Research Article

An Extreme Learning Machine Based on the Mixed Kernel Function of Triangular Kernel and Generalized Hermite Dirichlet Kernel

Senyue Zhang and Wenan Tan

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 211106, China

Correspondence should be addressed to Senyue Zhang; 35487300@qq.com

Received 2 June 2016; Accepted 16 August 2016

Academic Editor: Xing-Gang Yan

Copyright © 2016 S. Zhang and W. Tan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

According to the characteristics that the kernel function of extreme learning machine (ELM) and its performance have a strong correlation, a novel extreme learning machine based on a generalized triangle Hermitian kernel function was proposed in this paper. First, the generalized triangle Hermitian kernel function was constructed by using the product of triangular kernel and generalized Hermite Dirichlet kernel, and the proposed kernel function was proved as a valid kernel function of extreme learning machine. Then, the learning methodology of the extreme learning machine based on the proposed kernel function was presented. The biggest advantage of the proposed kernel is its kernel parameter values only chosen in the natural numbers, which thus can greatly shorten the computational time of parameter optimization and retain more of its sample data structure information. Experiments were performed on a number of binary classification, multiclassification, and regression datasets from the UCI benchmark repository. The experiment results demonstrated that the robustness and generalization performance of the proposed method are outperformed compared to other extreme learning machines with different kernels. Furthermore, the learning speed of proposed method is faster than support vector machine (SVM) methods.

1. Introduction

The kernel extreme learning machine (KELM) is proposed by Huang et al. in 2010 by applying the kernel functions to ELM algorithm [1, 2] and where the random hidden layer feature mapping in ELM is substituted by the kernel mapping [3]. It effectively improves the undesirable generalization performance and stability caused by the stochastic nature of hidden layer output matrix and greatly reduces computational complexity. In KELM, optimization of the number of hidden layer nodes is avoided and the least square optimal solution can be obtained. Compared with SVM [4] and basic ELM [5, 6], it can provide more stable and better generalization performance. Hence, KELM has been widely applied in classification and regression problems [7, 8].

It is well known that the learning ability and generalization performance of ELM mainly depend on the kernel function, and different kernel functions or same kernel function with different parameters has different influence on

the generalization performance. Besides, the time required to search for the optimal kernel parameters is different using various kernel functions. Normally, the selection and optimization of kernel parameters are much tedious and time-consuming [9, 10]. Liu et al. [11] pointed out that the common Gauss kernel function and polynomial kernel function are very sensitive to the changes of kernel parameters, so the selection range of kernel parameters is large with small stepsize leading to high computation complex issue. In order to solve this issue, a series of SVM kernel functions based on generalized orthogonal polynomials have been constructed in recent years [12–16]. Ozer et al. [12] introduced a set of Chebyshev kernel functions derived from the generalized Chebyshev polynomials. The test results show that the generalized Chebyshev kernel approaches the minimum support vector number for classification in general. Furthermore, Zhang et al. constructed a series of novel SVM kernel functions, such as Laguerre kernel function from generalized Laguerre polynomial and Hermite kernel function from

generalized Hermite polynomial [13–15]. Those algorithms can shorten the time of parameter optimization; however, the processing to the parameters of weight function is so simple that the influence of structure information of sample data on the generalization performance is neglected. Tian and Wang [17] verified that Gaussian Hermite kernel function can achieve the highest classification accuracy in the binary classification problem compared to the rest of above orthogonal polynomial kernel functions; however, the efficiency of its training and the robustness are relatively lower.

According to above analysis, based on Hermite orthogonal polynomials, a mixed kernel function called Generalized Triangular Hermite kernel function is constructed by using the product of triangular kernel and generalized Hermite Dirichlet kernel. This kernel function has only one parameter chosen from a small range of integer numbers, thus the parameter optimization is facilitated greatly, and more structure information of sample data is retained. It is proved that generalized triangle Hermite kernel can be used as an allowed kernel function of extreme learning machine in theory. The effectiveness of the proposed method for regression and binary, multiclass classification problems is demonstrated by performing numerical experiments on a number of real-world datasets from the UCI benchmark repository and comparing their results with SVM and other extreme learning machines with different kernels.

2. Kernel Extreme Learning Machine

2.1. Introduction to Kernel Extreme Learning Machine. A training set $\mathcal{X} = \{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in \mathbb{R}^d, \mathbf{t}_i \in \mathbb{R}^m, i = 1, \dots, N\}$ is given, the output function of hidden node is chosen as $G(\mathbf{a}_i, b_i, \mathbf{x})$ with L hidden nodes, and ELM algorithm can be written as follows.

Step 1. Randomly generate hidden node parameters $(\mathbf{a}_i, b_i)$, $i = 1, \dots, L$.

