Neural Comput & Applic
DOI 10.1007/s00521-012-1184-y

ISNN2012

Training extreme learning machine via regularized correntropy criterion

Hong-Jie Xing · Xin-Mei Wang

Received: 28 April 2012 / Accepted: 11 September 2012

Abstract In this paper, a regularized correntropy criterion (RCC) for extreme learning machine (ELM) is proposed to deal with the training set with noises or outliers. In RCC, the Gaussian kernel function is utilized to substitute Euclidean norm of the mean square error (MSE) criterion. Replacing MSE by RCC can enhance the anti-noise ability of ELM. Moreover, the optimal weights connecting the hidden and output layers together with the optimal bias terms can be promptly obtained by the half-quadratic (HQ) optimization technique with an iterative manner. Experimental results on the four synthetic data sets and the fourteen benchmark data sets demonstrate that the proposed method is superior to the traditional ELM and the regularized ELM both trained by the MSE criterion.

Keywords Extreme learning machine · Correntropy · Regularization term · Half-quadratic optimization

1 Introduction

Recently, Huang et al. [1] proposed a novel learning method for single-hidden layer feedforward networks (SLFNs), which is named as extreme learning machine (ELM). Interestingly, the weights connecting the input and hidden layers together with the bias terms are all randomly initialized. Then, the weights connecting the hidden and output layers can be directly determined by the least squares method which is based on the Moore–Penrose generalized inverse [2]. Therefore, the training speed of ELM is extremely fast, which is the main advantage of this method. ELM has been widely used in applications of face recognition [3], image processing [4, 5], text categorization [6], time series prediction [7], and nonlinear model identification [8].

In recent years, many variants of ELM have been emerged. Huang et al. [9] provided us with a detailed survey. Besides the methods reviewed in the literature [9], some new improved models of ELM have been proposed. Wang et al. [10] proposed an improved algorithm named EELM. EELM can ensure the full column rank of the hidden layer output matrix while the traditional ELM cannot fulfill sometimes. By incorporating the forgetting mechanism, Zhao et al. [11] proposed a novel online sequential ELM named FOS-ELM, which demonstrates higher accuracy with fewer training time in comparison with the ensemble of online sequential ELMs. Savitha et al. [12] proposed a fast learning fully complex-valued ELM classifier for handling real-valued classification problems. In order to automatically generate networks, Zhang et al. [13] proposed an improved ELM with adaptive growth of hidden nodes (AG-ELM). Experimental results demonstrate that AG-ELM can get more compact network architecture compared with the incremental ELM. Cao et al. [14] proposed an improved learning algorithm called self-adaptive evolutionary ELM (SaE-ELM), which performs better than the evolutionary ELM and possesses better generalization ability in comparison with its related methods.

However, there exists two limitations of ELM, which are listed below.

H.-J. Xing (✉)
Key Laboratory of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071002, China
e-mail: hjxing@hbu.edu.cn

X.-M. Wang
College of Mathematics and Computer Science, Hebei University, Baoding 071002, China

Published online: 26 September 2012

- When there are noises or outliers in the training set, ELM may yield poor generalization performance [15]. The reason lies that the mean square error (MSE) criterion assumes the Gaussian distributed error. However, this assumption does not always hold in the real-world applications.
- The original least square used in ELM is sensitive to the presence of noises or outliers. These unusual samples may skew the results of the least square analysis [16].

Recently, the definition and properties of correntropy were proposed by Santamaria et al. [17]. MSE is regarded as a global similarity measure, whereas correntropy a local one [18]. Due to its flexibility, correntropy has been successfully utilized to design different cost functions. Jeong et al. [19] extended the minimum average correlation energy (MACE) to its corresponding nonlinear version by using correntropy. Moreover, they verified that the correntropy MACE is more robust against distortion and possesses more generalization and rejection abilities in comparison with the linear MACE. Yuan and Hu [20] proposed a robust feature extraction framework based on correntropy and firstly solved their optimization problem by the half-quadratic optimization technique [21]. He et al. [22–24] introduced a sparse correntropy framework for deriving robust sparse representations of face images for recognition. Liu et al. [18] utilized correntropy to construct the objective function for training linear regression model. They primarily demonstrated that maximum correntropy criterion outperforms MSE and minimum error entropy on the regression examples with noises. However, the coefficients of their linear regressor are updated by the gradient-based optimization method which is time-consuming. To further enhance the performance of the method proposed by Liu et al. [18], He et al. [25] added an L1-norm regularization term into the maximum correntropy criterion. Moreover, they utilized half-quadratic optimization technique and feature-sign search algorithm to optimize the coefficients in their linear model.

In order to enhance the anti-noise ability of ELM, a regularized correntropy criterion (RCC) is proposed in the paper. The main contributions of the proposed method, that is, ELM-RCC are summarized in the following three aspects.

- In ELM-RCC, the correntropy-based criterion is utilized to replace the MSE criterion in ELM, which makes ELM-RCC more robust against noises compared with ELM.
- To further improve the generalization ability of ELM-RCC, a regularization term is added into its objective function.
- The related propositions of the proposed method are given. Moreover, the algorithmic implementation and computational complexity of ELM-RCC are also provided.

The organization of the paper is as follows. The traditional ELM, the regularized ELM, and the basic concept of correntropy are briefly reviewed in Sect. 2. In Sect. 3, the proposed method, that is, ELM-RCC, together with its optimization method is expatiated. Experiments to validate the proposed method are conducted in Sect. 4. Finally, the conclusions are summarized in Sect. 5.

2 Preliminaries

2.1 Extreme learning machine

Extreme learning machine was first proposed by Huang et al. [1]. For ELM, the weights connecting the input and hidden layers together with the bias terms are initialized randomly, while the weights connecting the hidden and output layers are determined analytically. Therefore, the learning speed of this method is extremely faster than that of the traditional gradient descent-based learning method.

Given N arbitrary distinct samples $\{\mathbf{x}_p, \mathbf{t}_p\}_{p=1}^{N}$, where $\mathbf{x}_p \in \mathcal{R}^d$ and $\mathbf{t}_p \in \mathcal{R}^m$. The output vector of a standard single-hidden layer feedforward network (SLFN) with N_H hidden units and the activation function $f(\cdot)$ is mathematically modeled as

$$\mathbf{y}_p = \sum_{j=1}^{N_H} \boldsymbol{\beta}_j f(\mathbf{w}_j \cdot \mathbf{x}_p + b_j), \quad p = 1, 2, \ldots, N, \tag{1}$$

where $\mathbf{w}_j$ is the weight vector connecting the jth hidden unit and all the input units, b_j denotes the bias term for the jth hidden unit, $\mathbf{w}_j \cdot \boldsymbol{x}_p$ denotes the inner product of $\mathbf{w}_j$ and $\mathbf{x}_p$, $\boldsymbol{\beta}_j$ is the weight vector connecting the jth hidden unit and all the output units, and $\mathbf{y}_p$ denotes the output vector of the SLFN for the pth input vector $\mathbf{x}_p$.

