Neural Comput & Applic
DOI 10.1007/s00521-014-1777-8

Manifold regularized extreme learning machine

Bing Liu · Shi-Xiong Xia · Fan-Rong Meng · Yong Zhou

Received: 7 January 2014 / Accepted: 29 November 2014

Abstract Extreme learning machine (ELM) works for generalized single-hidden-layer feedforward networks (SLFNs), and its essence is that the hidden layer of SLFNs need not be tuned. But ELM only utilizes labeled data to carry out the supervised learning task. In order to exploit unlabeled data in the ELM model, we first extend the manifold regularization (MR) framework and then demonstrate the relation between the extended MR framework and ELM. Finally, a manifold regularized extreme learning machine is derived from the proposed framework, which maintains the properties of ELM and can be applicable to large-scale learning problems. Experimental results show that the proposed semi-supervised extreme learning machine is the most cost-efficient method. It tends to have better scalability and achieve satisfactory generalization performance at a relatively faster learning speed than traditional semi-supervised learning algorithms.

Keywords Manifold regularization · Extreme learning machine (ELM) · Random feature mapping · Semi-supervised learning

1 Introduction

Lately, extreme learning machine (ELM) has been attracting a lot of attention from an increasing number of researchers [1–5]. It was originally developed for the single-hidden-layer feedforward neural networks [6–8], which was extended to the "generalized" SLFNs, i.e., may not be neuron alike [9, 10]. ELM has been already successfully applied in many domains such as bioinformatics [11–13], computer vision [14], data mining [15], robotics [16] and the reduced-reference assessment of perceived image quality [17, 18]. It has three main learning features: (1) The hidden layer does not need be tuned. Essentially, ELM was originally proposed to apply random computational nodes in the hidden layer. According to ELM theories [6–10], almost all nonlinear piecewise continuous functions used as feature mapping can make ELM satisfy universal approximation capability. (2) ELM aims to reach not only the smallest training error but also the smallest norm of output weights. Actually, SVM's maximal separating margin property and the ELM's minimal norm of output weight property are consistent [27]. (3) Different variants of SVM are required for different types of applications [19], while ELM provides a unified solution to different practical applications (e.g., regression, binary and multi-class classifications).

Semi-supervised learning is a class of machine learning techniques that typically make use of a small amount of labeled data with a large amount of unlabeled data. ELM was originally proposed for solving fast supervised learning problems. In fact, collecting a fully labeled training set is infeasible due to the high cost in manually labeling data. Thus, it is not applicable to semi-supervised learning problems. In addition, for the random choice of input weights and biases, the ELM algorithm without regularization sometimes does not make the hidden-layer output matrix $\boldsymbol{H}$ full column rank, which may affect the effectiveness of ELM [20, 21]. One way to address these problems of ELM is to exploit unlabeled samples by semi-

B. Liu · S.-X. Xia (✉) · F.-R. Meng (✉) · Y. Zhou
School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China
e-mail: xiasx@cumt.edu.cn

F.-R. Meng
e-mail: mengfr@cumt.edu.cn

Published online: 28 February 2015

supervised learning methods based on the manifold regularization. Belkin et al. [22] proposed a general manifold regularization (MR) framework developed in the setting of reproducing kernel Hilbert spaces (RKHS). This framework added an additional penalty term to the traditional regularization model and used it to measure the smoothness of functions on data manifolds. Such a term can improve the performance of learning algorithms by exploiting the intrinsic structure of data. Based on the MR framework, the discriminatively regularization least square classification (DRLSC) method built the penalty term on manifolds by integrating both discriminative and geometrical information in each local region [23]. In [24], a sparse regularization method for semi-supervised classification and sparse regularized least square classification (S-RLSC) algorithm were proposed, which improved the MR framework by sparse representations of data. Although these frameworks can handle semi-supervised learning problems and have the analytic solutions, they still need expensive computation when training large-scale datasets.

In order to deal with the high computational complexity of traditional manifold regularized algorithms, some regularized ELM methods have been proposed to handle fast semi-supervised problems. Li et al. [25] described a new regularization classification method (NRCM) based on ELM. But they also introduced some parameters in their method, which may be difficult to tune in practical applications. Liu et al. [26] presented a semi-supervised ELM (SELM) based on the manifold regularization. However, the penalty norm of the output weights is not added into the models of these two methods, that is, the complexity of the function in the ambient space is not effectively controlled, which could have a negative impact on the performance of algorithms according to structural risk minimization principle [22, 27]. To solve these problems, we establish the relationship between the MR framework and ELM, which indicates that ELM can be naturally introduced into the traditional MR framework and enhance the speed of traditional manifold regularized algorithms. Meanwhile, we incorporate manifold regularization terms and the penalty norm of the output weights into our model. Thus, compared with other manifold regularized ELM algorithms, the proposed manifold regularized extreme learning machine (MR-ELM) can not only generate more smooth decision functions, but also inherit the advantages of less computational complexity. Besides, similar to original ELM, it can also provide a unified solution to different practical applications.

In particular, the following contributions have been made in this paper.

1. An extended manifold regularized framework (E-MR) is presented, and the theorem about the form of decision functions is extended by substituting the function of MR framework with vector functions.
2. We demonstrate the relation between the MR framework and ELM. By discretizing the function $k(\boldsymbol{x},\cdot)$ of a reproducing kernel Hilbert spaces (RKHS), ELM with single output can be derived from the MR framework, and ELM with multi-outputs can be derived from the E-MR framework.
3. MR-ELM can be unified into the E-MR framework. Experiments on real-world datasets verify the effectiveness and outstanding classification performance of our approach.
4. Compared to other semi-supervised algorithms, MR-ELM is not the best but it is the most cost-efficient method.

The rest of this paper is organized as follows. Some previous works are introduced in Sect. 2. The relationship between the manifold regularization framework and ELM is discussed in Sect. 3. MR-ELM and its algorithm are presented in Sect. 4. Then in Sect. 5, the experiments using benchmark real-world datasets are reported. Finally, we conclude this paper in Sect. 6.

2 Extreme learning machine

The output function of ELM for generalized SLFNs in the case of one output node is

$$f_L(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i h_i(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_L]^T$ is the vector of the weights between a hidden layer of L nodes and the output node. Note that $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \ldots, h_L(\boldsymbol{x})]$ is the output (row) vector of the hidden layer with respect to the input $\boldsymbol{x}$. In fact, $\boldsymbol{h}(\boldsymbol{x})$ maps the data from the d-dimensional input space to the L-dimensional hidden-layer feature space (ELM feature space) $\boldsymbol{H}$. ELM is meant to minimize the training error as well as the norm of the output weights [6, 7]

$$\min_{\beta} \frac{C}{2}\|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{T}\|^2 + \frac{1}{2}\|\boldsymbol{\beta}\|^2, \tag{2}$$

where C is a trade-off parameter between the complexity and fitness of the decision function, and $\boldsymbol{H}$ is the hidden-layer output matrix, which is denoted by

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_1) \\ \boldsymbol{h}(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_n) \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{x}_1) & \ldots & h_L(\boldsymbol{x}_1) \\ h_1(\boldsymbol{x}_2) & \ldots & h_L(\boldsymbol{x}_2) \\ \vdots & \vdots & \vdots \\ h_1(\boldsymbol{x}_n) & \ldots & h_L(\boldsymbol{x}_n) \end{bmatrix}. \tag{3}$$

For completeness, we briefly introduce the multiclass classifiers of ELM. (Readers may refer to [27] for details.)

2.1 Multiclass ELM with single output

ELM can approximate any target continuous functions and the output of the ELM classifier $\boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta}$ can be as close to the class labels in the corresponding regions as possible. Thus, the classification problem for ELM with a single-output node can be formulated as [27]:

$$\begin{aligned} &\text{Minimize:} \quad L_{\text{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\varepsilon_i^2 \\ &\text{Subject to:} \quad \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} = t_i - \varepsilon_i, \quad i = 1, \ldots, n \end{aligned} \tag{4}$$

where t_i represents the expected output of single-output node, and ε_i is the training error of single-output node with respect to the training sample $\boldsymbol{x}_i$.

2.2 Multiclass ELM with multioutputs

If ELM has multioutput nodes, an m-class classifier is corresponding to m output nodes. If the original class label is l, the expected output vector of the m output nodes is $\boldsymbol{t}_i = [\overbrace{0, \ldots, 0, 1}^{l}, 0, \ldots, 0]^{\boldsymbol{T}}$. That is, the l-th element of $\boldsymbol{t}_i = [t_{i,1}, \ldots, t_{i,m}]^{\boldsymbol{T}}$ is one and the rest of the elements are zero. The classification problem for ELM with multioutput nodes is [27]

$$\begin{aligned} &\text{Minimize:} \quad L_{\text{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{n}\|\boldsymbol{\varepsilon}_i\|^2 \\ &\text{Subject to:} \quad \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta} = \boldsymbol{t}_i^{\boldsymbol{T}} - \varepsilon_i^{\boldsymbol{T}}, \quad i = 1, \ldots, n. \end{aligned} \tag{5}$$

where $\boldsymbol{\varepsilon}_i = [\varepsilon_{i1}, \ldots, \varepsilon_{im}]^{\boldsymbol{T}}$ is the training error vector of the m output nodes with respect to the training sample $\boldsymbol{x}_i$.

If a feature mapping $\boldsymbol{h}(\boldsymbol{x})$ is unknown to users, the output function of ELM classifier is

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\boldsymbol{T}}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{T} \\ &= [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{M}\right)^{-1}\boldsymbol{T}, \end{aligned} \tag{6}$$

where $\boldsymbol{M} = \boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}$, $m_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $k(\boldsymbol{x}, \boldsymbol{y})$ is a positive semi-definite kernel function. If a feature mapping $\boldsymbol{h}(\boldsymbol{x})$ is known, we have $\boldsymbol{h}(\boldsymbol{x}) = [G(\boldsymbol{a}_1, b_1, \boldsymbol{x}), \ldots, G(\boldsymbol{a}_L, b_L, \boldsymbol{x})]$, where $G(a, b, \boldsymbol{x})$ is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [8–10] and $\{(\boldsymbol{a}_i, b_i)\}_{i=1}^{L}$ are randomly generated according to any continuous probability distribution. The output function of ELM classifier is

$$f(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\boldsymbol{T}}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{T}, \tag{7}$$

or

$$f(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{T}, \tag{8}$$

where $\boldsymbol{T} = \begin{bmatrix} t_{11} & \ldots & t_{1m} \\ t_{21} & \ldots & t_{2m} \\ \vdots & \vdots & \vdots \\ t_{n1} & \ldots & t_{nm} \end{bmatrix}$.