Step 2. Calculate the hidden layer output matrix $\mathbf{H}$ (ensure $\mathbf{H}$ to be full rank):

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} \\ &= \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \cdots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}. \end{aligned} \quad (1)$$

Step 3. Calculate the output weight vector β :

$$\beta = \mathbf{H}^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T}, \quad (2)$$

where λ is the regularization coefficient and $\mathbf{T} = [\mathbf{t}_1 \cdots \mathbf{t}_N]_{m \times N}$.

The output function of ELM is

$$f(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \mathbf{H}^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T}. \quad (3)$$

If the hidden layer feature mapping $\mathbf{h}(\mathbf{x})$ is unknown, a kernel matrix can be defined to replace $\mathbf{H}\mathbf{H}^T$ by using Mercer's condition. Thus, KELM algorithm is generated as follows:

$$\begin{aligned} \Omega_{\text{ELM}} &= \mathbf{H}\mathbf{H}^T : \Omega_{\text{ELM}, i, j} = \mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) \\ &= K(\mathbf{x}_i, \mathbf{x}_j), \\ \mathbf{H}\mathbf{H}^T &= \Omega_{\text{ELM}} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_N, \mathbf{x}_1) & \cdots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}, \\ \mathbf{h}(\mathbf{x}) \mathbf{H}^T &= \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T. \end{aligned} \quad (4)$$

Finally, the output function of KELM is defined as

$$f(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{\lambda} + \Omega_{\text{ELM}} \right)^{-1} \mathbf{T}. \quad (5)$$

2.2. Introduction to the Property of KELM Kernel Function. Some of the well-known common KELM kernel functions are

- (1) polynomial kernel: $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z} + 1)^n$;
- (2) Gaussian kernel: $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2 / 2\sigma^2)$;
- (3) Laplacian kernel: $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\| / \sigma)$.

In addition to the above kernel functions, a new kernel function also can be constructed according to the property of the kernel function.

Property 1. Assume that $K_1(\mathbf{x}, \mathbf{z})$ and $K_2(\mathbf{x}, \mathbf{z})$ are valid kernel functions on $\mathbf{X} \times \mathbf{X}$; then kernel functions $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) + K_2(\mathbf{x}, \mathbf{z})$ and $K(\mathbf{x}, \mathbf{z}) = K_1(\mathbf{x}, \mathbf{z}) \times K_2(\mathbf{x}, \mathbf{z})$ also are valid on $\mathbf{X} \times \mathbf{X}$.

Theorem 1. Let $f : \mathbf{X} \rightarrow \mathbb{R}$ be an integrable bounded continuous function. Then the necessary and sufficient condition for the translation invariant function $K(\mathbf{x}, \mathbf{z}) = f(\mathbf{x} - \mathbf{z})$ to be a kernel function is $f(\mathbf{0}) > 0$ and Fourier transform is

$$F[K](\omega) = (2\pi)^{-n/2} \int_{\mathbb{R}^n} \exp(-i\omega \mathbf{x}) K(\mathbf{x}) d\mathbf{x} \geq 0. \quad (6)$$

Like SVM, a function is an allowed KELM kernel function as long as it satisfies Mercer's condition.

FIGURE 1: The graph of $K_{\text{RBF}}(\mathbf{x}, 0)$ and $K_{\text{mac}}(\mathbf{x}, 0)$, where $\sigma = 1, 2$.

Mercer Theorem [18]. Assume that, for $\mathbf{X} \subset R^n$, $K(\mathbf{x}, \mathbf{z})$ is a continuous symmetric real value function on $\mathbf{X} \times \mathbf{X}$ such that the following integration should always be nonnegative for every $f \in L_2(\mathbf{X})$:

$$\iint_{\mathbf{X} \times \mathbf{X}} K(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z} \geq 0. \text{ Then } K(\mathbf{x}, \mathbf{z}) \text{ must be a valid kernel function.}$$

3. Triangular Hermite Kernel Extreme Learning Machine

3.1. Construction of Triangular Kernel Function. Laplace kernel function $K(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|/\sigma)$ also is a radial basis function. Its classification performance is nearly equivalent to the Gaussian kernel $K_{\text{RBF}}(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/2\sigma^2)$, but it is less sensitive for the changes of parameter σ . The Laplace kernel can be used as an alternative when using the Gaussian becomes too expensive.

While $x \rightarrow 0$, using the Taylor formula, Laplace kernel can be simplified as

$$K_{\text{mac}}(\mathbf{x}, \mathbf{z}) = \left(1 - \frac{\|\mathbf{x} - \mathbf{z}\|}{\sigma}\right) \times \varepsilon \left(1 - \frac{\|\mathbf{x} - \mathbf{z}\|}{\sigma}\right). \quad (7)$$

Figure 1 shows the function curve of (7) and $K_{\text{RBF}}(\mathbf{x}, \mathbf{z})$, when $\sigma = 1, 2$, where $x \in [-7, 7]$ and $z = 0$.