For ELM, the weights connecting the input and hidden units and the bias terms are randomly generated rather than by tuning. By doing so, the nonlinear system can be converted to a linear system:

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta}, \tag{2}$$

where

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1), & \ldots & f(\mathbf{w}_{N_H} \cdot \mathbf{x}_1 + b_{N_H}) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1), & \ldots & f(\mathbf{w}_{N_H} \cdot \mathbf{x}_N + b_{N_H}) \end{pmatrix},$$

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{N_H})^T, \mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T,$$

and $\mathbf{T} = (\mathbf{t}_1, \ldots, \mathbf{t}_N)^T$. Moreover, $\mathbf{H} = (h_{pj})_{N \times N_H}$ is the output matrix of the hidden layer, $\boldsymbol{\beta}$ is the matrix of the weights connecting the hidden and output layers, $\mathbf{Y}$ is the output matrix of the output layer, and $\mathbf{T}$ is the matrix of targets. Thus, the matrix of the weights connecting the hidden and output layers can be determined by solving

$$\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_F, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The solution of (3) can be obtained as follows

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{4}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse. Different methods are used to determine $\mathbf{H}^{\dagger}$. When $\mathbf{H}^T\mathbf{H}$ is nonsingular, the orthogonal project method can be utilized to calculate $\mathbf{H}^{\dagger}$ [1]:

$$\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T. \tag{5}$$

However, there are still some shortages that restrict the further development of ELM, such as the noises in training samples are not necessarily Gaussian distributed, and the solution of the original least square problem (3) is sensitive to the noises.

2.2 Regularized ELM

In order to improve the stability and generalization ability of the traditional ELM, Huang et al. proposed the equality constrained optimization-based ELM [26]. In the proposed method, the solution (4) is re-expressed as follows

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{H}^T\mathbf{H} + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{H}^T\mathbf{T}, \tag{6}$$

where C is a constant and $\mathbf{I}$ is the identity matrix.

Let $\lambda = \frac{1}{C}$, (6) can be rewritten as

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{T}. \tag{7}$$

The solution (7) can be obtained by solving the following optimization problem:

$$\min_{\boldsymbol{\beta}} (\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_F^2 + \lambda\|\boldsymbol{\beta}\|_F^2), \tag{8}$$

where $\|\boldsymbol{\beta}\|_F^2 = \sum_{j=1}^{N_H} \|\boldsymbol{\beta}_j\|_2^2$ is regarded as the regularization term with $\|\cdot\|_2$ denotes the L2-norm of the vector $\boldsymbol{\beta}_j$. Moreover, λ in (8) denotes the regularization parameter, which plays an important role to control the trade-off between the training error and the model complexity.

Appendix 1 provides a detailed mathematical deduction of the solution (7) from the optimization problem (8).

2.3 Correntropy

From the information theoretic learning (ITL) [27] point of view, correntropy [18] is regarded as a generalized correlation function [17]. It is a measure of similarity by studying the interaction of the given feature vectors. Given two arbitrary random variables A and B, their correntropy can be defined as

$$V_\sigma(A, B) = E[\kappa_\sigma(A - B)], \tag{9}$$

where $\kappa_\sigma(\cdot)$ is the kernel function that satisfies Mercer's theorem [28] and $E[\cdot]$ denotes the mathematical expectation.

In the paper, we only consider the Gaussian kernel with finite number of data samples $\{(a_i, b_i)\}_{i=1}^M$. Thus, correntropy can be estimated by

$$\hat{V}_{M,\sigma}(A, B) = \frac{1}{M}\sum_{i=1}^{M} \kappa_\sigma(a_i - b_i), \tag{10}$$

where $\kappa_\sigma(\cdot)$ is given by $\kappa_\sigma(a_i - b_i) \triangleq G(a_i - b_i) = \exp(-\frac{(a_i - b_i)^2}{2\sigma^2})$. Therefore, (10) can be rewritten as

$$\hat{V}_{M,\sigma}(A, B) = \frac{1}{M}\sum_{i=1}^{M} G(a_i - b_i). \tag{11}$$

The maximum of correntropy function (9) is called the maximum correntropy criterion (MCC) [18]. Since correntropy is insensitive to noises or outliers, it is superior to MSE when there are impulsive noises in training samples [18].

3 ELM based on regularized correntropy criterion

In this section, the regularized correntropy criterion for training ELM, namely ELM-RCC is introduced. Moreover, its optimization method is also presented.

3.1 L2-norm-based RCC

Replacing the objective function (3) of ELM with maximizing the correntropy between the target and network output variables, we can get a new criterion as follows

$$J(\tilde{\boldsymbol{\beta}}) = \max_{\tilde{\boldsymbol{\beta}}} \sum_{p=1}^{N} G(\mathbf{t}_p - \mathbf{y}_p), \tag{12}$$

where $\mathbf{t}_p$ denotes the target vector for the pth input vector $\mathbf{x}_p$ and $\mathbf{y}_p$ denotes the output of the SLFN for $\mathbf{x}_p$, which is given by

$$\mathbf{y}_p = \sum_{j=1}^{N_H} \boldsymbol{\beta}_j f(\mathbf{w}_j \cdot \mathbf{x}_p + b_j) + \boldsymbol{\beta}_0, \quad p = 1, 2, \ldots, N, \tag{13}$$

where $\boldsymbol{\beta}_j$ is the weight vector connecting the jth hidden unit and all the output units, while $\boldsymbol{\beta}_0 = (\beta_{01}, \beta_{02}, \ldots, \beta_{0m})^T$ is the bias term vector for the output units.

The matrix form of (13) can be expressed as follows

$$\mathbf{Y} = \tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}}, \tag{14}$$

where

$$\tilde{\mathbf{H}} = \begin{pmatrix} 1, & f(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1), & \cdots, & f(\mathbf{w}_{N_H} \cdot \mathbf{x}_1 + b_{N_H}) \\ \vdots & \vdots & \ddots & \vdots \\ 1, & f(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1), & \cdots, & f(\mathbf{w}_{N_H} \cdot \mathbf{x}_N + b_{N_H}) \end{pmatrix}$$

and $\tilde{\boldsymbol{\beta}} = (\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{N_H})^T$. Moreover, (13) can be rewritten as

$$\mathbf{y}_p = \sum_{j=0}^{N_H} h_{pj}\tilde{\boldsymbol{\beta}}_j, \tag{15}$$

where h_{pj} denotes the (p, j)th element of $\mathbf{H}$ with indices $j = 1, 2, \ldots, N_H$ and $h_{p0} = 1$, while $\tilde{\boldsymbol{\beta}}_j = \boldsymbol{\beta}_j$ with indices $j = 1, 2, \ldots, N_H$ and $\tilde{\boldsymbol{\beta}}_0 = \boldsymbol{\beta}_0$.