1. Single-output node ($m = 1$): For multiclass problems, among all the multiclass labels, the predicted class label of a given testing sample is the closest to the output of ELM classifier. For the binary classification case, ELM only has one output node ($m = 1$), and the decision function of ELM classifier is

$$f(\boldsymbol{x}) = \text{sign}[k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_n)]\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{M}\right)^{-1}\boldsymbol{T}. \tag{9}$$

Equation (9) is available for the unknown feature mapping $\boldsymbol{h}(\boldsymbol{x})$. If $\boldsymbol{h}(\boldsymbol{x})$ is usually known, we have

$$f(\boldsymbol{x}) = \text{sign}\left\{\boldsymbol{h}(\boldsymbol{x})\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^{\boldsymbol{T}}\boldsymbol{T}\right\}. \tag{10}$$

Equation (10) can be applied to large-scale datasets or moderate datasets. The decision function applied to small-scale training samples is

$$f(\boldsymbol{x}) = \text{sign}\left\{\boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\boldsymbol{T}}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^{\boldsymbol{T}}\right)^{-1}\boldsymbol{T}\right\}. \tag{11}$$

2. Multioutput nodes ($m > 1$): For multiclass cases, the predicted class label of a given testing sample is the index number of the output node which has the highest output value. If $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^{\boldsymbol{T}}$, the predicted class label of sample $\boldsymbol{x}$ is

$$\text{label}(\boldsymbol{x}) = \underset{i \in \{1,2,\ldots,m\}}{\text{argmax}} \{f_i(\boldsymbol{x})\}. \tag{12}$$

3 Relation between manifold regularization framework and ELM

In order to avoid confusion, we list the main notations of this paper in Table 1. There is a probability distribution $\boldsymbol{P}$ on $\boldsymbol{X} \times \mathbb{R}$ and labeled examples are $(\boldsymbol{x}, \boldsymbol{y})$ pairs generated according to $\boldsymbol{P}$. Unlabeled examples are simply $\boldsymbol{x} \in \boldsymbol{X}$ drawn according to the marginal distribution $\boldsymbol{P}_{\boldsymbol{x}}$ of $\boldsymbol{P}$. Previous

Table 1 Notations

Notation	Explanation
$\mathbb{R}^{\mathrm{d}}$	The input d-dimensional Euclidean space
$\boldsymbol{X}$	$\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_l, \ldots, \boldsymbol{x}_{l+u}] \in \mathbb{R}^{d\times(l+u)}$ is the training data matrix. $\{\boldsymbol{x}_i\}_{i=1}^{l}$ are labeled points, and $\{\boldsymbol{x}_i\}_{i=l+1}^{l+u}$ are unlabeled points
m	The number of classes that the samples belong to
$\boldsymbol{Y}$	$\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l, 0, \ldots, 0) \in \mathbb{R}^{m\times(l+u)}$ is the 0–1 label matrix. $\boldsymbol{y}_i \in \mathbb{R}^{\mathrm{m}}$ is the label vector of $\boldsymbol{x}_i$, and all components of $\boldsymbol{y}_i$ are 0s except one being 1
$\boldsymbol{F}(\cdot)$	$\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$ is the discriminative vector function. The index of the class which $\boldsymbol{x}$ belongs to is that of the component with the maximum value
$k(\boldsymbol{x}, \boldsymbol{y})$	Kernel function of variables $\boldsymbol{x}$ and $\boldsymbol{y}$
$\boldsymbol{K}$	Kernel matrix $\boldsymbol{K} = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\} \in \mathbb{R}^{(l+u)\times(l+u)}$
$\boldsymbol{K}_l$	Kernel matrix $\boldsymbol{K}_l = \{k(\boldsymbol{x}_i, \boldsymbol{x}_j)\} \in \mathbb{R}^{l\times l}$
$\boldsymbol{B}$	$\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{l+u}) \in \mathbb{R}^{m\times(l+u)}$. Its columns are the coefficients of the kernel function to represent the discriminative function $\boldsymbol{F}(\cdot)$
$\|\|\|_{\mathcal{H}}$	Norm in the Hilbert space $\mathcal{H}_K$
$\langle\rangle_K$	Inner product in the Hilbert space $\mathcal{H}_K$
$tr(\boldsymbol{M})$	The trace of the matrix M, that is, the sum of the diagonal elements of the matrix $\boldsymbol{M}$

Table 2 Description of manifold regularized ELM algorithm

Manifold regularized ELM algorithm

Input: l labeled examples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{l}$, u unlabeled examples $\{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$

Output: Estimated function $\boldsymbol{F}^*(\boldsymbol{x}) = \left(f_1^*(\boldsymbol{x}), \ldots, f_2^*(\boldsymbol{x}), \ldots, f_m^*(\boldsymbol{x})\right)^{\mathrm{T}}$

Step 1: Construct data adjacency graph with $(l + u)$ nodes using k-nearest neighbors or a graph kernel. Choose edge weights $\boldsymbol{w}_{ij}$, for example, binary weights or heat kernel weights $\boldsymbol{w}_{ij} = e^{\left\|-\boldsymbol{x}_i-\boldsymbol{x}_j\right\|^2/4t}$

Step 2: Compute graph Laplacian matrix:$L_{\mathrm{G}} = \boldsymbol{D} - \boldsymbol{W}$ where $\boldsymbol{D}$ is a diagonal matrix given by $\boldsymbol{D}_{ii} = \sum_{j=1}^{l+u} \boldsymbol{w}_{ij}$

Step 3: Choose a kernel function $k(\boldsymbol{x}, \boldsymbol{y})$. Choose γ_I, C and L (the number of sample points), randomly generate$\{(a_i, b_i)\}_{i=1}^{L}$ and compute $\boldsymbol{H} = (\emptyset(\boldsymbol{x}_1), \emptyset(\boldsymbol{x}_2), \ldots, \emptyset(\boldsymbol{x}_{l+u}))^{\mathrm{T}}$, where $\emptyset(\boldsymbol{x}) = \frac{1}{\sqrt{L}}(k_{b_1}(\boldsymbol{x}, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}, \boldsymbol{a}_L))$

Step 4: If the number of the training datasets is very large $(l + u) \gg L$, select (35) for MR-ELM with single output $(m = 1)$ or select (33) for MR-ELM with multioutputs, otherwise, select (36) or (34), respectively

Step 5: Output function $\boldsymbol{F}^*(\boldsymbol{x}) = \left(f_1^*(\boldsymbol{x}), \ldots, f_2^*(\boldsymbol{x}), \ldots, f_m^*(\boldsymbol{x})\right)^{\mathrm{T}}$

studies have shown that there may be some connection between the conditional and marginal distributions. Thus, the knowledge of the marginal distribution $\boldsymbol{P}_{\boldsymbol{x}}$ can be exploited for better function learning. Specifically, if two points $\boldsymbol{x}_1$, $\boldsymbol{x}_2 \in \boldsymbol{X}$ are close in the intrinsic geometry of $\boldsymbol{P}_{\boldsymbol{x}}$, then the conditional probabilities $\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x}_1)$ and $\boldsymbol{P}(\boldsymbol{y}|\boldsymbol{x}_2)$ are similar, where $\boldsymbol{y} \in \{1, \ldots, m\}$ is the class label. Thus, the conditional probability distribution varies smoothly along the geodesics in the intrinsic geometry of $\boldsymbol{P}_{\boldsymbol{x}}$. We intend to utilize these geometric intuitions to extend the original ELM for learning a semi-supervised ELM model.

For a Mercer kernel K: $\boldsymbol{X} \times \boldsymbol{X} \to \mathbb{R}$, there is an associated RKHS $\mathcal{H}_K$ of functions $\boldsymbol{X} \to \mathbb{R}$ with the corresponding norm $\|\|\|_{\mathcal{H}}$. Given a set of l labeled examples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{l}$ and a set of u unlabeled examples $\{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, the manifold regularization framework estimates an unknown function by minimizing [22]

$$\boldsymbol{f}^* = \underset{\boldsymbol{f}\in\mathcal{H}}{\operatorname{argmin}}\left[\frac{1}{l}\sum_{i=1}^{l} V(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{f}) + \gamma_A\|\boldsymbol{f}\|_{\mathcal{H}}^2 + \gamma_I\|\boldsymbol{f}\|_{\boldsymbol{I}}^2\right] \tag{13}$$

where V is some loss function, such as the squared loss $(\boldsymbol{y}_i - f(\boldsymbol{x}_i))^2$ for RLS and the hinge loss function $\max[0, 1 - \boldsymbol{y}_i f(\boldsymbol{x}_i)]$ for SVM, $\|\boldsymbol{f}\|_{\mathcal{H}}^2$ is the RKHS norm penalty and represents the complexity of functions in RKHS H_K and $\|\boldsymbol{f}\|_I^2$ is a smoothness penalty corresponding to the sample probability distribution. γ_A controls the complexity of functions in the ambient space, while γ_I controls the complexity of functions in the intrinsic geometry of sample probability distribution.

When we consider the case that the support of $\boldsymbol{P}_{\boldsymbol{x}}$ is a compact submanifold $\mathcal{M} \subset \mathbb{R}^{\mathrm{d}}$, a natural choice for $\|\boldsymbol{f}\|_I$ is $\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}}\boldsymbol{f}\|^2 \mathrm{d}\boldsymbol{P}_{\boldsymbol{x}}(\boldsymbol{x})$ [22], where $\nabla_{\mathcal{M}}$ is the gradient of $\boldsymbol{f}$ along the manifold, and the integral is taken over the distribution $\boldsymbol{P}_{\boldsymbol{x}}$. In most applications, the marginal $\boldsymbol{P}_{\boldsymbol{x}}$ is not known. Therefore, we must attempt to get empirical estimates of $\boldsymbol{P}_{\boldsymbol{x}}$ and $\|\cdot\|_I$. In order to model the geometrical structure of $\mathcal{M}$, we construct a nearest-neighbor graph $\boldsymbol{G}$ and define the weight matrix $\boldsymbol{W}$ on the graph. Define $L_{\mathrm{G}} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is a diagonal matrix whose entries are column (or row) sums of $\boldsymbol{W}$, that is $\boldsymbol{D}_{ii} = \sum_{j=1}^{l+u} \boldsymbol{w}_{ij}$. L_{G} is called graph Laplacian [28]. By spectral graph theory, $\|\boldsymbol{f}\|_I^2$ can be discretely approximated as follows:

$$\begin{aligned}\|\boldsymbol{f}\|_I^2 &= \frac{1}{2(u+l)^2}\sum_{i,j=1}^{l+u}\left(f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)\right)^2\boldsymbol{w}_{ij}\\ &= \frac{1}{(u+l)^2}\left(\sum_{i=1}^{l+u} f(\boldsymbol{x}_i)^2\boldsymbol{D}_{ii} - \sum_{i=1,j=1}^{l+u} f(\boldsymbol{x}_i)f(\boldsymbol{x}_j)\boldsymbol{w}_{ij}\right)\\ &= \frac{1}{(u+l)^2}\left(\boldsymbol{f}^T\boldsymbol{D}\boldsymbol{f} - \boldsymbol{f}^T\boldsymbol{W}\boldsymbol{f}\right) = \frac{1}{(u+l)^2}\boldsymbol{f}^T L_{\mathrm{G}}\boldsymbol{f},\end{aligned} \tag{14}$$

where the normalizing coefficient $1/(l+u)^2$ is the natural scale factor for the empirical estimate of the Laplace operator.

