International Journal of Innovative
Computing, Information and Control
Volume **12**, Number **6**, December **2016**

ICIC International ©2016 ISSN 1349-4198
pp. 1893–1904

TRIANGULAR HERMITE KERNEL EXTREME LEARNING MACHINE

Yibo Li, Haixia Zhang and Xiaofei Ji

School of Automation
Shenyang Aerospace University
No. 37, Daoyi South Avenue, Shenyang 110136, P. R. China
liyibo_sau@163.com; 1054050231@qq.com; jixiaofei7804@126.com

Received March 2016; revised July 2016

Abstract. *The triangular Hermite kernel extreme learning machine methodology is presented based on Hermite polynomial. It introduces the triangular Hermite function which has been proved as a valid kernel function into extreme learning machine as kernel function. The most significant advantages of proposed kernel are that it has only one parameter chosen from a small set of natural numbers, thus the parameter optimization is facilitated greatly, and more structure information of sample data is retained. Experiments were performed on bi-spiral benchmark data set as well as a number of regression datasets from the UCI benchmark repository. Similar or better robustness and generalization performance of the proposed method in comparison to other extreme learning machine with different kernels and SVM (Support Vector Machine) methods demonstrates its effectiveness and usefulness.*

Keywords: Kernel extreme learning machine, Kernel parameter, Hermite orthogonal polynomial, Kernel selection, Triangular Hermite kernel function

1. **Introduction.** Combined the learning principle of support vector machine, Huang et al. proposed kernel extreme learning machine (KELM) in 2010 [1], which applied the kernel functions to ELM algorithm [2,3], and where the random hidden layer feature mapping based ELM is substituted by the kernel mapping. 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 and basic ELM, it can provide more stable and better generalization performance. Hence, KELM has been widely applied in classification and regression problems and practical applications [4-14].

It is well known that the learning ability and generalization performance of extreme learning machine mainly depend on the kernel function; different kernel functions or same kernel function with different parameters have different influence on the generalization performance. Besides, the time required for the optimal kernel parameters is different among various kernel functions, which is relevant to the setting of kernel parameters and the properties of kernel function. Normally, the selection and optimization of kernel parameters are much tedious and time-consuming. As an example, [15] 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 step-size leading to high computation complex issues. Quite a few kernel functions have been proposed in the literature to address such problems. Based on orthogonal polynomials, in [16-18], a series of SVM kernel functions is based on generalized orthogonal polynomials, which shortened the time of parameter optimization; however, the processing

to the parameter of weight function (Gaussian kernel function) is so simple that the influence of structure information of sample data on the generalization performance is neglected. [19] further verified that Gaussian Hermite kernel function achieved the highest classification accuracy in the binary classification problem compared to the rest of above orthogonal polynomial kernel functions, but the efficiency of its training is relatively lower and the robustness, and generalization performance of the algorithm have not been tested in the regression problems.

Motivated by [16-19], based on Hermite orthogonal polynomials, a mixed kernel function called triangular Hermite kernel function is constructed by using the product of triangular kernel and generalized Hermite Dirichlet kernel, which has only one parameter chosen from a small range of integer numbers, and thus the parameter optimization is facilitated greatly, and in which more structure information of sample data is retained. It is proved that triangle Hermite kernel can be used as an allowed kernel function of extreme learning machine in theory. The effectiveness of the proposed method for binary classification and regression problems is demonstrated by performing numerical experiments on bi-spiral benchmark data set and a number of real-world datasets from the UCI benchmark repository and comparing their results with SVM and other extreme learning machine with different kernels.

In summary, the main contributions of this paper are highlighted as follows: (a) triangular Kernel function is proposed to retain structure information of sample data; (b) triangular Kernel function is strictly proved to be a valid KELM kernel based on Fourier transform; (c) triangular Hermite kernel function is presented by using the product of triangular kernel and generalized Hermite Dirichlet kernel; (d) the feasibility and effectiveness of proposed method are verified in the binary classification and regression problems by using the MATLAB; (e) the coefficient of determination is the evaluation criterion for regression performance instead of the conventional RMSE.