Seen from Figure 1, two types of kernel functions are quite different, and $K_{\text{mac}}(\mathbf{x}, \mathbf{z})$ is relatively less sensitive to changes of the parameter σ . A typical triangle is presented on its function curve; thus, it is called triangular kernel function [19].

Given a sample set $\mathbf{X} = \{\bar{\mathbf{x}}_i\}_{i=1}^N$, where $\bar{\mathbf{x}}$ is the sample mean and N is the number of training samples, in order to further simplify (7), let σ_0 [16] be twice as long as the maximum distance of all sample points to sample mean such

that formula holds with probability one. Thus, the simplified triangular kernel function is obtained:

$$K_{\text{Tri}}(\mathbf{x}, \mathbf{z}) = 1 - \frac{\|\mathbf{x} - \mathbf{z}\|}{\sigma_0}, \quad (8)$$

where $\sigma_0 = 2 \times \arg \max \{\|\mathbf{x} - \bar{\mathbf{x}}\| \leq r, \mathbf{x} \in \mathbf{X}, \bar{\mathbf{x}} = (1/N) \sum_{i=1}^N \mathbf{x}_i\}$.

Equation (8) is a translation invariant function; according to Theorem 1, $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$ is strictly proved to be a valid KELM kernel function in the following proof.

Proof. $K(\mathbf{x}) = f(\mathbf{x}) = 1 - \|\mathbf{x}\|/\sigma_0$; it is well known that $f(\mathbf{x})$ is an integrable bounded continuous function on R^n by function analysis and $f(\mathbf{0}) = 1 > 0$ such that the Fourier transform of $K(\mathbf{x})$ is

$$\begin{aligned} F[K](\omega) &= (2\pi)^{-n/2} \int_{R^n} \exp(-i\omega\mathbf{x}) K(\mathbf{x}) d\mathbf{x} \\ &= (2\pi)^{-n/2} \int_{R^n} \exp(-i\omega\mathbf{x}) \left(1 - \frac{\|\mathbf{x}\|}{\sigma_0}\right) d\mathbf{x} = (2\pi)^{-n/2} \\ &\cdot \int_{-\infty}^{+\infty} \exp(-i\omega\mathbf{x}) \left(1 - \frac{\|\mathbf{x}\|}{\sigma_0}\right) d\mathbf{x} = (2\pi)^{-n/2} \\ &\cdot \left[\int_{-\infty}^{+\infty} \exp(-i\omega\mathbf{x}) \times 1 d\mathbf{x} \right. \\ &\quad \left. - \frac{1}{\sigma_0} \int_{-\infty}^{+\infty} \exp(-i\omega\mathbf{x}) \times \|\mathbf{x}\| d\mathbf{x} \right] = (2\pi)^{-n/2} \\ &\cdot \left[2\pi\delta(\omega) + \frac{2}{\sigma_0\omega^2} \right] \geq 0. \end{aligned} \quad (9)$$

The proof is completed. $\square$

References [12–16] constructed the weighting function as $K_{\text{Gau}}(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/d)$, where d denotes the dimension of vector $\mathbf{x}, \mathbf{z}$, in which since the parameter σ of Gaussian kernel K_{RBF} is directly set with $\sqrt{d/2}$, the data structure information is lost although the parameter optimization is simplified. However, the setting of parameter σ in triangular kernel function proposed in this paper just makes up for the shortcoming of it.

In order to reflect the difference more intuitively between $K_{\text{Gau}}(\mathbf{x}, \mathbf{z})$ and $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$, Figures 2 and 3 show the graph of the above two kernel functions, where $x \in [-0.5, 0.5], [-1, 1], [-1.5, 1.5]$, and $[-2, 2]$, $z = 0.2, 0.4$.

As is shown in Figures 2 and 3, both the vectors $\mathbf{x}$ and $\mathbf{z}$ are one-dimensional ($d = 1$). When $\mathbf{z}$ takes a fixed value, the graph of kernel function is invariably constant, but the graph of kernel function changes as the value interval of $\mathbf{x}$ changes. It is well known that the choice of different kernel functions is to select different criteria to measure the similarity and the degree of similarity [18].

Consequently, for the same point $(\mathbf{x}, \mathbf{z})$, the similarity in different intervals should be different, and the value of $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$ increases as the expansion of the interval, while the value of $K_{\text{Gau}}(\mathbf{x}, \mathbf{z})$ in the four intervals always remains changeless. Therefore, it can be said that the parameter σ_0

FIGURE 2: The graph of $K_{\text{Gau}}(\mathbf{x}, \mathbf{z})$ and $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$, where $z = 0.2$. FIGURE 3: The graph of $K_{\text{Gau}}(\mathbf{x}, \mathbf{z})$ and $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$, where $z = 0.4$.

of kernel function $K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$ is set to retain more distance similarity information of sample data; in addition, its computational cost is very low.