The following theorem shows that the solution of optimization problem (13) has an expression in terms of both labeled and unlabeled examples, which is important to our algorithms.

Theorem 1 ([22]) *The minimizer of optimization problem* (13) *admits an expansion*

$$f^*(\boldsymbol{x}) = \sum_{i=1}^{l+u} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) \tag{15}$$

in terms of the labeled and unlabeled examples. □

In the MR framework, if $\boldsymbol{y}_i$ is an m-dimensional label vector with the elements 0 or 1, where m is the number of classes, and $\boldsymbol{x}_i$ belongs to the kth class, then the kth component of $\boldsymbol{y}_i$ takes the value 1 and the rest components the value 0. The corresponding vector function is defined as $\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T$. The extended manifold regularization framework (E-MR) estimates an unknown vector function by minimizing

$$\begin{aligned} F^* &= \underset{\boldsymbol{F}\in\boldsymbol{S}}{\operatorname{argmin}} \left[\frac{1}{l}\sum_{i=1}^{l} V(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{F}) + \gamma_A \|\boldsymbol{F}\|_{\mathcal{H}}^2 + \gamma_I \|\boldsymbol{F}\|_I^2 \right] \\ &= \underset{\boldsymbol{F}\in\boldsymbol{S}}{\operatorname{argmin}} \left[\frac{1}{l}\sum_{i=1}^{l} V(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{F}) + \gamma_A \|\boldsymbol{F}\|_{\mathcal{H}}^2 + \frac{\gamma_I}{(u+l)^2} \sum_{i,j=1}^{l+u} \left\|\boldsymbol{F}(\boldsymbol{x}_i) - \boldsymbol{F}(\boldsymbol{x}_j)\right\|^2 \boldsymbol{w}_{ij} \right] \\ &= \underset{\boldsymbol{F}\in\boldsymbol{S}}{\operatorname{argmin}} \left[\frac{1}{l}\sum_{i=1}^{l} V(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{F}) + \gamma_A \|\boldsymbol{F}\|_{\mathcal{H}}^2 + \frac{\gamma_I}{(u+l)^2} tr(\boldsymbol{F}\boldsymbol{L}_{\mathrm{G}}\boldsymbol{F}^T) \right] \end{aligned} \tag{16}$$

where

$\boldsymbol{S} = \left\{ (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))^T | f_i(\boldsymbol{x}) \in \text{RKHS } \mathcal{H}_K, 1 \le i \le m \right\}$,

$\boldsymbol{F} = (\boldsymbol{F}(\boldsymbol{x}_1), \ldots, \boldsymbol{F}(\boldsymbol{x}_{l+u})) \in \mathbb{R}^{m\times(l+u)}$.

Theorem 2 *The minimizer of optimization problem* (16) *admits an expansion*

$$\boldsymbol{F}^*(\boldsymbol{x}) = \sum_{i=1}^{l+u} \boldsymbol{\beta}_i k(\boldsymbol{x}_i, \boldsymbol{x}) \tag{17}$$

in terms of the labeled and unlabeled examples, where $\boldsymbol{\beta}_i = (\boldsymbol{\beta}_{1i}, \boldsymbol{\beta}_{2i}, \ldots, \boldsymbol{\beta}_{mi})^{\mathrm{T}}$. □

The proof of Theorem 2 is similar to that of Theorem 2 in [22], and we need not discuss it in details. In addition, ELM can be derived from the E-MR or MR framework. In order to demonstrate the relationship between them, we firstly present Theorem 3 and derive ELM based on kernel functions from the MR framework.

Theorem 3 *Defining a mapping from a Euclidean space* $\mathbb{R}^{\mathrm{d}}$ *to a RKHS* $\mathcal{H}_K$ *corresponding to a Mercer kernel function* $k(\boldsymbol{x}, \boldsymbol{x}')$:

$\emptyset : \boldsymbol{X} \to k(\boldsymbol{x}, \cdot)$,

where $\boldsymbol{X} \in \mathbb{R}^{\mathrm{d}}$ *and* $k(\boldsymbol{x}, \cdot) \in \mathcal{H}_K$. *If the mapping* $\boldsymbol{h}(\boldsymbol{x})$ *in ELM is* $k(\boldsymbol{x}, \cdot)$, *loss function of the E-MR framework is the square-loss function,* $l = 2/C$, $\gamma_A = 1/2$, $\gamma_I = 0$, *and* $u = 0$, *then the E-MR framework degenerates into ELM based on kernel functions.* □

Proof Let $\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{l+u}) \in \mathbb{R}^{m\times(l+u)}$ be the matrix of coefficients $\boldsymbol{\beta}_i$, $\boldsymbol{b}_j = (\boldsymbol{b}_{j1}, \boldsymbol{b}_{j2}, \ldots, \boldsymbol{b}_{j(l+u)})$ be the row vector of matrix $\boldsymbol{B}$, and $\boldsymbol{K} = (k(\boldsymbol{x}_i, \boldsymbol{x}_j)) \in \mathbb{R}^{(l+u)\times(l+u)}$ be the kernel matrix over labeled and unlabeled points. Then, we have

$$\|\boldsymbol{F}\|_{\mathcal{H}}^2 = \sum_{i=1}^{m} \|\boldsymbol{f}_i\|_{\mathcal{H}}^2 = \sum_{i=1}^{m} \boldsymbol{b}_i \boldsymbol{K} \boldsymbol{b}_i^{\boldsymbol{T}} = tr(\boldsymbol{B}\boldsymbol{K}\boldsymbol{B}^T). \tag{18}$$

Let $\boldsymbol{F} = \boldsymbol{B}\boldsymbol{K}$. Then

$$\begin{aligned} \|\boldsymbol{F}\|_I^2 &= \frac{1}{(l+u)^2} \sum_{i,j=1}^{l+u} \left\|\boldsymbol{F}(\boldsymbol{x}_i) - \boldsymbol{F}(\boldsymbol{x}_j)\right\|^2 \boldsymbol{w}_{ij} \\ &= \frac{1}{(l+u)^2} tr(\boldsymbol{F}\boldsymbol{L}_{\mathrm{G}}\boldsymbol{F}^T) = \frac{1}{(l+u)^2} tr(\boldsymbol{B}\boldsymbol{K}\boldsymbol{L}_{\mathrm{G}}\boldsymbol{K}\boldsymbol{B}^T). \end{aligned} \tag{19}$$

Assume that $Y = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l, \boldsymbol{0}, \ldots, \boldsymbol{0})) \in \mathbb{R}^{m\times(l+u)}$ is the label matrix with elements 0 or 1, where $\boldsymbol{0} \in \mathbb{R}^m$ is the zero vector, and $\boldsymbol{y}_i$ $(1 \le i \le l)$ is an m-dimensional label vector with the elements 0 or 1, and $\boldsymbol{J} \in \mathbb{R}^{(l+u)\times(l+u)}$ is a diagonal matrix with the first l diagonal elements being 1 and the rest being 0. Substituting (18) and (19) into (16) and choosing loss functions of the E-MR framework to be the square-loss function, we have the following convex optimization problem:

$$\begin{aligned} \boldsymbol{B}^* = \underset{\boldsymbol{B}\in\mathbb{R}^{m\times(l+u)}}{\operatorname{argmin}} \Bigg\{ &\frac{1}{l} tr\left((\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{J})(\boldsymbol{Y} - \boldsymbol{B}\boldsymbol{K}\boldsymbol{J})^{\mathrm{T}} \right) + \gamma_A tr\left(\boldsymbol{B}\boldsymbol{K}\boldsymbol{B}^T\right) \\ &+ \frac{\gamma_I}{(l+u)^2} tr(\boldsymbol{B}\boldsymbol{K}\boldsymbol{L}_{\mathrm{G}}\boldsymbol{K}\boldsymbol{B}^T) \Bigg\}. \end{aligned} \tag{20}$$

Similar to the result of the sparse regularization optimization problem [24], the solution of the optimization problem (20) is given by

$$\boldsymbol{B}^* = \boldsymbol{Y}\left(\boldsymbol{K}\boldsymbol{J} + \gamma_A l \mathbf{I} + \frac{\gamma_I l}{(l+u)^2} \boldsymbol{K}\boldsymbol{L}_{\mathrm{G}} \right)^{-1}, \tag{21}$$

where $\mathbf{I}$ is the identity matrix. Then, the solution of the problem (20) is

$$\boldsymbol{F}^*(\boldsymbol{x}) = (f_1^*(\boldsymbol{x}), \ldots, f_2^*(\boldsymbol{x}), \ldots, f_m^*(\boldsymbol{x}))^{\mathrm{T}} = \sum_{i=1}^{l+u} \boldsymbol{\beta}_i^* k(\boldsymbol{x}_i, \boldsymbol{x})$$

with $\boldsymbol{B}^* = (\boldsymbol{\beta}_1^*, \ldots, \boldsymbol{\beta}_{l+u}^*)$. Thus, the multiclass classifier based on E-MR is obtained, that is, $g^*(\boldsymbol{x}) = i^* = \operatorname{argmax}_{i\in\{1,2,\ldots,m\}} \{f_i(\boldsymbol{x})\}$. If unlabeled points are not taken into account, i.e., $u = 0$, then $\boldsymbol{J} \in \mathbb{R}^{l\times l}$ is a diagonal matrix with diagonal elements being 1 and $\gamma_I = 0$. Let