Furthermore, we augment (12) by adding an L2 regularization term, which is defined as follows

$$J_{L2}(\tilde{\boldsymbol{\beta}}) = \max_{\tilde{\boldsymbol{\beta}}} \left[\sum_{p=1}^{N} G\left(\mathbf{t}_p - \sum_{j=0}^{N_H} h_{pj}\tilde{\boldsymbol{\beta}}_j \right) - \lambda \|\tilde{\boldsymbol{\beta}}\|_F^2 \right], \tag{16}$$

where λ is the regularization parameter. There are many approaches for solving the optimization problem (16) in the literature, for example, half-quadratic optimization technique [20, 21], expectation-maximization (EM) method [29], and gradient-based method [18, 19]. In this paper, the half-quadratic optimization technique is utilized.

According to the theory of convex conjugated functions [21, 30], the following proposition [20] exists.

Proposition 1 *For* $G(\mathbf{z}) = \exp\left(-\frac{\|\mathbf{z}\|^2}{2\sigma^2}\right)$, *there exists a convex conjugated function* φ, *such that*

$$G(\mathbf{z}) = \sup_{\alpha \in \mathcal{R}^-} \left(\alpha \frac{\|\mathbf{z}\|^2}{2\sigma^2} - \varphi(\alpha) \right). \tag{17}$$

Moreover, for a fixed $\mathbf{z}$, *the supremum is reached at* $\alpha = -G(\mathbf{z})$ [20].

Hence, introducing (17) into the objective function of (16), the following augmented objective function can be obtained.

$$\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}) = \max_{\tilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}} \Bigg[\sum_{p=1}^{N} \left(\alpha_p \frac{\|\mathbf{t}_p - \sum_{j=0}^{N_H} h_{pj}\tilde{\boldsymbol{\beta}}_j\|_2^2}{2\sigma^2} - \varphi(\alpha_p) \right) - \lambda \|\tilde{\boldsymbol{\beta}}\|_F^2 \Bigg], \tag{18}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ stores the auxiliary variables appeared in the half-quadratic optimization. Moreover, for a fixed $\tilde{\boldsymbol{\beta}}$, the following equation holds

$$J_{L2}(\tilde{\boldsymbol{\beta}}) = \hat{J}_{L2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\alpha}). \tag{19}$$

The local optimal solution of (18) can be calculated iteratively by

$$\alpha_p^{\tau+1} = -G\left(\mathbf{t}_p - \sum_{j=0}^{N_H} h_{pj}\tilde{\boldsymbol{\beta}}_j^{\tau} \right), \tag{20}$$

and

$$\tilde{\boldsymbol{\beta}}^{\tau+1} = \arg\max_{\tilde{\boldsymbol{\beta}}} \left[Tr\left((\mathbf{T} - \tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}})^T \boldsymbol{\Lambda} (\mathbf{T} - \tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}}) - \lambda \tilde{\boldsymbol{\beta}}^T \tilde{\boldsymbol{\beta}} \right) \right], \tag{21}$$

where τ denotes the τth iteration and $\boldsymbol{\Lambda}$ is a diagonal matrix with its primary diagonal element $\boldsymbol{\Lambda}_{ii} = \alpha_i^{\tau+1}$.

Through mathematical deduction, we can obtain the solution of (21) below

$$\tilde{\boldsymbol{\beta}}^{\tau+1} = (\tilde{\mathbf{H}}^T \boldsymbol{\Lambda} \tilde{\mathbf{H}} - \lambda \mathbf{I})^{-1} \tilde{\mathbf{H}}^T \boldsymbol{\Lambda} \mathbf{T}. \tag{22}$$

One can refer to Appendix 2 for the detailed deduction.

It should be mentioned here that the objective function $\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}, \boldsymbol{\alpha})$ in (18) converges after a certain number of iterations, which can be summarized as follows

Proposition 2 *The sequence* $\{\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau}, \boldsymbol{\alpha}^{\tau}), \tau = 1, 2, \ldots\}$ *generated by the iterations* (20) *and* (22) *converges.*

Proof According to Proposition 1 and (21), we can find that $\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau}, \boldsymbol{\alpha}^{\tau}) \leq \hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau+1}, \boldsymbol{\alpha}^{\tau}) \leq \hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau+1}, \boldsymbol{\alpha}^{\tau+1})$. Therefore, the sequence $\{\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau}, \boldsymbol{\alpha}^{\tau}), \tau = 1, 2, \ldots\}$ is non-decreasing. Moreover, it was shown in [18] that correntropy is bounded. Thus, we know that $J_{L2}(\tilde{\boldsymbol{\beta}})$ is bounded, and by (19), $\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau}, \boldsymbol{\alpha}^{\tau})$ is also bounded. Consequently, we verify that $\{\hat{J}_{L2}(\tilde{\boldsymbol{\beta}}^{\tau}, \boldsymbol{\alpha}^{\tau}), \tau = 1, 2, \ldots\}$ converges. □

Furthermore, comparing (22) with (5), we have

Proposition 3 *Let the elements of the bias term vector* $\boldsymbol{\beta}_0$ *be zero. When* $\lambda = 0$ *and* $\boldsymbol{\Lambda} = -\mathbf{I}$, *RCC is equivalent to the MSE criterion.*

Proof As suggested in Proposition 1, the optimal weight matrix $\tilde{\boldsymbol{\beta}}^*$ of (18) can be obtained after a certain number of iterations. Moreover, if $\lambda = 0$ and $\boldsymbol{\Lambda} = -\mathbf{I}$, $\tilde{\boldsymbol{\beta}}^* = (\tilde{\mathbf{H}}^T \boldsymbol{\Lambda} \tilde{\mathbf{H}} - \lambda \mathbf{I})^{-1} \tilde{\mathbf{H}}^T \boldsymbol{\Lambda} \mathbf{T} = (\tilde{\mathbf{H}}^T \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{T}$. Since all the elements of $\boldsymbol{\beta}_0$ are assumed to be zero, so we have $\tilde{\boldsymbol{\beta}}^* = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{T}$, which is the same with (5). Therefore, the proposition holds. □