3.2. Construction of Generalized Hermite Dirichlet Kernel. Hermite polynomial [16] is a kind of orthogonal polynomials with respect to the weighting function e^{-x^2} between the intervals $(-\infty, +\infty)$, which is defined as

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2})$$

$$= \sum_{k=0}^{n/2} (-1)^k \frac{n!}{k!(n-2k)!} (2x)^{n-2k}, \quad (10)$$

$$n = 0, 1, 2, \dots$$

It satisfies the orthogonal relationship:

$$\int_{-\infty}^{+\infty} e^{-x^2} H_m(x) H_n(x) dx = \begin{cases} 0, & m \neq n, \\ 2^n n! \sqrt{\pi}, & m = n. \end{cases} \quad (11)$$

It has a recursive relationship:

$$H_0(x) = 1,$$

$$H_1(x) = 2x,$$

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad n = 1, 2, \dots \quad (12)$$

Owing to the orthogonality, variability, and universal approximation function capability of Hermite polynomial, a general Hermite kernel function can be constructed as a good alternative to other common kernel functions (Gaussian kernel, polynomial kernel, etc.). For this purpose, let the scalar variable x be instead of row vector $\mathbf{x}$ and x^{n+1} be substituted as follows correspondingly:

$$x^{n+1} \rightarrow \begin{cases} (\mathbf{x}\mathbf{x}^T)^{(n+1)/2}, & n = 2k + 1, \\ (\mathbf{x}\mathbf{x}^T)^{n/2} \mathbf{x}, & n = 2k, \end{cases} \quad (13)$$

$$k = 0, 1, 2, \dots,$$

where $\mathbf{x}^T$ is the transpose of x^{n+1} .

Therefore, for vector input, it can define the generalized Hermite polynomial as

$$H_0(\mathbf{x}) = 1,$$

$$H_1(\mathbf{x}) = 2\mathbf{x},$$

$$H_{n+1}(\mathbf{x}) = 2\mathbf{x}H_n^T(\mathbf{x}) - 2nH_{n-1}(\mathbf{x}), \quad n = 1, 2, \dots \quad (14)$$

By using generalized Hermite polynomial, this paper defines generalized n th order Hermite Dirichlet kernel [12] as

$$K_{\text{Hem}}(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^n H_i(\mathbf{x}) H_i^T(\mathbf{z}). \quad (15)$$

It can evaluate and verify the Mercer Theorem for $K_{\text{Hem}}(\mathbf{x}, \mathbf{z})$ as follows by assuming that each element is independent from others:

$$\iint_{\mathbf{X} \times \mathbf{X}} K(\mathbf{X}, \mathbf{Z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z}$$

$$= \iint_{\mathbf{X} \times \mathbf{X}} \left[\sum_{i=1}^n H_i(\mathbf{x}) H_i^T(\mathbf{z}) \right] f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z}$$

$$= \sum_{i=1}^n \iint_{\mathbf{X} \times \mathbf{X}} H_i(\mathbf{x}) H_i^T(\mathbf{z}) f(\mathbf{x}) f(\mathbf{z}) d\mathbf{x} d\mathbf{z}$$

$$\begin{aligned}
&= \sum_{i=1}^n \left(\int_{\mathbf{x}} H_i(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \right) \left(\int_{\mathbf{x}} H_i^T(\mathbf{z}) f(\mathbf{z}) d\mathbf{z} \right) \\
&= \sum_{i=1}^n \left(\int_{\mathbf{x}} H_i(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} \right)^2 \geq 0.
\end{aligned} \tag{16}$$

Therefore, $K_{\text{Hem}}(\mathbf{x}, \mathbf{z})$ is a valid KELM kernel, and its kernel parameters can only be natural numbers, which greatly simplifies the selection and optimization of kernel parameters.

3.3. Generalized Triangular Hermite Kernel Extreme Learning Machine. According to Property 1, a new KELM kernel called Generalized Triangular Hermite kernel is constructed, which is the multiplication of triangular kernel and generalized Hermite Dirichlet kernel, which is defined as

$$K_{\text{Tri-H}}(\mathbf{x}, \mathbf{z}) = \left(1 - \frac{\|\mathbf{x} - \mathbf{z}\|}{\sigma_0} \right) \sum_{i=1}^n H_i(\mathbf{x}) H_i^T(\mathbf{z}). \tag{17}$$

Generalized Triangular Hermite kernel combines the advantages of triangular kernel and generalized Hermite Dirichlet kernel, which not only retains more distance similarity information of sample data, but also just chooses natural numbers to its parameter. Accordingly, it can greatly shorten the time of parameter optimization. Although it has two kernel parameters, that can be determined quickly and easily, which greatly reduces parameter optimization cost. The Generalized Triangular Hermite kernel function up to the 3rd order is listed as in Table 1.