$\boldsymbol{K}_l = (k(\boldsymbol{x}_{ii}, \boldsymbol{x}_j)) \in \mathbb{R}^{l\times l}$ be the kernel matrix over labeled points. Thus, (21) can be transformed into $\boldsymbol{B}^* = \boldsymbol{Y}(\boldsymbol{K}_l + \gamma_A l\mathbf{I})^{-1}$. Let $l = 2/C$, $\gamma_A = 1/2$. Furthermore, we have $\boldsymbol{B}^* = \boldsymbol{Y}\left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}$. Obviously, $\left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}$ is a symmetric matrix and $\boldsymbol{B}^{*\mathrm{T}} = \left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}$. Thus, we have

$$\begin{aligned}\boldsymbol{F}^*(\boldsymbol{x})^T &= [k(\boldsymbol{x}_1,\boldsymbol{x}), k(\boldsymbol{x}_2,\boldsymbol{x}), \ldots k(\boldsymbol{x}_l,\boldsymbol{x})]\boldsymbol{B}^{*\mathrm{T}} \\ &= [k(\boldsymbol{x}_1,\boldsymbol{x}), k(\boldsymbol{x}_2,\boldsymbol{x}), \ldots k(\boldsymbol{x}_l,\boldsymbol{x})]\left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}.\end{aligned} \tag{22}$$

It is clear that (22) is the output function of ELM based on kernel functions in [27]. When $m = 1$, the E-MR framework is simplified into the MR framework. The form of (22) is

$$\boldsymbol{f}^*(\boldsymbol{x}) = [k(\boldsymbol{x}_1,\boldsymbol{x}), k(\boldsymbol{x}_2,\boldsymbol{x}), \ldots k(\boldsymbol{x}_l,\boldsymbol{x})]\left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}\boldsymbol{y}^{\mathrm{T}}, \tag{23}$$

where $\boldsymbol{y} = (y_1, \ldots, y_l)$ is the label vector with elements 0 or 1, $\boldsymbol{y}_i$ $(1 \le i \le l)$ is the sample label, which is consistent with that of ELM. This completes the proof of Theorem 3. □

In the process of proving Theorem 3, it can be seen from (21) that our algorithm needs to compute the inverse of a $(l+u)\times(l+u)$ matrix if the manifold regularization term is added. This may be impractical for large datasets. In order to maintain the properties of ELM, it is necessary to solve this problem by obtaining approximate solutions. The solution to the problem can be simplified by relaxing the form of (17). Thus, we have

$$\begin{aligned}\boldsymbol{F}^*(\boldsymbol{x}) &= \sum_{i=1}^{l+u}\boldsymbol{\beta}_i k(\boldsymbol{x}_i,\boldsymbol{x}) = \boldsymbol{B}[k(\boldsymbol{x}_1,\boldsymbol{x}), k(\boldsymbol{x}_2,\boldsymbol{x}), \ldots, k(\boldsymbol{x}_{l+u},\boldsymbol{x})]^{\mathrm{T}} \\ &= \boldsymbol{B}\left[\left\langle k(\boldsymbol{x}_1,\cdot), k(\boldsymbol{x},\cdot)\right\rangle_K, k(\boldsymbol{x}_2,\cdot), k(\boldsymbol{x},\cdot)\right\rangle_K, \ldots, k(\boldsymbol{x}_{l+u},\cdot), k(\boldsymbol{x},\cdot)\rangle_K\Big]^{\mathrm{T}}.\end{aligned} \tag{24}$$

Thus, a set of functions $k(\boldsymbol{x}_i,\cdot)(1 \le i \le l+u)$ and $k(\boldsymbol{x},\cdot)$ including parameters of functions can be discretized. Theorem 3 shows that ELM based on kernel functions can be derived from the E-MR framework. We further prove two more theorems as shown below by relaxing the form of solutions. Theorem 4 demonstrates the relationship between the E-MR framework and ELM with multioutputs, while Theorem 5 demonstrates the relationship between the MR framework and ELM with single output.

Theorem 4 *In the E-MR framework, if (17) is discretized by using the following formula* $\boldsymbol{F}^*(\boldsymbol{x}) = \boldsymbol{B}\left[k(\boldsymbol{x}_1,\cdot), k(\boldsymbol{x},\cdot)_K, \ldots, k(\boldsymbol{x}_{l+u},\cdot), k(\boldsymbol{x},\cdot)_K\right]^{\mathrm{T}} \approx \boldsymbol{B}\boldsymbol{H}\emptyset^T(\boldsymbol{x})$, *where* $\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{l+u}) \in \mathbb{R}^{m\times(l+u)}$ *is the matrix of coefficients* $\boldsymbol{\beta}_i$,

$$\boldsymbol{H} = \frac{1}{\sqrt{L}}\begin{bmatrix} k_{b_1}(\boldsymbol{x}_1,\boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_1,\boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_1,\boldsymbol{a}_L) \\ k_{b_1}(\boldsymbol{x}_2,\boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_2,\boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_2,\boldsymbol{a}_L) \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ k_{b_1}(\boldsymbol{x}_{l+u},\boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_{l+u},\boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_{l+u},\boldsymbol{a}_L)\end{bmatrix},$$

$\emptyset(\boldsymbol{x}) = \frac{1}{\sqrt{L}}(k_{b_1}(\boldsymbol{x},\boldsymbol{a}_1), k_{b_2}(\boldsymbol{x},\boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x},\boldsymbol{a}_L))$, $\{(\boldsymbol{a}_i, b_i)\}_{i=1}^{L}$ (b_i *is the parameter of kernel function*) *are random sampling point from any continuous probability distribution and L is the number of sampling points. Let the mapping* $\boldsymbol{h}(\boldsymbol{x})$ *in ELM be* $\emptyset(\boldsymbol{x})$, $l = 2/C$, $\gamma_A = 1/2$, $\gamma_I = 0$ *and* $u = 0$. *Then, the E-MR framework degenerates into ELM with multi-outputs in the case of the square-loss function.* □

Proof From the proof of Theorem 3, we get $\boldsymbol{B}^{*\mathrm{T}} = \left(\boldsymbol{K}_l + \frac{\mathbf{I}}{C}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}$.If $\boldsymbol{H}_l = (\emptyset(\boldsymbol{x}_1), \emptyset(\boldsymbol{x}_2), \ldots, \emptyset(\boldsymbol{x}_l)))^{\mathrm{T}}$, then $\boldsymbol{K}_l \approx \boldsymbol{H}_l\boldsymbol{H}_l^T$ and $\boldsymbol{B}^{*\mathrm{T}} = \left(\boldsymbol{H}_l\boldsymbol{H}_l^T + \frac{\mathbf{I}}{C}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}$. If the number of training samples is not huge, the output function is

$$\boldsymbol{F}^*(\boldsymbol{x})^{\mathrm{T}} = \emptyset(\boldsymbol{x})\boldsymbol{H}_l^T\boldsymbol{B}^{*\mathrm{T}} = \emptyset(\boldsymbol{x})\boldsymbol{H}_l^T\left(\boldsymbol{H}_l\boldsymbol{H}_l^T + \frac{I}{C}\right)^{-1}\boldsymbol{Y}^{\mathrm{T}}. \tag{25}$$

If the number of training samples is huge, according to the Sherman–Morrison–Woodbury (SMW) formula for matrix inversion, we have

$$\begin{aligned}\boldsymbol{H}_l^T\left(\boldsymbol{H}_l\boldsymbol{H}_l^T + \frac{\mathbf{I}}{C}\right)^{-1} &= C\left[\mathbf{I} - \boldsymbol{H}_l\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}_l^T\boldsymbol{H}_l\right)^{-1}\boldsymbol{H}_l^T\right] \\ &= C\cdot\frac{1}{C}\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}_l^T\boldsymbol{H}_l\right)^{-1}\boldsymbol{H}_l^T \\ &= \left(\frac{\mathbf{I}}{C} + \boldsymbol{H}_l^T\boldsymbol{H}_l\right)^{-1}\boldsymbol{H}_l^T,\end{aligned} \tag{26}$$

$$\boldsymbol{F}^*(\boldsymbol{x})^{\mathrm{T}} = \emptyset(\boldsymbol{x})\boldsymbol{H}_l^T\boldsymbol{B}^{*\mathrm{T}}. \tag{27}$$

Substituting (26) into (27), we obtain

$$\boldsymbol{F}^*(\boldsymbol{x})^{\mathrm{T}} = \emptyset(\boldsymbol{x})\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}_l^T\boldsymbol{H}_l\right)^{-1}\boldsymbol{H}_l^T\boldsymbol{Y}^{\mathrm{T}} \tag{28}$$

Let the mapping $\boldsymbol{h}(\boldsymbol{x})$ in ELM be $\emptyset(\boldsymbol{x})$. We get the same output functions (25) and (28) as that of ELM. This completes the proof of Theorem 4. □

Theorem 5 *In the MR framework, if (15) is discretized by using the following formula* $f^*(\boldsymbol{x}) = \boldsymbol{\alpha}\left[\left\langle k(\boldsymbol{x}_1,\cdot), k(\boldsymbol{x},\cdot)\right\rangle_K, \ldots, k(\boldsymbol{x}_{l+u},\cdot), k(\boldsymbol{x},\cdot)\right\rangle_K]^{\mathrm{T}} \approx \boldsymbol{\alpha}\boldsymbol{H}\emptyset^T(\boldsymbol{x})$, *where* $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{l+u}) \in \mathbb{R}^{l+u}$ *is the* $(l+u)$*-dimensional vector, and*

$$\boldsymbol{H} = \frac{1}{\sqrt{L}} \begin{bmatrix} k_{b_1}(\boldsymbol{x}_1, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_1, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_1, \boldsymbol{a}_L) \\ k_{b_1}(\boldsymbol{x}_2, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_2, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_2, \boldsymbol{a}_L) \\ \vdots \quad\quad\quad \vdots \quad\quad\quad \vdots \\ k_{b_1}(\boldsymbol{x}_{l+u}, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}_{l+u}, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}_{l+u}, \boldsymbol{a}_L) \end{bmatrix},$$

$\emptyset(\boldsymbol{x}) = \frac{1}{\sqrt{L}}(k_{b_1}(\boldsymbol{x}, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}, \boldsymbol{a}_L))$, $\{(\boldsymbol{a}_i,\ b_i)\}_{i=1}^{L}$ (b_i *is the parameter of kernel function*) *are random sampling point from any continuous probability distribution and L is the number of sampling points. If the mapping* $\boldsymbol{h}(\boldsymbol{x})$ *in ELM is* $\emptyset(\boldsymbol{x})$, $l = 2/C$, $\gamma_A = 1/2$, $\gamma_I = 0$ *and* $u = 0$, *then the MR framework degenerates into ELM with single output in the case of the square-loss function.* □

The proof is similar to that of Theorem 4, and we do not discuss it further.