The whole procedure of training ELM with the RCC criterion is summarized in Algorithm 1. It should be mentioned here that E_τ in Algorithm 1 is the correntropy in the τth iteration, which is calculated by

Algorithm 1 Training ELM using RCC

```
**Input:** Input matrix $\mathbf{X} = (x_{p,i})_{N \times d}$, target matrix $\mathbf{T} = (t_{p,k})_{N \times m}$
**Output:** The optimal weight vectors $\tilde{\boldsymbol{\beta}}_j^*, j = 1, \cdots, N_H$, the optimal bias term vector $\tilde{\boldsymbol{\beta}}_0^*$
**Initialization:** Number of hidden units $N_H$, regularization parameter $\lambda$, maximum number of iterations $I_{HQ}$, termination tolerance $\epsilon$
**Step 1:** Randomly initialize the $N_H$ weight vectors connecting the input and hidden layers $\{\mathbf{w}_j\}_{j=1}^{N_H}$ together with their corresponding bias terms $\{b_j\}_{j=1}^{N_H}$.
**Step 2:** Calculate the hidden layer output matrix $\tilde{\mathbf{H}}$.
**Step 3:** Update the weight vectors $\tilde{\boldsymbol{\beta}}_j, j = 1, \cdots, N_H$ and the bias term vector $\tilde{\boldsymbol{\beta}}_0$.
**repeat**
  **for** $\tau = 1, 2, \ldots, I_{HQ}$ do
    Update the elements of the auxiliary vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^T$ by (20).
    Update the bias term vector and the weight vectors $\tilde{\boldsymbol{\beta}}_j, j = 0, \cdots, N_H$ by (22).
  **end for**
**until** $|E_\tau - E_{\tau-1}| < \epsilon$
```

$$E_\tau = \frac{1}{N}\sum_{p=1}^{N} G(\mathbf{t}_p - \mathbf{y}_p). \tag{23}$$

3.2 Computational complexity

According to Algorithm 1, Step 2 takes $N(N_H + 1)[2(N_H + 1) + m] + (N_H + 1)^3$ operations. For Step 3, the computational cost of the auxiliary vector $\boldsymbol{\alpha}$ in each iteration is $O(N)$. Moreover, the calculation of $\widetilde{\boldsymbol{\beta}}$ in each iteration takes $N\left[2 + (N_H + 1)m + 2(N_H + 1)^2\right] + (N_H + 1)^3$ operations according to (22). Therefore, the overall computational complexity for Step 3 is $O(I_{HQ}[N(4 + 4N_H + mN_H + 2N_H^2)] + I_{HQ}(N_H + 1)^3)$, where I_{HQ} is the number of iterations of the half-quadratic optimization. Usually, $N \gg N_H$, so the total computational complexity of Algorithm 1 is $O(I_{HQ}N[(4 + m)N_H + 2N_H^2])$.

In contrast, the computational cost of training both the traditional ELM and RELM by the MSE criterion is $O(N(mN_H + 2N_H^2))$ [31]. Moreover, if the appropriate values of the width parameter σ and the regularization parameter λ are chosen, the value of I_{HQ} is usually less than 5. Therefore, RCC may not greatly increase the computational burden in comparison with MSE. In the experiments, we find that ELM trained with RCC is still suitable for tackling the real-time regression and classification tasks.

4 Experimental results

In this section, the performance of ELM-RCC is compared with those of ELM and regularized ELM (RELM) on four synthetic data sets and fourteen benchmark data sets. For the three approaches, that is, ELM, RELM, and ELM-RCC, the weights connecting the input and hidden layers together with the bias terms are randomly generated from the interval $[-1, 1]$. The activation function used in the three methods is the sigmoid function $f(z) = \frac{1}{1+e^{-z}}$.

The error functions for regression problems are all root mean square error (RMSE), while the error functions for classification problems are all error rate. The input vectors of a given data set are scaled to mean zero and unit variance for both the regression and classification problems, while the output vectors are normalized to [0,1] for the regression problems. In addition, all the codes are written in Matlab 7.1.

4.1 Synthetic data sets

In this subsection, two synthetic regression data sets and two synthetic classification data sets are utilized to validate the proposed method. The description of them is as follows.

Sinc: This synthetic data set is generated by the function $y = sinc(x) = \frac{\sin(x)}{x} + \rho$, where ρ is a Gaussian distributed noise. For each noise level, we generate a set of data points $\{(x_i, y_i)\}_{i=1}^{100}$ with x_i drawn uniformly from [−6, 6].

Func: This artificial data set is generated by the function $y(x_1, x_2) = x_1 \exp\{-(x_1^2 + x_2^2)\} + \rho$, where ρ is also a Gaussian distributed noise. For each noise level, 200 data points are constructed by randomly chosen from the evenly spaced 30 × 30 on [−2 ,2].

Two-Moon: There are 200 two-dimensional data points in this synthetic data set. Let $z \sim U(0, \pi)$. Then, the 100 positive samples in the upper moon are generated by the function $\begin{cases} x_1 = \cos z \\ x_2 = \sin z \end{cases}$, while the rest 100 negative samples in the lower moon are generated by $\begin{cases} x_1 = 1 + \cos z \\ x_2 = \frac{1}{2} - \sin z \end{cases}$. To alternate the noise level, different percentages of data points in each class are randomly selected and their class labels are reversed, namely from +1 to −1, while −1 to +1.

Ripley: Ripley's synthetic data set [32] is generated from mixtures of two Gaussian distributions. There are 250 two-dimensional samples in the training set, while 1000 samples in the test set. The manner of changing the noise level of the training samples is the same as that of *Two-Moon*.

The settings of parameters for the three different methods upon the four data sets are summarized in Table 1.

Figure 1 demonstrates the results of the three methods upon *Sinc* with two different noise levels. In Fig. 1a, the regression errors of ELM, RELM, and ELM-RCC are 0.0781, 0.0623, and 0.0581. Moreover, in Fig. 1b, the errors for ELM, RELM, and ELM-RCC are 0.1084, 0.0883, and 0.0791. Therefore, ELM-RCC is more robust against noises on *Sinc* than ELM and RELM.

Figure 2 demonstrates the results of the three approaches upon *Func* with the Gaussian distributed noise $\rho \sim N(0,0.16)$. The regression errors of ELM, RELM, and ELM-RCC are 0.0952, 0.0815, and 0.0775. Then, we can find that ELM-RCC is more robust against noises upon *Func* than ELM and RELM.