The paper is organized as follows. In Section 2, we briefly introduce the kernel extreme learning machine and the property of KELM kernel function respectively. The proposed triangular Hermite kernel extreme learning machine is introduced in Section 3. Results and discussion are presented in Section 4. We conclude our work in Section 5 followed by acknowledgment and references.

2. Introduction to KELM and the Property of Its Kernel Function.

2.1. Kernel extreme learning machine. Given a training set $\aleph = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in \mathrm{R}^d, \boldsymbol{t}_i \in \mathrm{R}^m, i = 1, \ldots, N\}$, hidden node output function $G(\boldsymbol{a}_i, b_i, \boldsymbol{x})$, and hidden node number L, ELM algorithm can be written as follows.

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

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

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

Step 3: Calculate the output weight vector β:

$$\beta = \boldsymbol{H}^{\mathrm{T}} \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} \right)^{-1} \boldsymbol{T} \tag{2}$$

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

The output function of ELM is:

$$f(x) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{3}$$

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

$$\boldsymbol{\Omega}_{ELM} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} : \Omega_{ELMi,j} = \boldsymbol{h}(\boldsymbol{x}_i)\cdot\boldsymbol{h}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{4}$$

$$\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{\Omega}_{ELM} = \begin{bmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix} \tag{5}$$

$$\boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\mathrm{T}} = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}} \tag{6}$$

Finally, the output function of KELM is defined as:

$$f(x) = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}} \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{\Omega}_{ELM}\right)^{-1}\boldsymbol{T} \tag{7}$$

2.2. **Introduction to the property of KELM kernel function.** Some of the well-known common KELM kernel functions are: (1) Polynomial Kernel $K(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{z} + 1)^n$, (2) Gaussian Kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right)$, (3) Laplacian Kernel $K(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\right)$.

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

Property 2.1. [20] *Assume that $K_1(\boldsymbol{x}, \boldsymbol{z})$ and $K_2(\boldsymbol{x}, \boldsymbol{z})$ are valid kernel functions on $\boldsymbol{X} \times \boldsymbol{X}$, then kernel function $K(\boldsymbol{x}, \boldsymbol{z}) = K_1(\boldsymbol{x}, \boldsymbol{z}) + K_2(\boldsymbol{x}, \boldsymbol{z})$ and $K(\boldsymbol{x}, \boldsymbol{z}) = K_1(\boldsymbol{x}, \boldsymbol{z}) \times K_2(\boldsymbol{x}, \boldsymbol{z})$ are also valid on $\boldsymbol{X} \times \boldsymbol{X}$.*

Theorem 2.1. [20] *Let $f : \boldsymbol{X} \to \boldsymbol{R}$ be an integrable bounded continuous function. Then the necessary and sufficient condition for the translation invariant function $K(\boldsymbol{x}, \boldsymbol{z}) = f(\boldsymbol{x} - \boldsymbol{z})$ to be a kernel function is: $f(\boldsymbol{0}) > 0$, and Fourier transform $F[\mathrm{K}](\omega) = (2\pi)^{-n/2}\int_{R^n}\exp(-i\omega\boldsymbol{x})K(\boldsymbol{x})d\boldsymbol{x} \geq 0$.*

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

Mercer Theorem [21]. *Assume that for $\boldsymbol{X} \subset \mathrm{R}^n$, $K(\boldsymbol{x}, \boldsymbol{z})$ is a continuous symmetric real value function on $\boldsymbol{X} \times \boldsymbol{X}$ such that the following integration should always be non-negative for every $f \subset L_2(\boldsymbol{X}) : \iint_{\boldsymbol{X}\times\boldsymbol{X}} K(\boldsymbol{x}, \boldsymbol{z})f(\boldsymbol{x})f(\boldsymbol{z})d\boldsymbol{x}d\boldsymbol{z} \geq 0$. Then $K(\boldsymbol{x}, \boldsymbol{z})$ must be a valid kernel function.*

3. Triangular Hermite Kernel Extreme Learning Machine.

3.1. The construction of triangular kernel function.

Laplace kernel function

$$K(\boldsymbol{x},\boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\right)$$

is also a radial basis function, which is nearly equivalent to the Gaussian kernel

$$K_{RBF}(\boldsymbol{x},\boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{2\sigma^2}\right)$$

about classification performance, but is less sensitive for changes in the sigma parameter σ. The laplace kernel can be used as an alternative when using the Gaussian becomes too expensive.