From Theorems 4 and 5, it can be seen that the number of sampling points L is corresponding to the number of hidden nodes in ELM. We can come to the conclusion that if the hidden-layer output function $\boldsymbol{h}(\boldsymbol{x})$ in ELM is $\emptyset(\boldsymbol{x})$ defined above, the essence of random feature mapping in ELM is actually discretizing the function $k(\boldsymbol{x}, \cdot)$ of a RKHS $\mathcal{H}_K$ randomly. According to the interpolation theorem and universal approximation theorem in [7, 29], the generalization performance of ELM is not sensitive to the dimensionality of the feature space (L) and good performance can be reached as long as L is large enough.

4 Extreme learning machine with manifold regularization

On the basis of theoretical analysis mentioned in Sect. 3, we can construct MR-ELM with multioutputs from the E-MR framework and the output function of MR-ELM classifier can be derived. MR-ELM with single output, as a specific case of multioutput nodes, is also obtained. The classification problem for MR-ELM with multioutput nodes can be formulated as

$$F^* = \underset{\boldsymbol{F} \in \boldsymbol{S}}{\operatorname{argmin}} \left[\frac{C}{2} \sum_{i=1}^{l} (\boldsymbol{y}_i - \boldsymbol{F}(\boldsymbol{x}_i))^2 + \frac{1}{2} \|\boldsymbol{F}\|_{\mathcal{H}}^2 + \frac{\gamma_I}{(u+l)^2} tr(\boldsymbol{F} L_{\mathrm{G}} \boldsymbol{F}^T) \right]. \tag{29}$$

Substituting (18) and (19) into (29), we have the following convex optimization problem:

$$\boldsymbol{B}^* = \underset{\mathrm{B} \in \mathbb{R}^{m \times (l+u)}}{\operatorname{argmin}} \left\{ \frac{C}{2} tr\left((\boldsymbol{Y} - \boldsymbol{BKJ})(\boldsymbol{Y} - \boldsymbol{BKJ})^{\mathrm{T}} \right) + \frac{1}{2} tr(\boldsymbol{BKB}^T) + \frac{\gamma_I}{(l+u)^2} tr(\boldsymbol{BK} L_{\mathrm{G}} \boldsymbol{KB}^T) \right\}. \tag{30}$$

The solution of the optimization problem (30) is given by

$$\boldsymbol{B}^* = \boldsymbol{Y} \left(\frac{\mathbf{I}}{C} + \boldsymbol{KJ} + \frac{2\gamma_I}{(l+u)^2 C} \boldsymbol{K} L_{\mathrm{G}} \right)^{-1}, \tag{31}$$

where $\mathbf{I}$ is the identity matrix. Thus, we have the following function

$$\begin{aligned} \boldsymbol{F}^*(\boldsymbol{x}) &= \sum_{i=1}^{l+u} \boldsymbol{\beta}_i^* k(\boldsymbol{x}_i, \boldsymbol{x}) \approx \boldsymbol{B}^* \boldsymbol{H} \emptyset^T(\boldsymbol{x}) \\ &= \boldsymbol{Y} \left(\frac{\mathbf{I}}{C} + \boldsymbol{KJ} + \frac{2\gamma_I}{(l+u)^2 \mathrm{C}} \boldsymbol{K} L_{\mathrm{G}} \right)^{-1} \boldsymbol{H} \emptyset^T(\boldsymbol{x}), \end{aligned} \tag{32}$$

where $\boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l, 0, \ldots, 0)) \in \mathbb{R}^{m \times (l+u)}$ is the label matrix with elements 0 or 1, $\mathbf{0} \in \mathbb{R}^m$ is the zero vector, and $\boldsymbol{y}_i$ $(1 \le i \le l)$ is an m-dimensional label vector with the elements 0 or 1, $\boldsymbol{J} \in \mathbb{R}^{(l+u) \times (l+u)}$ is a diagonal matrix with the first l diagonal elements being 1 and the rest being 0, $\boldsymbol{K} = \left(k(\boldsymbol{x}_i, \boldsymbol{x}_j)\right) \in \mathbb{R}^{(l+u) \times (l+u)}$ is the kernel matrix over labeled and unlabeled points, $\boldsymbol{H} = (\emptyset(\boldsymbol{x}_1), \emptyset(\boldsymbol{x}_2), \ldots \emptyset(\boldsymbol{x}_{l+u}))^{\mathrm{T}}$, $\emptyset(\boldsymbol{x}) = \frac{1}{\sqrt{L}}(k_{b_1}(\boldsymbol{x}, \boldsymbol{a}_1), k_{b_2}(\boldsymbol{x}, \boldsymbol{a}_2), \ldots, k_{b_L}(\boldsymbol{x}, \boldsymbol{a}_L))$, $\boldsymbol{K} \approx \boldsymbol{HH}^T$. Further, we have

$$\boldsymbol{F}^*(\boldsymbol{x}) \approx \boldsymbol{Y} \left(\frac{\mathbf{I}}{C} + \boldsymbol{HH}^T \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C} \boldsymbol{HH}^T L_{\mathrm{G}} \right)^{-1} \boldsymbol{H} \emptyset^T(\boldsymbol{x}). \tag{33}$$

It can be seen that the output function (33) of MR-ELM still entails the inversion of a possibly massive matrix of order $(l+u) \times (l+u)$. Thus, we make use of the Sherman–Morrison–Woodbury (SMW) formula for matrix inversion which results in the following formula:

$$\begin{aligned} \boldsymbol{F}^*(\boldsymbol{x}) &\approx \boldsymbol{Y} \left(\frac{\mathbf{I}}{C} + \boldsymbol{HH}^T \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C} \boldsymbol{HH}^T L_{\mathrm{G}} \right)^{-1} \boldsymbol{H} \emptyset^T(\boldsymbol{x}) \\ &= C\boldsymbol{Y} \left(\mathbf{I} - \boldsymbol{H} \left(\frac{\mathbf{I}}{C} + \boldsymbol{H}^T \boldsymbol{QH} \right)^{-1} \boldsymbol{H}^T \boldsymbol{Q} \right) \boldsymbol{H} \emptyset^T(\boldsymbol{x}), \end{aligned} \tag{34}$$

where $\boldsymbol{Q} = \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C} L_{\mathrm{G}}$. Correspondingly, if the number of training samples is not huge, the output function of MR-ELM with single output is

$$f^*(\boldsymbol{x}) \approx \boldsymbol{Y}' \left(\frac{\mathbf{I}}{C} + \boldsymbol{HH}^T \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C} \boldsymbol{HH}^T L_{\mathrm{G}} \right)^{-1} \boldsymbol{H} \emptyset^T(\boldsymbol{x}). \tag{35}$$

Otherwise,

$$f^*(\boldsymbol{x}) \approx C\boldsymbol{Y}' \left(\mathbf{I} - \boldsymbol{H} \left(\frac{\mathbf{I}}{C} + \boldsymbol{H}^T \boldsymbol{QH} \right)^{-1} \boldsymbol{H}^T \boldsymbol{Q} \right) \boldsymbol{H} \emptyset^T(\boldsymbol{x}), \tag{36}$$

where $\boldsymbol{Y}'$ is an $(l+u)$ dimensional label vector given by: $\boldsymbol{Y}' = (y_1, \ldots, y_l, 0, \ldots, 0)$.

The formula (34) and (36) involve the inversion of a matrix of order $L \times L$. As long as L is large enough, the generalization performance of MR-ELM is not sensitive to the dimensionality of the feature space (L) and good performance can be reached, which will be verified later in Sect. 5. The manifold regularized ELM algorithm is summarized in Table 2.

Similar to ELM, MR-ELM has the unified solutions for regression, binary and multiclass classification. We will use the classification as an example to demonstrate the performance of MR-ELM.

1. MR-ELM classifier with single-output node ($m = 1$): For multiclass problems, among all the multiclass labels, the predicted class label of a given testing sample is the closest to the output of MR-ELM classifier. The decision function of MR-ELM classifier applicable to moderate-scale training samples is

$$f^*(\boldsymbol{x}) \approx \boldsymbol{Y}'\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}\boldsymbol{H}^T\boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C}\boldsymbol{H}\boldsymbol{H}^T L_{\mathrm{G}}\right)^{-1}\boldsymbol{H}\emptyset^T(\boldsymbol{x}), \tag{37}$$

or the one applicable to large-scale training samples is

$$f^*(\boldsymbol{x}) \approx C\boldsymbol{Y}'\left(\mathbf{I} - \boldsymbol{H}\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}^T\boldsymbol{Q}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Q}\right)\boldsymbol{H}\emptyset^T(\boldsymbol{x}). \tag{38}$$

Correspondingly, for the binary classification case, the decision function of MR-ELM classifier is

$$f^*(\boldsymbol{x}) \approx \mathrm{sign}\left(\boldsymbol{Y}'\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}\boldsymbol{H}^T\boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C}\boldsymbol{H}\boldsymbol{H}^T L_{\mathrm{G}}\right)^{-1}\boldsymbol{H}\emptyset^T(\boldsymbol{x})\right), \tag{39}$$

or

$$f^*(\boldsymbol{x}) \approx \mathrm{sign}\left(C\boldsymbol{Y}'\left(\mathbf{I} - \boldsymbol{H}\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}^T\boldsymbol{Q}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Q}\right)\boldsymbol{H}\emptyset^T(\boldsymbol{x})\right), \tag{40}$$

where $\boldsymbol{Q} = \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C}L_{\mathrm{G}}$.

2. MR-ELM classifier with multioutput nodes ($m > 1$): For multiclass cases, the predicted class label of a given testing sample is the index number of the output node which has the highest output value for the given testing sample. The decision function of MR-ELM classifier applicable to moderate-scale training samples is

$$\begin{aligned}\boldsymbol{F}^*(\boldsymbol{x}) &= \left(f_1^*(\boldsymbol{x}), \ldots, f_2^*(\boldsymbol{x}), \ldots, f_m^*(\boldsymbol{x})\right)^{\mathrm{T}} \\ &\approx \boldsymbol{Y}\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}\boldsymbol{H}^T\boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C}\boldsymbol{H}\boldsymbol{H}^T L_{\mathrm{G}}\right)^{-1}\boldsymbol{H}\emptyset^T(\boldsymbol{x}),\end{aligned} \tag{41}$$

or the one applicable to large-scale training samples is

$$\begin{aligned}\boldsymbol{F}^*(\boldsymbol{x}) &= \left(f_1^*(\boldsymbol{x}), \ldots, f_2^*(\boldsymbol{x}), \ldots, f_m^*(\boldsymbol{x})\right)^{\mathrm{T}} \\ &\approx C\boldsymbol{Y}\left(\mathbf{I} - \boldsymbol{H}\left(\frac{\mathbf{I}}{C} + \boldsymbol{H}^T\boldsymbol{Q}\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{Q}\right)\boldsymbol{H}\emptyset^T(\boldsymbol{x}),\end{aligned} \tag{42}$$

where $\boldsymbol{Q} = \boldsymbol{J} + \frac{2\gamma_I}{(l+u)^2 C}L_{\mathrm{G}}$. The predicted class label of sample $\boldsymbol{x}$ is

$$\mathrm{label}(\boldsymbol{x}) = \underset{i \in \{1,2,\ldots,m\}}{\mathrm{argmax}} \{f_i(\boldsymbol{x})\}. \tag{43}$$

5 Experiments

To evaluate the accuracy and efficiency of the MR-ELM algorithm, we performed experiments on three real-world datasets: the USPS handwritten digit dataset [30], the MIT CBCL dataset and the Extended Yale B dataset [31]. Comparison was made with the following classification methods: NRCM [25], SELM [26], ELM algorithm [27], cutting plane semi-supervised support vector machine algorithm (CutS^{3}VM) [32], linear transductive L_2-SVMs with multiple switchings (L_2-TSVM-MFN), deterministic annealing for semi-supervised linear L_2-SVMs (DA L_2-SVM-MFN) [33] and the sparse regularized least square classification (S-RLSC) algorithm [24]. All the experiments were carried out in MATLAB 7.0.1 environment running in a 3.10GHZ Intel Core$^{\mathrm{TM}}$ i5-2400 with 3-GB RAM.