Figures 4 and 5 show the Generalized Triangular Hermite kernel output up to the 3rd order at two different coordinate scales of vertical axis, where changes are within the range of $[-2, 2]$ and are fixed at a constant value. Figure 4 shows the kernel function, while Figure 5 shows the value, where the 0th and 1st orders correspond to the left vertical axis, and the 2nd and 3rd orders correspond to the right vertical axis.

Finally, the kernel $K_{\text{Tri-H}}(\mathbf{x}, \mathbf{z})$ to KELM algorithm is introduced; as a result, the Generalized Triangular Hermite kernel extreme learning machine algorithm is obtained as follows.

Given a training set $\mathcal{N} = \{(\mathbf{x}_i, \mathbf{t}_i) \mid \mathbf{x}_i \in R^d, \mathbf{t}_i \in R^m, i = 1, \dots, N\}$, the output function is

$$f(\mathbf{x}) = \begin{bmatrix} K(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ K(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T \left(\frac{\mathbf{I}}{\lambda} + \mathbf{\Omega}_{\text{ELM}} \right)^{-1} \mathbf{T}, \tag{18}$$

where $K(\mathbf{x}_i, \mathbf{x}_j) = (1 - \|\mathbf{x}_i - \mathbf{x}_j\|/\sigma_0) \sum_{k=1}^n H_k(\mathbf{x}_i) H_k^T(\mathbf{x}_j)$.

4. Experiments and Analysis

In order to test the performance of Generalized Triangular Hermite kernel extreme learning machine (Tri-H KELM) algorithms, this section compares it concerning testing accuracy, training time, and regression determination coefficient

FIGURE 4: The Triangular Hermite kernel output $K_{\text{Tri-H}}(\mathbf{x}, \mathbf{z})$ up to the 3rd order: $x \in [-0.5, 0.5]$ and $z = 0.2$.

FIGURE 5: The Triangular Hermite kernel output $K_{\text{Tri-H}}(\mathbf{x}, \mathbf{z})$ up to the 3rd order: $x \in [-0.5, 0.5]$ and $z = 0.4$.

with other various algorithms in bispiral dataset and real-world benchmark regression, binary, multiclass classification datasets. Table 2 lists the various algorithms used in the experiments and the range of corresponding kernel parameter value, which also includes Gaussian kernel (Gauss), polynomial kernel (Poly), Gaussian Hermite kernel (Gau-H) [15] extreme learning machine algorithms, and triangular Hermite kernel support vector machine algorithm (Tri-H SVM). In multiclass simulations, we use the LIBSVM toolbox available and train SVM for each class separately as one versus all. The SVM cost parameter value is 100 in each corresponding experiment. Better test results are given in boldface in Tables 2–9.

TABLE 1: List of the Generalized Triangular Hermite kernel function up to the 3rd order.

n	$K_{\text{Tri}}(\mathbf{x}, \mathbf{z})$
0	$1 \cdot (1 - \ \mathbf{x} - \mathbf{z}\ /\sigma_0)$
1	$(1 + 4\mathbf{x}\mathbf{z}^T) \cdot (1 - \ \mathbf{x} - \mathbf{z}\ /\sigma_0)$
2	$[1 + 4 \cdot (2\ \mathbf{x}\ ^2 - 1)(2\ \mathbf{z}\ ^2 - 1) + 4\mathbf{x}\mathbf{z}^T] \cdot (1 - \ \mathbf{x} - \mathbf{z}\ /\sigma_0)$
3	$[1 + 4(2\ \mathbf{x}\ ^2 - 1)(2\ \mathbf{z}\ ^2 - 1) + 4\mathbf{x}\mathbf{z}^T + 16(2\ \mathbf{x}\ ^2 - 3)(2\ \mathbf{z}\ ^2 - 3) \cdot \mathbf{x}\mathbf{z}^T] \cdot (1 - \ \mathbf{x} - \mathbf{z}\ /\sigma_0)$

TABLE 2: The various algorithms and the range of their kernel parameters.

Algorithm	Kernel parameter	Range	Stepsize
Gauss KELM	σ	1–10	0.2
Poly KELM	n	0–3	1
Gau-H KELM	n	0–3	1
Tri-H KELM	n	0–3	1
Tri-H SVM	n	0–3	1

TABLE 3: The maximum testing rate (%) and the corresponding kernel parameter value.