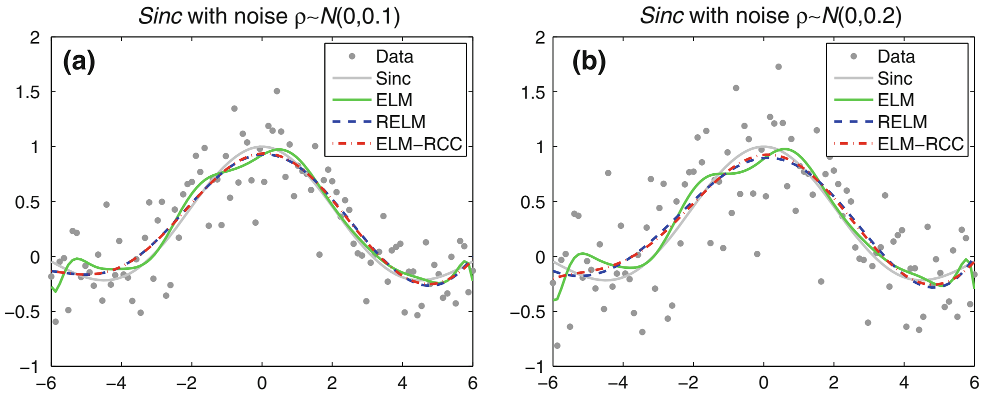

Fig. 1 The regression results of the three methods on *Sinc*. **a** *Sinc* with Gaussian noise $\rho \sim N(0,0.1)$. **b** *Sinc* with Gaussian noise $\rho \sim N(0,0.2)$

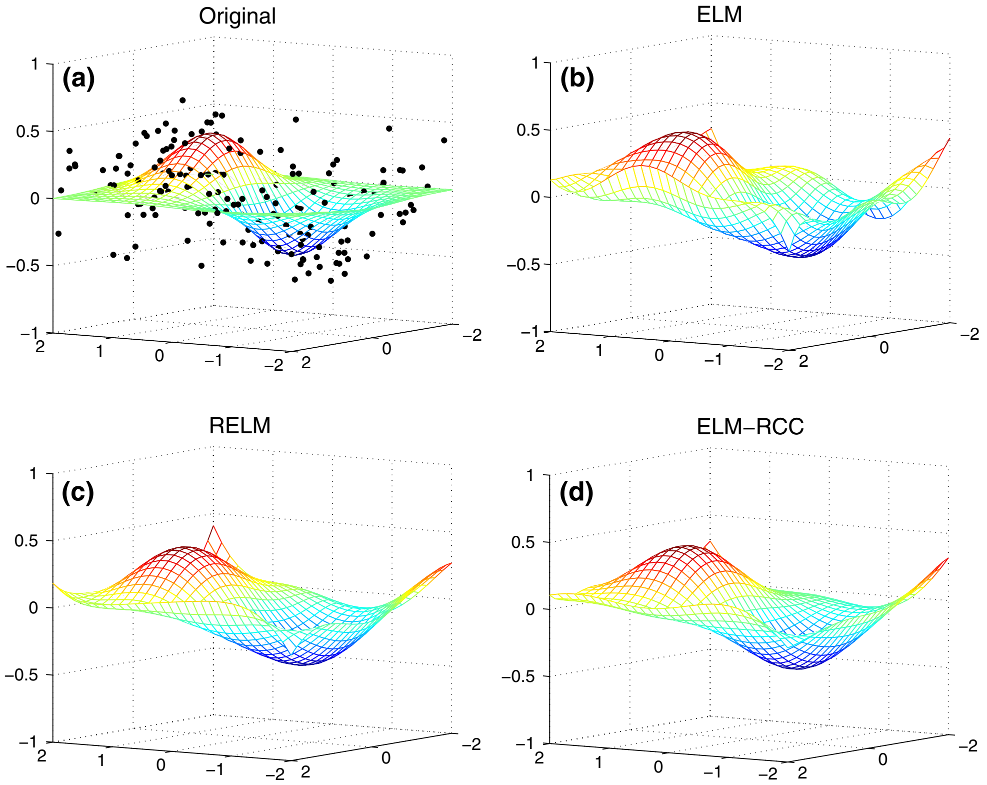

Fig. 2 The regression results of the three methods upon *Func* with Gaussian noise $\rho \sim N(0,0.16)$. **a** The original function and the polluted training samples. **b** The result of ELM. **c** The result of RELM. **d** The result of ELM-RCC

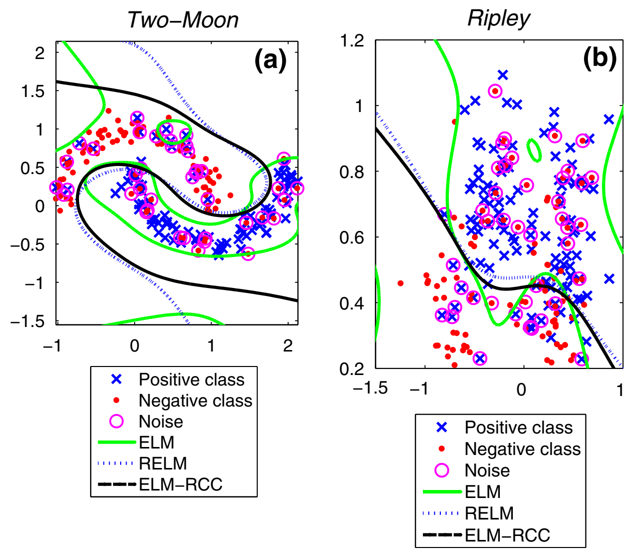

Fig. 3 The classification results of the three criterions upon *Two-Moon* and *Ripley* with noises. **a** *Two-Moon*. **b** *Ripley*

Table 1 Settings of parameters for the three different approaches

Datasets	ELM	RELM		ELM-RCC		
	N_H	N_H	λ	N_H	σ	λ
Sinc	25	25	10^{-7}	25	2	10^{-7}
Func	35	35	10^{-6}	35	3	10^{-6}
Two-Moon	25	25	10^{-3}	25	1	10^{-3}
Ripley	25	25	10^{-2}	25	6	10^{-2}

N_H—Number of hidden units; σ—Width parameter; λ—Regularization parameter

For *Two-Moon*, we randomly choose 20% samples in each class and reverse their class labels to conduct the anti-noise experiment. The classification results of ELM, RELM, and ELM-RCC are illustrated in Fig. 3a. The error rates of ELM, RELM, and ELM-RCC are 3, 2, and 0.5 %. Therefore, ELM-RCC outperforms ELM and RELM upon *Two-Moon* with noises. We can also observe from Fig. 3a that ELM-RCC produces smoother boundary in comparison with ELM and RELM.