5.1 Dataset description

The MIT CBCL dataset contains 2,429 face images and 4,548 non-face images. Each image has 19 × 19 pixels and was transformed into a 361-dimensional vector. In the experiment, we used a subset of this database which consists of 1,000 face and 1,000 non-face images. Figure 1 shows some face and non-face images from this dataset.

The USPS database contains 8-bit gray-scale images of classes '0'–'9' with 1,100 data points of each class. Each data point of this dataset is a 16 × 16 image of a handwritten digit and was transformed into a 256-dimensional vector. For each class, we randomly selected 250 data points for our experiments. Therefore, our USPS dataset

Fig. 1 Some image samples from the CBCL dataset. The *first two rows* show some face images, and the *last two rows* show some non-face images

contains 2,500 data points. Some sample data points of class '0' from this dataset are shown in Fig. 2.

The Extended Yale B face database contains 2,114 frontal-face images of 38 individuals. Each data point was a cropped 64 × 64 gray-scale face image and was captured under various lighting conditions. For each class, there are about 60 images, and each image was stacked into a 1,024-dimensional data vector. We use the whole dataset as one of our experimental datasets. Figure 3 shows some sample face images of two individuals from the dataset.

5.2 Parameter selection and experimental settings

In our experiments, the MR-ELM algorithm constructed data adjacency graphs using k-nearest neighbors. Binary edge weights were chosen, and the neighborhood size k was set to be 7 for all the three datasets. We used the Sigmoid function $1/(1+\exp(-(\boldsymbol{a}\cdot\boldsymbol{x}+\mathrm{b})))$ and the Gaussian function $\exp\left(-b\|\boldsymbol{x}-\boldsymbol{a}\|^2\right)$, respectively, for testing test MR-ELM and ELM. We let $L = 1{,}000$ for both ELM and MR-ELM, as the ELM algorithm achieves good generalization performance as long as L is large enough [27]. In [22], two regularization parameters γ_A and $\gamma_I/(l+u)^2$ were split into the ratio 1:9 and we also let $CA = 2\gamma_A/C = 1/C(\gamma_A = 1/2)$, $CI = 2\gamma_I/((l+u)^2\cdot C) = 9/C$ and discuss the effect of parameter selection later. The parameter C was chosen from the range $\{2^{-24}, 2^{-23},\ldots, 2^{24}, 2^{25}\}$ by cross-validation. In order to compare MR-ELM with other semi-supervised ELM algorithms, we further tested SELM and NRCM by using Sigmoid function $1/(1+\exp(-(\mathrm{a}\cdot\boldsymbol{x}+\mathrm{b})))$ and let $L = 1{,}000$ for both SELM and NRCM.

The S-RLSC methods also have regularization parameters γ_A and γ_I. Let $CA = \gamma_A l$, $CI = \gamma_I l/(l+u)^2$, and the kernel function $k(\boldsymbol{x},\boldsymbol{y}) = \exp\left(-\|\boldsymbol{x}-\boldsymbol{y}\|^2/\sigma^2\right)$. In [24], it was found that this algorithm performs well with a wide range of regularization parameters. In our experiments, we also set $CA = 0.005$, $CI = 0.01$ and $\sigma = 0.5$ for comparison.

We performed L_2-TSVM-MFN with multiple switches and DA L_2-SVM-MFN with parameter $\lambda = 0.001$ and $\lambda' = 1$ on all datasets. We also tested CutS^{3}VM with parameters $C_l = C_u = 1$, and set r in the balancing constraint of above three algorithms to the true ratio of the positive points in the unlabeled set.

For each dataset $\boldsymbol{X}$, the order of data points was rearranged randomly. Then, in each class of $\boldsymbol{X}$, 15 % of the data points were left for out-of-sample extension experiment. We denoted by $\boldsymbol{X}_r$ the rest data points of the dataset $\boldsymbol{X}$. In each class of $\boldsymbol{X}_r$, l labeled data points were used to train the algorithms. The number of l (in each class) of the labeled data points varied from 10 to 650 for the

Fig. 2 Some image samples from the USPS handwritten dataset of class '0'

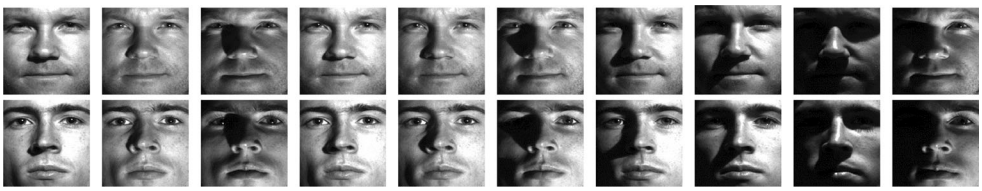

Fig. 3 Some face image samples of two individuals from the Extended Yale B dataset

CBCL dataset, from 5 to 160 for the USPS dataset and from 5 to 50 for the Extended Yale B dataset, respectively. In each independent trial, l labeled data points and u unlabeled data were randomly selected. For the ELM classifier, the training set comprised only the labeled data points from $\boldsymbol{X}_r$. For the other classifiers, the training set consists of the whole $\boldsymbol{X}_r$, including the labeled and the unlabeled data points. For multiclass datasets, the L_2-TSVM-MFN, DA L_2-SVM-MFN and CutS^{3}VM classifiers were trained with a one-versus-rest approach. After obtaining the classifiers, classification was performed first on the unlabeled data points in $\boldsymbol{X}_r$ and then on the out-of-sample extension data points.

5.3 Experimental results

5.3.1 Performance comparison between ELM and MR-ELM

For the CBCL dataset, we first tested MR-ELM and ELM with Sigmoid additive hidden nodes. Thirty independent trials were performed for ELM and MR-ELM. Average classification results of the unlabeled data points in $\boldsymbol{X}_r$ are shown in Table 3, where l is the number of labeled data points in each class. As can be seen from Table 3, due to the small number of training samples (e.g., $l = 5, 10$), the classification accuracy is very low for both algorithms. With the increase in the number of labeled data, MR-ELM has better recognition results than ELM because the manifold regularization term of the MR-ELM model can enhance the smoothness of the classifier function on the manifold. The classification accuracy of MR-ELM with Sigmoid additive hidden nodes is very close to that of MR-ELM with Gaussian kernels, but MR-ELM with Sigmoid additive hidden nodes performs faster, since the inversion of the kernel matrix has to be computed in MR-ELM with Gaussian kernels, which is very time-consuming.

The key parameter selection in MRELM is the ratio of γ_A and $\gamma_I/(l+u)^2$, where γ_A controls the complexity of functions in the ambient space while γ_I controls the smoothness of the final decision function. For the CBCL dataset, we further discuss the impact of this ratio on the performance of MRELM. Table 4 lists the performance of MRELM as a function of the ratio for the remaining unlabeled data when l equals 50. From Table 4, we can observe that MRELM achieves a better performance with proper ratios (around 1:9) for both Sigmoid and Gaussian functions. When the ratio is bigger, MRELM cannot utilize the manifold structure of unlabeled data efficiently. But, the small the ratio is, the more quickly the complexity of functions in the intrinsic geometry of sample probability distribution increases, which could lead to over-fitting problems. As can be seen from Table 4, the performance of MRELM becomes worse when the ratio is smaller. Thus, it is essential to select a medium ratio in real applications.

The experimental results of ELM and MR-ELM on USPS and Extended Yale B datasets are shown in Fig. 4. Obviously, MR-ELM performs better than ELM by utilizing the unlabeled data effectively. In order to compare the separating boundary of MR-ELM with that of ELM, we tested

MR-ELM and ELM on an artificial dataset, which consists of 400 positive data and 400 negative data. We selected 50 labeled data from each class, then trained MR-ELM using 100 labeled data and 700 unlabeled data and

Table 3 Performance comparison between ELM and MR-ELM for the CBCL dataset

The number of labeled data points l	ELM				MR-ELM			
	Sigmoid additive node		Gaussian kernel		Sigmoid additive node		Gaussian kernel	
	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)
$l = 10$	69.40	0.233	55.24	0.001	73.57	0.653	72.64	0.863
$l = 50$	74.47	0.250	63.37	0.016	92.75	0.660	91.15	0.875
$l = 100$	75.36	0.360	69.76	0.032	93.63	0.647	93.04	0.872
$l = 250$	76.88	0.498	75.04	0.156	95.14	0.676	95.26	0.982

Table 4 Parameter selection of MR-ELM for the CBCL dataset

The ratio of γ_A and $\gamma_I/(l+u)^2$	MR-ELM	
	Sigmoid function	Gaussian function
1:1	85.47	83.75
1:2	87.26	85.26
1:4	87.53	86.98
1:6	90.83	89.45
1:9	92.64	91.17
1:10	91.26	91.76
1:20	89.04	88.23
1:50	84.87	83.58
1:100	80.34	79.67

trained ELM only using 100 labeled data. Figure 5 shows final separating boundaries of MR-ELM and ELM. It can be seen that MR-ELM can classify different classes better than ELM by utilizing the manifold structure of the dataset.