KELM algorithm	Maximum rate and optimization time
Gauss KELM	92.5 ($T = 7.2384$ s)
Poly KELM	91.75 ($T = 2.6988$ s)
Gau-H KELM	92.25 ($T = 0.9360$ s)
Tri-H KELM	93 ($T = 0.8424$ s)

TABLE 4: Specifications of binary classification cases.

Datasets	Training data	Testing data	Attributes
Liver	230	115	6
Australian	460	230	6
Breast-cancer	200	77	9
Wpbc	128	70	33
Sonar	150	58	60

TABLE 5: Specifications of multiclass classification cases.

Datasets	Training data	Testing data	Attributes	Classes
Iris	100	50	4	3
Glass	142	72	9	6
Wine	118	60	13	8
Vehicle	564	282	18	4
Segment	1540	770	18	7
Waveform	800	400	21	3

4.1. Classification Performance Comparison on Banana Dataset. The Banana dataset is a well-known binary class dataset from the UCI benchmark repository used in many pattern recognition tests. Each vector of the dataset has 2 features. The training set consists of 800 randomly selected samples (400 for (+1) class and 400 for (−1) class); meanwhile, 400 samples are randomly chosen as a testing set, which contains 200 samples of each class. At each trial of simulation, regularization coefficient is assumed. The simulation results, the maximum testing accuracy, and the

FIGURE 6: The boundaries of Gauss KELM algorithms in Banana dataset.

FIGURE 7: The boundaries of Poly KELM algorithms in Banana dataset.

corresponding kernel parameter optimization time are given in Table 3.

Figures 6–9 show the boundaries of different KELM algorithms on benchmark Banana dataset.

Seen from Table 3 and Figures 6–9, Tri-H KELM achieves the highest testing accuracy in Banana dataset compared with Gauss, Poly, and Gau-H KELM. Furthermore, for the process of parameter optimization, the kernel parameter of two types of Hermite KELM algorithms is only taken from 0 to 3, which greatly shortens the time.

TABLE 6: Specifications of multiclass classification cases.

Datasets	Gauss KELM			Poly KELM			Algorithms			Tri-H ₁ KELM			Tri-H SVM		
	Testing rate (%)	Training time (s)	Testing rate (%)	Testing rate (%)	Training time (s)	Testing rate (%)	Testing rate (%)	Training time (s)	Testing rate (%)	Testing rate (%)	Training time (s)	Testing rate (%)	Testing rate (%)	Training time (s)	Testing rate (%)
Liver ($C = 1$)	60 ($\sigma = 7.4$)	0.0037	72.17 ($n = 3$)	59.13 ($n = 0$)	0.0062	73.91 ($n = 0$)	73.91 ($n = 0$)	0.0036	73.91 ($n = 0$)	73.65 ($n = 0$)	0.0027	73.65 ($n = 0$)	73.65 ($n = 0$)	0.5714	73.65 ($n = 0$)
Breast cancer ($C = 0.1$)	81.82 ($\sigma = 4.2$)	0.0016	76.62 ($n = 1$)	79.22 ($n = 0$)	0.0058	81.82 ($n = 0$)	81.82 ($n = 0$)	0.0053	81.82 ($n = 0$)	81.03 ($n = 0$)	0.0061	81.03 ($n = 0$)	81.03 ($n = 0$)	0.6413	81.03 ($n = 0$)
Wpbc ($C = 0.1$)	77.14 ($\sigma = 0.4$)	0.0126	77.14 ($n = 2$)	80 ($n = 1$)	0.003	78.57 ($n = 1$)	78.57 ($n = 1$)	0.0026	78.57 ($n = 1$)	78.03 ($n = 1$)	0.0026	78.03 ($n = 1$)	78.03 ($n = 1$)	0.4638	78.03 ($n = 1$)
Sonar ($C = 1$)	84.48 ($\sigma = 0.6$)	0.001	84.48 ($n = 3$)	81.03 ($n = 2$)	0.0024	84.48 ($n = 1$)	84.48 ($n = 1$)	0.002	84.48 ($n = 1$)	83.12 ($n = 1$)	0.0084	83.12 ($n = 1$)	83.12 ($n = 1$)	0.8127	83.12 ($n = 1$)