Table 2 Benchmark data sets used for regression

Datasets	# Features	Observations	
		# Train	# Test
Servo	5	83	83
Breast heart	11	341	341
Concrete	9	515	515
Wine red	12	799	799
Housing	14	253	253
Sunspots	9	222	50
Slump	10	52	51

# features–Number of features; # train–Number of training samples; # test–Number of testing samples

Table 3 Benchmark data sets used for classification

Datasets	# Features	Observations		# Classes
		# Train	# Test	
Glass	11	109	105	7
Hayes	5	132	28	3
Balance scale	5	313	312	3
Dermatology	35	180	178	6
Vowel	13	450	450	10
Spect heart	45	134	133	2
Wdbc	31	285	284	2

# classes–Number of classes

Toward *Ripley*, 20 % training samples in each class are chosen and their class labels are reversed. The classification results of the three approaches are demonstrated in Fig. 3b. The classification boundaries of ELM-RCC are smoother than those of ELM and RELM. The training error rates of ELM, RELM, and ELM-RCC are 12, 13.6, and 14.8%, while the testing error rates of them are, respectively, 9.8, 9.9, and 9.3 %. Note that the samples in the test set are noise-free. Therefore, ELM-RCC generalizes better than ELM and RELM.

4.2 Benchmark data sets

In the following comparisons, the parameters of the three methods, that is, ELM, RELM, and ELM-RCC are all chosen by the fivefold cross-validation on each training set. The optimal number of hidden units for ELM is chosen from $\{5, 10, \ldots, 100\}$ and $\{5, 10, \ldots, 50\}$ for the regression and classification cases, respectively. For RELM, there are two parameters, that is, the number of hidden units and the penalty parameter λ. The number of hidden units is chosen from the same set as that for ELM. The penalty parameter λ is chosen from {1, 0.5, 0.25, 0.1, 0.075, 0.05, 0.025, 0.01,

Table 4 Setting of the parameters of the three methods on the regression data sets

Datasets	ELM	RELM		ELM-RCC		
	N_H	N_H	λ	N_H	σ	λ
Breast heart	30	20	0.1	50	0.1	0.05
Concrete	35	50	0.1	50	0.25	0.025
Housing	40	45	0.075	45	0.3	0.0005
Servo	35	50	0.0005	45	0.9	0.001
Slump	80	20	0.25	20	0.2	0.075
Sunspots	35	40	0.05	50	0.3	0.025
Wine red	30	40	0.0001	35	0.9	1

0.005, 0.001, 0.0005, 0.0001}. For ELM-RCC, there are three parameters, that is, the number of hidden units, the width parameter σ, and the penalty parameter λ. The number of hidden units is chosen from the same set as that for ELM and the penalty parameter is chosen from the same set as that for RELM. The width parameter σ for the regression tasks and the classification cases are, respectively, chosen from the ranges $\{0.001, 0.005, 0.01, 0.05, 0.25, 0.1, 0.2, \ldots, 1\}$ and $\{15, 14, \ldots, 1, 0.1, 0.25, 0.5, 0.75\}$. Except for *Sunspots*, all the data sets are chosen from the UCI repository of machine learning databases [33]. The descriptions of the regression data sets and the classification data sets are included in Tables 2 and 3, respectively. For *Sunspots* and *Hayes*, their training sets and test sets are fixed. For each of the other data sets, 50 % samples are randomly chosen for training while the rest 50 % samples are used for testing.

Table 5 Setting of the parameters of the three methods on the classification data sets

Datasets	ELM	RELM		ELM-RCC		
	N_H	N_H	λ	N_H	σ	λ
Balance scale	45	15	0.25	45	9	0.1
Dermatology	40	30	0.25	35	7	0.5
Glass	40	50	0.01	45	8	0.075
Hayes	30	50	0.005	35	6	0.0005
Spect heart	5	15	0.5	45	12	0.5
Vowel	50	50	0.01	50	3	0.0001
Wdbc	20	35	1	50	14	0.0005

All the parameters of the three approaches in the experiments for regression and classification tasks are summarized in Tables 4 and 5, respectively.

For the three methods, one hundred trials are conducted on each data set and their corresponding average results are reported. The average training and testing errors together with their corresponding standard deviations for the 100 trails upon the regression data sets are shown in Table 6. Moreover, the average training and testing error rates with their corresponding standard deviations on the classification data sets are included in Table 7.

It is shown in Table 6 that the proposed ELM-RCC outperforms ELM and RELM on all the seven regression data sets. Moreover, the values of standard deviation in Table 6 show that ELM-RCC is more stable than ELM and RELM, even though the value of standard deviation for ELM-RCC is higher than that of RELM upon *Breast heart*,

Table 6 The results of the three methods on the regression data sets

Datasets	ELM		RELM		ELM-RCC	
	E_{train} (Mean ± SD)	E_{test} (Mean ± SD)	E_{train} (Mean ± SD)	E_{test} (Mean ± SD)	E_{train} (Mean ± SD)	E_{test} (Mean ± SD)
Breast heart	0.2937 ± 0.0262	0.3487 ± 0.0320	0.3100 ± 0.0291	0.3490 ± 0.0327	0.3185 ± 0.0387	0.3387 ± 0.0371
Concrete	8.1769 ± 0.3445	8.9393 ± 0.4460	7.5839 ± 0.3171	8.4891 ± 0.3822	7.4522 ± 0.3323	8.4795 ± 0.3839
Housing	3.7983 ± 0.3211	5.1187 ± 0.4737	3.6908 ± 0.3141	4.9211 ± 0.4148	3.6699 ± 0.3964	4.8779 ± 0.4335
Servo	0.4440 ± 0.0865	0.8768 ± 0.1894	0.4028 ± 0.0861	0.7849 ± 0.1289	0.4434 ± 0.0895	0.7757 ± 0.1119
Slump	0.0000 ± 0.0000	3.9376 ± 0.8711	2.5592 ± 0.3338	3.4182 ± 0.5297	2.3035 ± 0.3393	3.2873 ± 0.5018
Sunspots	11.3481 ± 0.3723	22.0417 ± 1.7435	11.3353 ± 0.3141	21.2425 ± 1.2183	10.8793 ± 0.2328	21.1526 ± 1.1702
Wine red	0.6277 ± 0.0142	0.6627 ± 0.0161	0.6174 ± 0.0137	0.6646 ± 0.0163	0.6234 ± 0.0132	0.6574 ± 0.0138

E_{train}–Training RMSE; E_{test}–Testing RMSE

Table 7 The results of the three methods on the classification data sets (%)