5.3.2 *Performance comparison between MR-ELM and other semi-supervised ELM algorithms*

Next, we compared MR-ELM with other semi-supervised ELM algorithms on USPS and Extended Yale B datasets. Twenty independent trials were repeated by using Sigmoid additive hidden nodes. Average classification results of all algorithms are shown in Fig. 6. As can be seen from Fig. 6, by adding the penalty norm of the output weights into our model, our method obtains the higher accuracy than NRCM and SELM, which indicates that the penalty norm of the output weights has important influence on the performance of semi-supervised ELM algorithms. The accuracy of NRCM is slightly higher than SELM due to the intra-class and inter-class regularization terms. Thus, compared with other manifold regularized ELM algorithms, MR-ELM can generate more smooth decision functions and achieve better performance.

5.3.3 *Performance comparison between MR-ELM and other algorithms*

We compared other algorithms with MR-ELM using Sigmoid additive hidden nodes. The recognition results of all the algorithms are shown in Tables 5, 6 and 7, respectively. For each dataset, the classification accuracy and training time are averaged over 20 independent trials. In Tables 5, 6 and 7, the best classification results are in boldface for each fixed value of l. As can be seen from the tables, the classification accuracy is low for all algorithms when a small number of labeled data are available. The performance of all algorithms is improved with the increase in labeled data. The classification accuracy of the S-RLSC algorithm is much better than L_2-TSVM-MFN, DA L_2-SVM-MFN and CutS^{3}VM. The recognition result of the proposed MR-ELM algorithm is very close to that of the S-RLSC algorithm, but it performed much faster than the S-RLSC algorithm. As observed from Tables 6 and 7, the recognition accuracy of the L_2-TSVM-MFN, DA L_2-SVM-MFN and CutS^{3}VM classifiers decreases due to the complexity of natural data distribution of the datasets. Moreover, this kind of multiclass classification approach also increases the running time of these algorithms. In contrast, the speed of the MR-ELM algorithm is not sensitive to the data distribution of datasets. It can perform well in multiclass classification cases by means of the intrinsic geometry of data distribution.

In addition, we tested MR-ELM with Sigmoid additive hidden nodes under different numbers of hidden nodes L on the USPS and the Extended Yale B datasets. For each dataset, the classification accuracy and training time are averaged over 30 independent trials. The experimental result of the MR-ELM algorithm is shown in Table 8. As can be seen from Table 8, when $l = 100$ and $L = 2{,}000$, the recognition accuracy of MR-ELM algorithm exceeds that of the S-RLSC in Table 6 for the USPS dataset. When $l = 30$ and $L = 2{,}000$, the recognition accuracy of MR-ELM algorithm is only a little lower than that of the

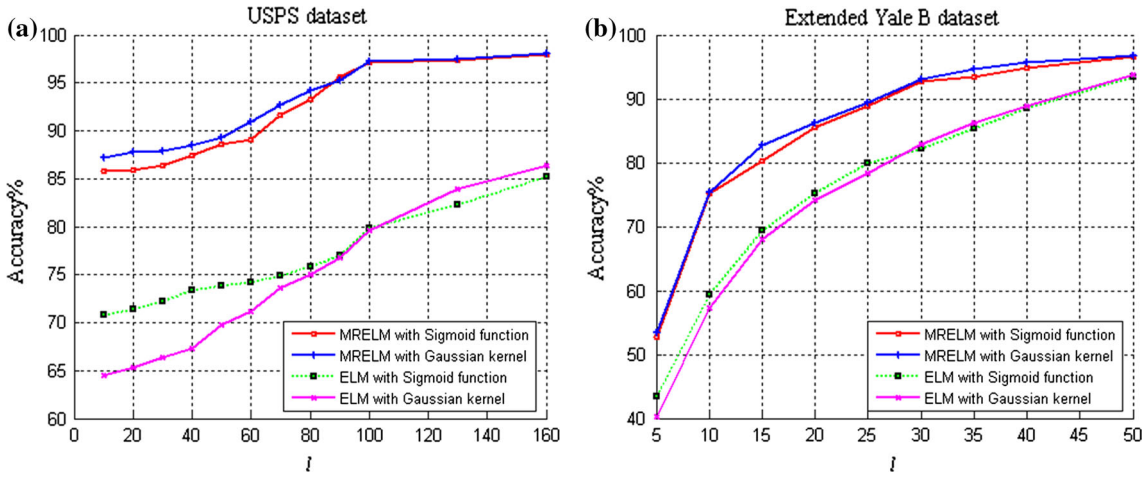

Fig. 4 Performance comparison between ELM and MR-ELM on USPS and Extended Yale B datasets. **a** Result on the USPS dataset. **b** Result on the Extended Yale B dataset

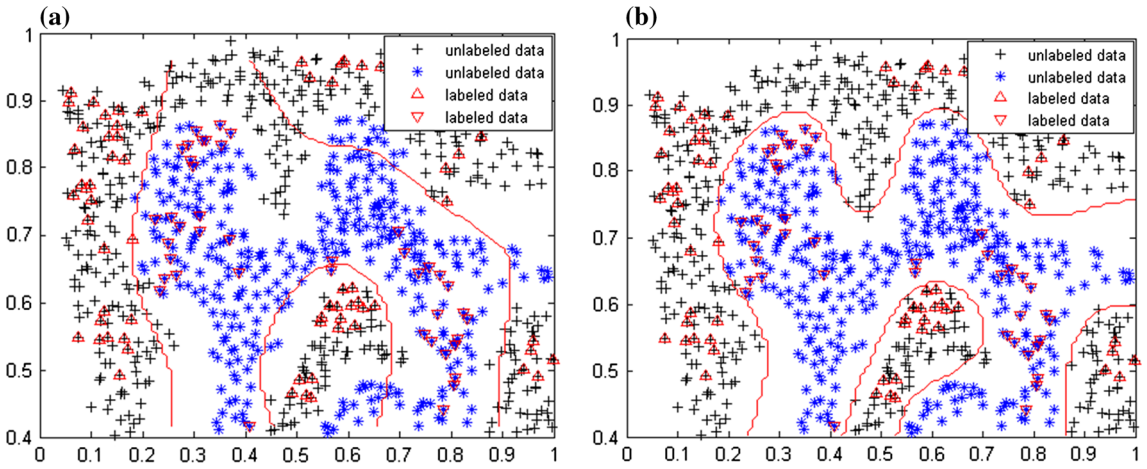

Fig. 5 Separating boundaries of ELM and MR-ELM classifiers on the artificial dataset. **a** ELM. **b** MR-ELM

S-RLSC in Table 7. Moreover, it always runs faster than the S-RLSC algorithm. It should be noted that the larger the number of hidden nodes is, the more slowly the MR-ELM runs. If L is large to some extent, it is difficult to improve the performance of MR-ELM further, which is consistent with the theoretical analysis.

The out-of-sample extension results of the algorithms on three datasets are shown in Figs. 7, 8 and 9, respectively. We performed MR-ELM using 1,000 hidden nodes. As can be seen from Figs. 7, 8 and 9, the MR-ELM algorithm has better recognition results than the L_2-TSVM-MFN, DA L_2-SVM-MFN or CutS^{3}VM for testing data and has comparable generalization performance to that of the S-RLSC algorithm. So our proposed MR-ELM algorithm tends to have better scalability and achieve similar or much better generalization performance at a relatively faster learning speed.

6 Conclusions

In this paper, we extend the MR framework and demonstrate the relation between the E-MR framework and ELM. The proposed MR-ELM can be naturally derived from the E-MR framework, which can be applied to classification and regression problems with labeled and unlabeled examples available. Experiments on real-world datasets verify the effectiveness of MR-ELM. Although the classification accuracy rate of MR-ELM is a little lower than that of S-RLSC, it runs about dozens of times faster

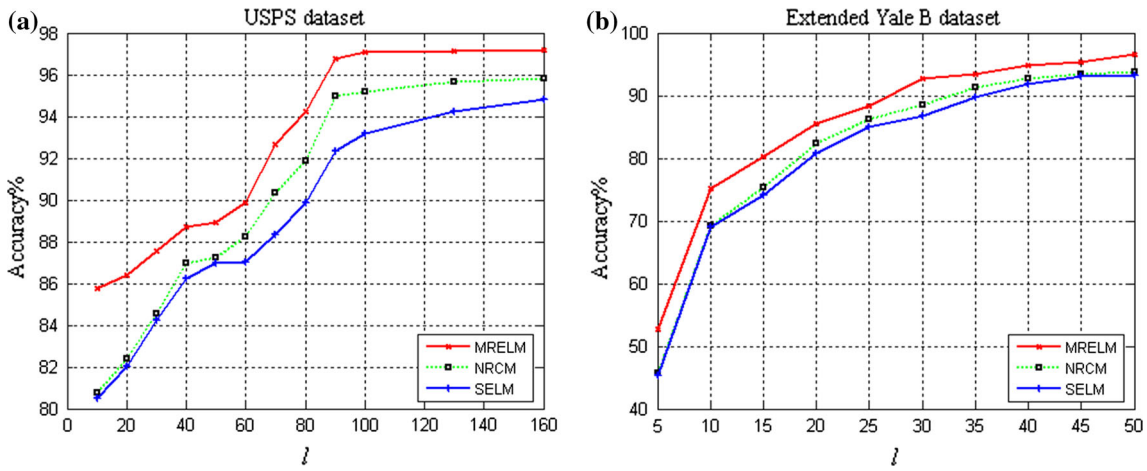

Fig. 6 Results of all semi-supervised ELM algorithms on USPS and Extended Yale B datasets. **a** Result on the USPS dataset. **b** Result on the Extended Yale B dataset

Table 5 Performance comparison of all the algorithms for the CBCL Dataset

The number of labeled data points l	MR-ELM		S-RLSC		L_2-TSVM-MFN		DA L_2-SVM-MFN		CutS^{3}VM	
	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)
$l = 5$	69.65	0.860	**73.89**	58.579	62.62	2.328	50.81	14.594	53.59	2.190
$l = 10$	73.57	0.863	**74.75**	58.647	58.41	2.172	59.67	11.656	64.46	1.976
$l = 50$	92.75	0.875	**94.58**	60.152	72.13	1.572	67.21	10.672	82.85	1.620
$l = 250$	95.14	0.982	**98.66**	62.674	76.72	1.456	76.39	7.219	84.46	1.569
$l = 450$	96.27	0.993	**98.21**	64.339	77.05	1.203	77.15	8.375	85.38	1.421
$l = 650$	97.28	1.036	**98.47**	62.785	77.71	1.031	78.69	5.010	87.15	1.247