Algorithms

Datasets	Gauss KELM		Poly KELM		Algorithms		Tri-H, KELM		Tri-H SVM	
	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)
Iris ($C = 0.1$)	98 ($\sigma = 2$)	0.0005	97 ($n = 4$)	0.0012	98 ($n = 0$)	0.0038	98 ($n = 0$)	0.0007	96 ($n = 0$)	0.073
Glass ($C = 1$)	77.78 ($\sigma = 0.4$)	0.0026	72.22 ($n = 3$)	0.004	73.61 ($n = 0$)	0.0036	76.37 ($n = 0$)	0.0031	74.58 ($n = 0$)	4.54
Wine ($C = 100$)	80 ($\sigma = 4$)	0.0012	96.67 ($n = 1$)	0.0046	97 ($n = 0$)	0.0048	97.48 ($n = 0$)	0.0025	96.44 ($n = 0$)	1.41
Vehicle ($C = 10$)	80.5 ($\sigma = 1.4$)	0.0163	81.21 ($n = 2$)	0.0076	82.62 ($n = 2$)	0.0206	82.62 ($n = 1$)	0.064	81.58 ($n = 1$)	1.82
Segment ($C = 10$)	87.4 ($\sigma = 7$)	0.2002	88.44 ($n = 2$)	0.0834	87.27 ($n = 0$)	0.4636	88.68 ($n = 0$)	0.5265	87.51 ($n = 0$)	14.65
Waveform ($C = 1$)	86.75 ($\sigma = 10$)	0.0361	84.25 ($n = 1$)	0.0254	86.25 ($n = 0$)	0.0438	87.04 ($n = 0$)	0.1266	85.33 ($n = 0$)	2.87

TABLE 8: Specifications of binary classification cases.

Datasets	Training data	Testing data	Attributes
Cloud	682	342	9
Autoprice	106	53	15
Housing	337	169	13
Body fat	168	84	14
NIR	50	10	401

FIGURE 8: The boundaries of Gau-H KELM algorithms in Banana dataset.

4.2. Classification Performance Comparison on UCI Benchmark Datasets. In order to extensively verify the performance of different algorithms, wide types of datasets have been tested in simulations, which include 5 binary classification cases and 6 multiclass classification cases. Most of the datasets are taken from UCI Machine Learning Repository. The specifications of binary classification and multiclass classification cases are listed in Tables 4 and 5. Comparing Tri-H KELM with Gauss, Poly, and Gau-H KELM and Tri-H SVM, experiment results, including the maximum testing accuracy and corresponding kernel parameter value, and the training time are given in Tables 6 and 7.

(1) *Classification Performance Comparison on Binary Classification Cases.* As demonstrated by the simulation results, Tri-H KELM achieves much better generalization performance in binary classification cases than other KELM algorithms (Gauss, Poly, and Gau-H KELM). More importantly, Tri-H KELM has achieved the highest classification accuracy just within a smaller range of integer numbers of kernel parameter, which greatly reduces the parameter optimization time. Besides, it is observed that Tri-H KELM can always achieve comparable performance with Tri-H SVM at much faster learning speed.

(2) *Classification Performance Comparison on Multiclass Classification Cases.* The experiment results infer that Tri-H KELM performs better than other KELM algorithms and Tri-H SVM on average for multiclass classification cases. Its biggest advantage is that the parameter selection time is

FIGURE 9: The boundaries of Tri-H KELM algorithms in Banana dataset.

short, because the best classification performance is achieved with given kernel parameters. A further study reveals that all KELM algorithms have better scalability and run at much faster learning speed than traditional SVM.

4.3. Regression Performance Comparison on UCI Benchmark Datasets. This subsection selects 5 regression cases from UCI Machine Learning Repository; data is described in Table 8. The simulations of different algorithms on all the regression datasets were carried out. The regression performance of ELM was evaluated by the coefficient of determination R^2 which is defined within the interval of $[0, 1]$ and is closer to 1. The simulations results (R^2 and the training time) are given in Table 9.

As Table 9 shows, in comparison to other several KELM algorithms, Tri-H KELM obtains the maximum coefficient of determination in most regression datasets, and it has all achieved the maximum with respect to kernel parameter. It has better generalization performance for regression problem. Note that Tri-H KELM achieves similar regression performance as SVM at much faster learning speeds.

5. Conclusion

In this work, a novel extreme learning machine based on the mixed kernel function of triangular kernel and generalized Hermite Dirichlet kernel (Tri-H KELM) has been put forward, which introduces the triangular Hermite kernel function to kernel extreme learning machine algorithm. Because the presented kernel has only one parameter chosen from a small set of integers, the parameter optimization is facilitated greatly. Besides, more structure information of sample data is retained in the proposed kernel. Numerical experiments have been performed with different algorithms (Tri-H SVM, Gauss, Poly, Gau-H, and Tri-H KELM) on bispiral benchmark dataset and a number of real-world benchmark datasets and their results have been compared with Tri-H SVM and Gauss, Poly, and Gau-KELM for regression and binary, multiclass classification. Comparable

TABLE 9: Performance comparison of deferent algorithms in regression datasets.