Datasets	ELM		RELM		ELM-RCC	
	Err_{train} (Mean ± SD)	Err_{test} (Mean ± SD)	Err_{train} (Mean ± SD)	Err_{test} (Mean ± SD)	Err_{train} (Mean ± SD)	Err_{test} (Mean ± SD)
Balance scale	7.22 ± 0.78	10.19 ± 1.07	11.01 ± 0.89	11.54 ± 1.18	9.17 ± 0.62	9.96 ± 0.82
Dermatology	1.36 ± 0.86	4.80 ± 1.61	2.04 ± 0.94	4.86 ± 1.49	1.64 ± 0.87	4.48 ± 1.59
Glass	2.67 ± 1.36	21.33 ± 4.88	1.88 ± 1.23	19.30 ± 4.86	3.95 ± 1.52	17.77 ± 4.22
Hayes	14.73 ± 1.49	30.50 ± 5.89	13.54 ± 0.50	28.36 ± 3.89	13.64 ± 0.91	26.57 ± 3.83
Spect heart	21.02 ± 1.63	21.26 ± 1.58	10.30 ± 2.03	23.08 ± 3.15	17.95 ± 2.34	20.92 ± 2.09
Vowel	20.95 ± 2.00	36.64 ± 2.88	21.25 ± 1.78	36.38 ± 2.66	20.87 ± 2.07	35.88 ± 3.03
Wdbc	2.83 ± 0.88	4.40 ± 1.24	2.44 ± 0.79	4.00 ± 1.29	1.82 ± 0.60	4.16 ± 1.13

Err_{train}–Training error rate; Err_{test}–Testing error rate

Concrete, and *Housing*. Upon the classification data sets, the results in Table 7 demonstrate that ELM-RCC has lower classification error rate except for *Wdbc* and lower standard deviation except for *Dermatology* and *Vowel* in comparison with ELM and RELM.

5 Conclusions

In this paper, a novel criterion for training ELM, namely regularized correntropy criterion is proposed. The proposed training method can inherit the advantages of both ELM and the regularized correntropy criterion. Experimental results on both synthetic data sets and benchmark data sets indicate that the proposed ELM-RCC has higher generalization performance and lower standard deviation in comparison with ELM and the regularized ELM.

To make our method more promising, there are four tasks for future investigation. First, it is a tough issue to find the appropriate parameters for ELM-RCC. Finding the influences of parameters on the model remains further study. Second, the optimization method for ELM-RCC needs to be improved to avoid the unfavorable influence of the singularity of generalized inverse. Third, the ELM-RCC ensemble based on fuzzy integral [34] will be investigated to further improve the generalization ability of ELM-RCC. Fourth, our future work will also focus on the localized generalization error bound [35] of ELM-RCC and the comparison of the prediction accuracy between ELM-RCC and rule-based systems [36, 37] together with their refinements [38, 39].

Acknowledgments This work is partly supported by the National Natural Science Foundation of China (Nos. 60903089; 61075051; 61073121; 61170040) and the Foundation of Hebei University (No. 3504020). In addition, the authors would like to express their special thanks to the referees for their valuable comments and suggestions for improving this paper.

Appendix 1

From the optimization problem (8), we can obtain that

$$\begin{aligned}&\min_{\boldsymbol{\beta}}(\|\mathbf{H}\boldsymbol{\beta}-\mathbf{T}\|_F^2+\lambda\|\boldsymbol{\beta}\|_F^2)\\&=\min_{\boldsymbol{\beta}}[Tr((\mathbf{H}\boldsymbol{\beta}-\mathbf{T})^T(\mathbf{H}\boldsymbol{\beta}-\mathbf{T}))+\lambda Tr(\boldsymbol{\beta}^T\boldsymbol{\beta})]\\&=\min_{\boldsymbol{\beta}}[Tr(\boldsymbol{\beta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\beta}-\boldsymbol{\beta}^T\mathbf{H}^T\mathbf{T}-\mathbf{T}^T\mathbf{H}\boldsymbol{\beta}+\mathbf{T}^T\mathbf{T})+\lambda Tr(\boldsymbol{\beta}^T\boldsymbol{\beta})],\end{aligned}\tag{24}$$

where $Tr(\cdot)$ denotes the trace operator. Taking the derivative of the objective function in (24) with respect to the weight matrix $\boldsymbol{\beta}$ and setting it equal to zero, we can get

$$2\mathbf{H}^T\mathbf{H}\hat{\boldsymbol{\beta}}-2\mathbf{H}^T\mathbf{T}+2\lambda\hat{\boldsymbol{\beta}}=0.\tag{25}$$

Therefore, the optimal solution (7) can be obtained

$$\hat{\boldsymbol{\beta}}=\left(\mathbf{H}^T\mathbf{H}+\lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{T}.\tag{26}$$

Appendix 2

From the optimization problem (18), we can get

$$\begin{aligned}\hat{J}_{L2}(\tilde{\boldsymbol{\beta}},\boldsymbol{\alpha})&=\max_{\tilde{\boldsymbol{\beta}},\boldsymbol{\alpha}}\left[\sum_{p=1}^{N}\left(\alpha_p\frac{\|\mathbf{t}_p-\sum_{j=0}^{N_H}h_{pj}\tilde{\boldsymbol{\beta}}_j\|_2^2}{2\sigma^2}+\varphi(\alpha_p)\right)\right.\\&\qquad\left.-\lambda\|\tilde{\boldsymbol{\beta}}\|_F^2\right]\\&\Longleftrightarrow\max_{\tilde{\boldsymbol{\beta}},\boldsymbol{\alpha}}\left[\sum_{p=1}^{N}\left(\alpha_p(\mathbf{t}_p-\mathbf{y}_p)^T(\mathbf{t}_p-\mathbf{y}_p)\right)\right.\\&\qquad\left.-\lambda Tr(\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}})+const\right].\end{aligned}\tag{27}$$

The optimal solution of (27) can be derived by (20) and

$$\begin{aligned}\tilde{\boldsymbol{\beta}}^{\tau+1}&=\arg\max_{\tilde{\boldsymbol{\beta}}}\left[Tr\left((\mathbf{T}-\tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}})^T\boldsymbol{\Lambda}(\mathbf{T}-\tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}})-\lambda\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}}\right)\right]\\&=\arg\max_{\tilde{\boldsymbol{\beta}}}\left[Tr(\mathbf{T}^T\boldsymbol{\Lambda}\mathbf{T}-\mathbf{T}^T\boldsymbol{\Lambda}\tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}}-\tilde{\boldsymbol{\beta}}^T\mathbf{H}^T\boldsymbol{\Lambda}\mathbf{T}\right.\\&\qquad\left.+\tilde{\boldsymbol{\beta}}^T\tilde{\mathbf{H}}^T\boldsymbol{\Lambda}\tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}}-\lambda\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}})\right].\end{aligned}\tag{28}$$