Table 6 Performance comparison of all the algorithms for the USPS Dataset

The number of labeled data points l	MR-ELM		S-RLSC		L_2-TSVM-MFN		DA L_2-SVM-MFN		CutS^{3}VM	
	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)
$l = 5$	**73.59**	2.095	71.63	125.732	69.23	29.363	67.49	163.420	68.54	13.420
$l = 10$	85.74	2.147	**85.58**	125.854	77.53	20.312	71.36	125.749	75.37	12.658
$l = 40$	88.69	2.183	**94.30**	126.749	79.92	17.565	78.28	95.218	80.35	10.962
$l = 100$	97.09	2.274	**97.16**	127.504	80.45	15.689	80.75	82.730	82.63	8.594
$l = 160$	**97.95**	2.282	97.68	129.286	81.71	12.672	82.54	67.062	83.18	7.103

Table 7 Performance comparison of all the algorithms for the Extended Yale B dataset

The number of labeled data points l	MR-ELM		S-RLSC		L_2-TSVM-MFN		DA L_2-SVM-MFN		CutS^{3}VM	
	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)	Accuracy (%)	Training time(s)
$l = 5$	52.68	5.603	**64.41**	152.644	55.17	60.427	38.47	361.928	63.93	75.249
$l = 10$	75.25	5.620	**83.18**	155.972	62.76	58.958	58.71	273.536	69.82	69.162
$l = 20$	85.61	5.622	**93.10**	152.177	67.72	49.514	68.49	247.923	75.91	65.368
$l = 30$	92.78	5.619	**95.24**	154.921	71.35	42.943	73.59	190.380	79.56	58.532
$l = 40$	94.86	5.624	**97.12**	152.228	75.25	35.519	76.84	163.476	80.13	54.348
$l = 50$	96.55	5.603	**97.12**	155.597	78.87	28.524	79.39	144.258	82.52	46.363

Table 8 Experimental results of the MR-ELM algorithm on USPS and Extended Yale B datasets

Datasets	The number of labeled data points l	The number of hidden nodes L	Accuracy (%)	Training time(s)
USPS	$l = 100$	500	96.49	1.282
		1,000	97.09	2.274
		2,000	97.25	4.674
The Extended Yale B	$l = 30$	500	91.32	2.673
		1,000	92.78	5.619
		2,000	94.86	10.843

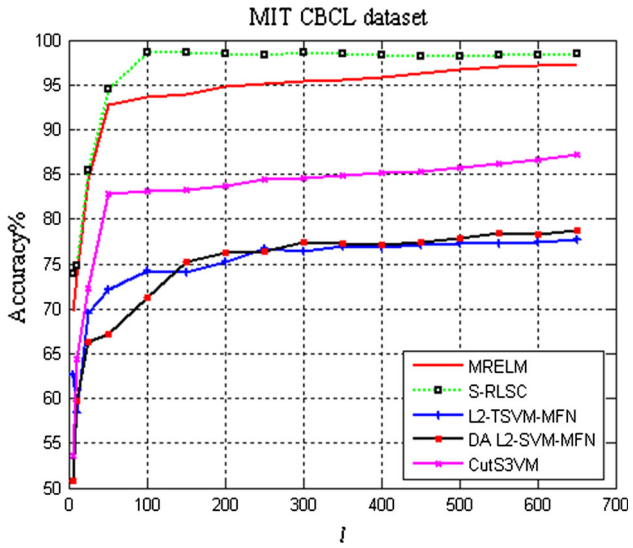

Fig. 7 Out-of-sample extension classification results on the CBCL dataset

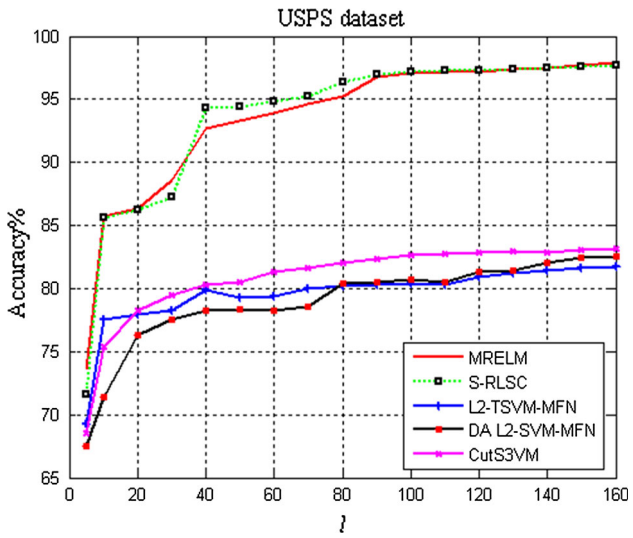

Fig. 8 Out-of-sample extension classification results on the USPS dataset

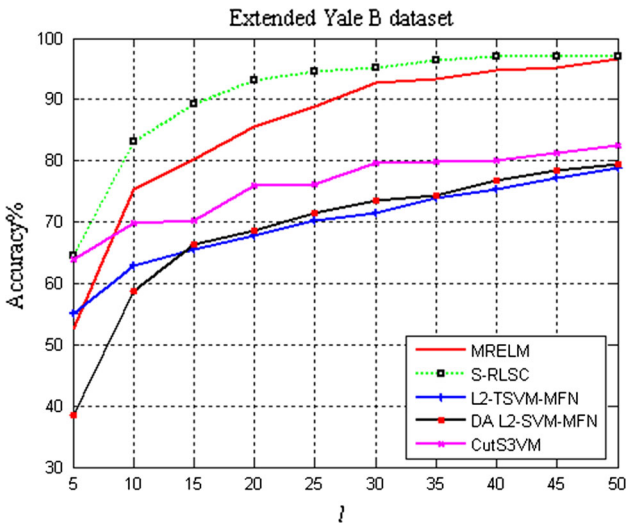

Fig. 9 Out-of-sample extension classification results on the Extended Yale B dataset

than S-RLSC. MR-ELM significantly outperforms L_2-TSVM-MFN, DA L_2-SVM-MFN, CutS^{3}VM, NRCM and SELM. Thus, it is the most cost-efficient method. In the near future, we will further optimize our proposed framework and study the sparse regularization problem for our framework.

Acknowledgments This work is supported by the National Natural Science Foundation of China, China (No. 61403394) and the Fundamental Research Funds for the Central Universities (No. 2014QNA45).

References

1. Tang X, Han M (2009) Partial Lanczos extreme learning machine for single-output regression problems. Neurocomputing 72(13):3066–3076
2. Liu Q, He Q, Shi ZZ (2008) Extreme support vector machine classifier. Lect Notes Comput Sci 5012:222–233
3. Frénay B, Verleysen M (2010) Using SVMs with randomised feature spaces: an extreme learning approach. In: Proceedings of the 18th ESANN, Bruges, Belgium, pp 315–320
4. Miche Y et al (2010) OP-ELM: optimally pruned extreme learning machine. IEEE Trans Neural Netw 21(1):158–162
5. Deng W, Zheng Q, Chen L (2009) Regularized extreme learning machine. In: Proceedings of the IEEE symposium on CIDM, pp 389–395
6. Huang G-B, Zhu Q-Y, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In Proceedings of the IJCNN, Budapest, Hungary, vol 2, pp 985–990
7. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
8. Huang G-B, Chen L, Siew C-K (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
9. Huang G-B, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70(16):3056–3062

10. Huang G-B, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71(16):3460–3468
11. Wang G, Zhao Y, Wang D (2008) A protein secondary structure prediction frame- work based on the extreme learning machine. Neurocomputing 72(1–3):262–268
12. Lan Y, Soh YC, Huang G-B (2008) Extreme Learning Machine based bacterial protein subcellular localization prediction. In: Proceedings of the IEEE international joint conference on neural networks, IJCNN 2008, Hong Kong, pp 1859–1863
13. Zhang R, Huang G-B, Sundararajan N, Saratchandran P (2007) Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans Comput Biol Bioinform 4(3):485–495
14. Mohammed AA, Minhas R, Jonathan Wu QM, Sid-Ahmed MA (2011) Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognit 44(10–11):2588–2597
15. Nizar AH, Dong ZY, Wang Y (2008) Power utility nontechnical loss analysis with extreme learning machine method. IEEE Trans Power Syst 23(3):946–955
16. Decherchi S, Gastaldo P, Dahiya RS, Valle M, Zunino R (2011) Tactile data classification of contact materials using computational intelligence. IEEE Trans Robot 27(3):635–639
17. Decherchi S, Gastaldo P, Zunino R, Cambria E, Redi J (2013) Circular-ELM for the reduced-reference assessment of perceived image quality. Neurocomputing 102:78–89
18. Cambria E, Hussain A (2012) Sentic album: content-, concept-, and context-based online personal photo management system. Cogn Comput 4(4):477–496
19. Kumar MA (2010) An investigation on linear SVM and its variants for text categorization. In: Second international conference on machine learning and computing (ICMLC), pp 27–31
20. Wang YG, Cao FL, Yuan YB (2011) A study on effectiveness of extreme learning machine. Neurocomputing 74:2483–2490
21. Cao FL, Liu B, Park DS (2013) Image classification based on effective extreme learning machine. Neurocomputing 102:90–97
22. Belkin M, Sindhwani V, Niyogi P (2006) Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 7:2399–2434
23. Xue H, Chen S, Yang Q (2009) Discriminatively regularized least-squares classification. Pattern Recognit 42(1):93–104
24. Fan M, Gu N, Qiao H et al (2011) Sparse regularization for semi-supervised classification. Pattern Recognit 44(8):1777–1784
25. Li Lina, Liu Dayou, Ouyang Jihong (2012) A new regularization classification method based on extreme learning machine in network data. J Inf Comput Sci 9(12):3351–3363
26. Liu Junfa, Chen Yiqiang, Liu Mingjie et al (2011) SELM: semi-supervised ELM with application in sparse calibrated location estimation. Neurocomputing 74(16):2566–2572
27. Huang G-B, Zhou HM, Ding XJ (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B Cybern 42(2):513–529
28. He Xiaofei, Cai Deng, Shao Yuanlong, Bao Hujun, Han Jiawei (2011) Laplacian regularized Gaussian mixture model for data clustering. IEEE Trans Knowl Data Eng 23(9):1406–1418
29. Huang G-B et al (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2:107–122
30. Hull JJ (1998) A data base for hand written text recognition research. IEEE Trans Pattern Anal Mach Intell 16(5):550–554
31. Lee KC, Ho J, Kriegman D (2005) Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans Pattern Anal Mach Intell 27(5):684–698
32. Zhao B, Wang F, Zhang C (2008) CutS3VM: a fast semi-supervised SVM algorithm. In: The 14th ACM SIGKDD international conference on knowledge discovery & data mining (KDD), Las Vegas, NV, USA, pp 830–838
33. Sindhwani V, Keerthi SS (2006) Large scale semi-supervised linear SVMs. In: 29th Annual international ACM SIGIR. Technical report