recognition, vol 15. IEEE, pp 1336–1341
21. Huang G, Song S, Gupta JND, Wu C (2014) Semi-supervised and unsupervised extreme learning machines. IEEE Trans Cybern 44(12):2405
22. Zhou Y, Liu B, Xia S, Liu B (2015) Semi-supervised extreme learning machine with manifold and pairwise constraints regularization. Neurocomputing 149(PA):180–186

23. Liu B, Xia SX, Meng FR, Zhou Y (2016) Manifold regularized extreme learning machine. Neural Comput Applic 27(2):255–269
24. Mangasarian O, Musicant L, David R (2001) Lagrangian support vector machines. J Mach Learn Res 1(3):161–177
25. Balasundaram S, Tanveer M (2013) On lagrangian twin support vector regression. Neural Comput Appl 22(1):257–267
26. Tanveer M, Shubham K, Aldhaifallah M, Nisar KS (2016) An efficient implicit regularized lagrangian twin support vector regression. Appl Intell 44(4):1–18
27. Shao YH, Chen WJ, Zhang JJ, Wang Z, Deng NY (2014) An efficient weighted lagrangian twin support vector machine for imbalanced data classification. Pattern Recogn 47(9):3158–3167
28. Balasundaram S, Gupta D, Prasad SC (2016) A new approach for training lagrangian twin support vector machine via unconstrained convex minimization. Appl Intell 46(1):1–11
29. Balasundaram S, Gupta D (2014) On implicit lagrangian twin support vector regression by newton method International. J Comput Intell Syst 7(1):50–64
30. Tanveer M, Shubham K (2017) A regularization on lagrangian twin support vector regression. Int J Mach Learn Cybern 8(3):807–821
31. Balasundaram S, Gupta D (2014) Training lagrangian twin support vector regression via unconstrained convex minimization. Knowl-Based Syst 59(59):85–96
32. Tanveer M (2015) Newton method for implicit lagrangian twin support vector machines. Int J Mach Learn Cybern 6(6):1029–1040
33. Shao YH, Hua XY, Liu LM, Yang ZM, Deng NY (2015) Combined outputs framework for twin support vector machines. Appl Intell 43(2):424–438
34. Bertsekas DP (1997) Nonlinear programming. J Oper Res Soc 48(3):334–334