Datasets	Algorithms					
	Gauss KELM		Poly KELM		Gau-H KELM	
	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)	Testing rate (%)	Training time (s)
Cloud ($C = 1$)	0.4163 ($\sigma = 2$)	0.0221	0.9646 ($n = 1$)	0.0184	0.9658 ($n = 2$)	0.0379
Autoprice ($C = 1$)	0.8251 ($\sigma = 10$)	0.0016	0.8661 ($n = 2$)	0.0018	0.9183 ($n = 1$)	0.0012
Housing ($C = 1$)	0.8661 ($\sigma = 3$)	0.0051	0.8994 ($n = 3$)	0.0134	0.9141 ($n = 1$)	0.007
Body fat ($C = 0.1$)	0.8936 ($\sigma = 5.6$)	0.0011	0.9791 ($n = 3$)	0.0034	0.9643 ($n = 2$)	0.0014
NIR ($C = 1000$)	0.2784 ($\sigma = 10$)	0.0034	0.9371 ($n = 1$)	0.0049	0.9452 ($n = 0$)	0.0077
					0.9709 ($n = 2$)	0.0364
					0.9194 ($n = 1$)	0.0011
					0.9102 ($n = 1$)	0.0071
					0.9865 ($n = 1$)	0.0013
					0.9614 ($n = 0$)	0.0022
					0.9642 ($n = 2$)	0.0502
					0.9057 ($n = 1$)	0.0616
					0.8969 ($n = 1$)	0.1914
					0.9725 ($n = 1$)	0.0455
					0.9678 ($n = 0$)	0.0638

generalization and robustness performance of the proposed approach with the rest of the methods considered at a much faster learning speed than Tri-H SVM indicate its usefulness and effectiveness. Future work will be on the study of Tri-H ELM in its practical applications.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 985–990, Budapest, Hungary, July 2004.
- [2] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [3] G.-B. Huang, X. Ding, and H. Zhou, "Optimization method based extreme learning machine for classification," *Neurocomputing*, vol. 74, no. 1–3, pp. 155–163, 2010.
- [4] Z.-L. Wu, C.-H. Li, J. K.-Y. Ng, and K. R. P. H. Leung, "Location estimation via support vector regression," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 311–321, 2007.
- [5] I. Marques and M. Graña, "Face recognition with lattice independent component analysis and extreme learning machines," *Soft Computing*, vol. 16, no. 9, pp. 1525–1537, 2012.
- [6] W. B. Zheng, Y. T. Qian, and H. J. Lu, "Text categorization based on regularization extreme learning machine," *Neural Computing and Applications*, vol. 22, no. 3, pp. 447–456, 2013.
- [7] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [8] A. Iosifidis, A. Tefas, and I. Pitas, "Large-scale nonlinear facial image classification based on approximate kernel Extreme Learning Machine," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '15)*, pp. 2449–2453, IEEE, Quebec, Canada, September 2015.
- [9] B. Li, X. Rong, and Y. Li, "An improved kernel based extreme learning machine for robot execution failures," *The Scientific World Journal*, vol. 2014, Article ID 906546, 7 pages, 2014.
- [10] O. F. Alcin, A. Sengur, S. Ghofrani, and M. C. Ince, "GA-SELM: greedy algorithms for sparse extreme learning machine," *Measurement*, vol. 55, pp. 126–132, 2014.
- [11] S. Liu, L. Feng, Y. Xiao, and H. Wang, "Robust activation function and its application: semi-supervised kernel extreme learning method," *Neurocomputing*, vol. 144, no. 20, pp. 318–328, 2014.
- [12] S. Ozer, C. H. Chen, and H. A. Cirpan, "A set of new Chebyshev kernel functions for support vector machine pattern classification," *Pattern Recognition*, vol. 44, no. 7, pp. 1435–1447, 2011.
- [13] R. Zhang, W. J. Wang, Y. D. Zhang et al., "Legendre kernel function for support vector classification," *Computer Science*, vol. 48, no. 36, pp. 50–53, 2012.
- [14] R. Zhang, W. J. Wang, J. Q. Wang et al., "New set of kernel functions based on Laguerre orthogonal polynomial," *Computer Engineering and Applications*, vol. 48, no. 36, pp. 50–53, 2012.
- [15] R. Zhang, H. Gao, and L. W. Zhang, "A new set of hemite kernel functions for support vector machine," *Journal of Shanxi University (Natural Science Edition)*, vol. 35, no. 1, pp. 38–42, 2012.
- [16] M. Tian and W. J. Wang, "A set of improved Hermite Kernel function," *Computer Science*, vol. 41, no. 5, pp. 239–242, 2014.
- [17] M. Tian and W.-J. Wang, "Research on the properties of orthogonal polynomial kernel functions," *Pattern Recognition and Artificial Intelligence*, vol. 27, no. 5, pp. 385–393, 2014.
- [18] C. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, NY, USA, 2007.
- [19] F. Fleuret and H. Sahibi, "Scale-invariance of support vector machines based on the triangular kernel," in *Proceedings of the 3rd International Workshop on Statistical and Computational Theories of Vision*, Nice, France, 2003.