Taking the derivation of the objective function in (28) with respect to the weight matrix $\tilde{\boldsymbol{\beta}}$ and letting it equal to zero, we can obtain

$$2\tilde{\mathbf{H}}^T\boldsymbol{\Lambda}\tilde{\mathbf{H}}\tilde{\boldsymbol{\beta}}-2\lambda\tilde{\boldsymbol{\beta}}-2\tilde{\mathbf{H}}^T\boldsymbol{\Lambda}\mathbf{T}=0.\tag{29}$$

Therefore, the solution (22) for the ($\tau + 1$)th iteration can be obtained

$$\tilde{\boldsymbol{\beta}}^{\tau+1}=(\tilde{\mathbf{H}}^T\boldsymbol{\Lambda}\tilde{\mathbf{H}}-\lambda\mathbf{I})^{-1}\tilde{\mathbf{H}}^T\boldsymbol{\Lambda}\mathbf{T}.\tag{30}$$

References

1. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
2. Rao CR, Mitra SK (1971) Generalize inverse of matrices and its application. Wiley, New York
3. Zong WW, Huang GB (2011) Face recognition based on extreme learning machine. Neurocomputing 74:2541–2551
4. Wu J, Wang S, Chung FL (2011) Positive and negative fuzzy rule system, extreme learning machine and image classification. Int J Mach Learn Cybern 2(4):261–271
5. Chacko BP, Vimal Krishnan V, Raju G, Babu Anto P (2012) Handwritten character recognition using wavelet energy and

extreme learning machine. Int J Mach Learn Cybern 3(2): 149–161
6. Zheng WB, Qian YT, Lu HJ (2012) Text categorization based on regularization extreme learning machine. Neural Comput Appl. doi:10.1007/s00521-011-0808-y
7. Wang H, Qian G, Feng XQ (2012) Predicting consumer sentiments using online sequential extreme learning machine and intuitionistic fuzzy sets. Neural Comput Appl. doi:10.1007/s00521-012-0853-1
8. Deng J, Li K, Irwin WG (2011) Fast automatic two-stage nonlinear model identification based on the extreme learning machine. Neurocomputing 74:2422–2429
9. Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
10. Wang YG, Cao FL, Yuan YB (2011) A study of effectiveness of extreme learning machine. Neurocomputing 74:2483–2490
11. Zhao JW, Wang ZH, Park DS (2012) Online sequential extreme learning machine with forgetting mechanism. Neurocomputing 87:79–89
12. Savitha R, Suresh S, Sundararajan N (2012) Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems. Inform Sci 187:277–290
13. Zhang R, Lan Y, Huang GB, Xu ZB (2012) Universal approximation of extreme learning machine with adaptive growth of hidden nodes. IEEE Trans Neural Netw Learn Syst 23:365–371
14. Cao JW, Lin ZP, Huang GB (2012) Self-adaptive evolutionary extreme learning machine. Neural Process Lett. doi:10.1007/s11063-012-9236-y
15. Miche Y, Bas P, Jutten C, Simula O, Lendasse A (2008) A methodology for building regression models using extreme learning machine: OP-ELM. In: European symposium on artificial neural networks. Bruges, Belgium, pp 247–252
16. Golub GH, Van Loan CF (1983) Matrix computation. Johns Hopkins University Press, Baltimore
17. Santamaria I, Pokharel PP, Principe JC (2006) Generalized correlation function: definition, properties, and application to blind equalization. IEEE Trans Signal Process 54(6):2187–2197
18. Liu W, Pokharel PP, Principe JC (2007) Correntropy: properties and applications in non-Gaussian signal processing. IEEE Trans Signal Process 55(11):5286–5297
19. Jeong KH, Liu W, Han S, Hasanbelliu E, Principe JC (2009) The correntropy MACE filter. Pattern Recognit 42(5):871–885
20. Yuan X, Hu BG (2009) Robust feature extraction via information theoretic learning. In: Proceedings of the 26th annual international conference on machine learning. ACM, New York, pp 1193–1200
21. Rockfellar R (1970) Convex analysis. Princeton University Press, Princeton
22. He R, Zheng WS, Hu BG (2011) Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell 33(8):1561–1576
23. He R, Hu BG, Zheng WS, Kong XW (2011) Robust principal component analysis based on maximum correntropy criterion. IEEE Trans Image Process 20(6):1485–1494
24. He R, Tan T, Wang L, Zheng WS (2012) L21 Regularized correntropy for robust feature selection. In: Proceedings of IEEE international conference on computer vision and pattern recognition
25. He R, Zheng WS, Hu BG, Kong XW (2011) A regularized correntropy framework for robust pattern recognition. Neural Comput 23(8):2074–2100
26. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B Cybern 42(2):513–529
27. Principe JC, Xu D, Fisher J (2012) Information theoretic learning. In: Haykin S (eds) Unsupervised adaptive filtering. Wiley, New York
28. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
29. Yang S, Zha H, Zhou S, Hu B (2009) Variational graph embedding for globally and locally consistent feature extraction. In: Proceedings of the Europe conference on machine learning, vol 5782, pp 538–553
30. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
31. Zhao GP, Shen ZQ, Man ZH (2011) Robust input weight selection for well-conditioned extreme learning machine. Int J Inform Technol 17(1):1–13
32. Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
33. Frank A, Asuncion A (2010) UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, Irvine, CA. http://archive.ics.uci.edu/ml
34. Wang X, Chen A, Feng H (2011) Upper integral network with extreme learning mechanism. Neurocomputing 74:2520–2525
35. Yeung D, Ng W, Wang D, Tsang E, Wang X (2007) Localized generalization error model. IEEE Trans Neural Netw 18(5): 1294–1305
36. Wang X, Hong JR (1999) Learning optimization in simplifying fuzzy rules. Fuzzy Set Syst 106(3):349–356
37. Wang XZ, Wang YD, Xu XF, Ling WD, Yeung DS (2001) A new approach to fuzzy rule generation: fuzzy extension matrix. Fuzzy Set Syst 123(3):291–306
38. Tsang ECC, Wang X, Yeung DS (2000) Improving learning accuracy of fuzzy decision trees by hybrid neural networks. IEEE Trans Fuzzy Syst 8(5):601–614
39. Wang X, Yeung DS, Tsang ECC (2001) A comparative study on heuristic algorithms for generating fuzzy decision trees. IEEE Trans Syst Man Cybern Part B Cybern 31(2):215–226