While $x \to 0$, using the Taylor formula $e^{-x} \approx 1 - x$, Laplace kernel can be simplified as:

$$K_{mac}(\boldsymbol{x},\boldsymbol{z}) = \left(1 - \frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\right) \times \varepsilon\left(1 - \frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma}\right) \tag{8}$$

Figure 1 shows the function curve of Formula (8) and $K_{RBF}(\boldsymbol{x},\boldsymbol{z})$ when $\sigma = 1, 2$, where $x \in [-7, 7]$ and $z = 0$. Seen from Figure 1, two types of kernel functions are quite different, $K_{mac}(\boldsymbol{x},\boldsymbol{z})$ is relatively less sensitive to changes in the parameters σ, and a typical triangle is presented on its function curve, and thus, it is called triangular kernel function [22].

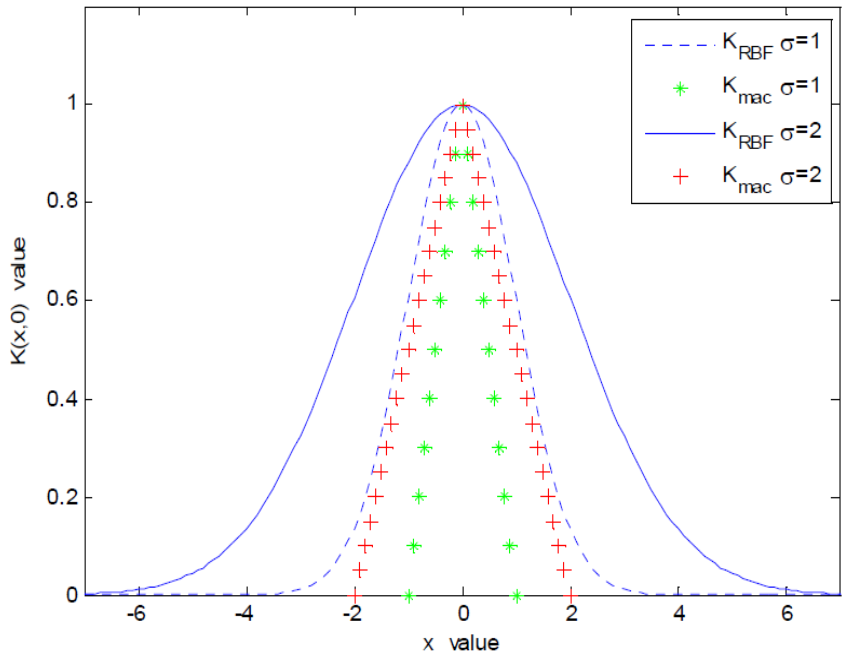


FIGURE 1. The graph of $K_{RBF}(\boldsymbol{x},0)$ and $K_{mac}(\boldsymbol{x},0)$, where $\sigma = 1, 2$

Given a sample set $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, where $\overline{\boldsymbol{x}}$ is the sample mean and N is the number of training samples, in order to further simplify Formula (8), let σ_0 be twice as long as the maximum distance of all sample points to sample mean such that formula $\frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma_0} < 1$ holds with probability one. Thus the simplified triangular kernel function is obtained:

$$K_{Tri}(\boldsymbol{x},\boldsymbol{z}) = 1 - \frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma_0} \tag{9}$$

where $\sigma_0 = 2 \times \arg\max \left\{ \|\boldsymbol{x} - \overline{\boldsymbol{x}}\| \le r, \boldsymbol{x} \in \boldsymbol{X}, \overline{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \right\}$.

Formula (9) is a translation invariant function, according to Theorem 2.1, $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$ is strictly proved to be a valid KELM kernel function in the following proof.

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

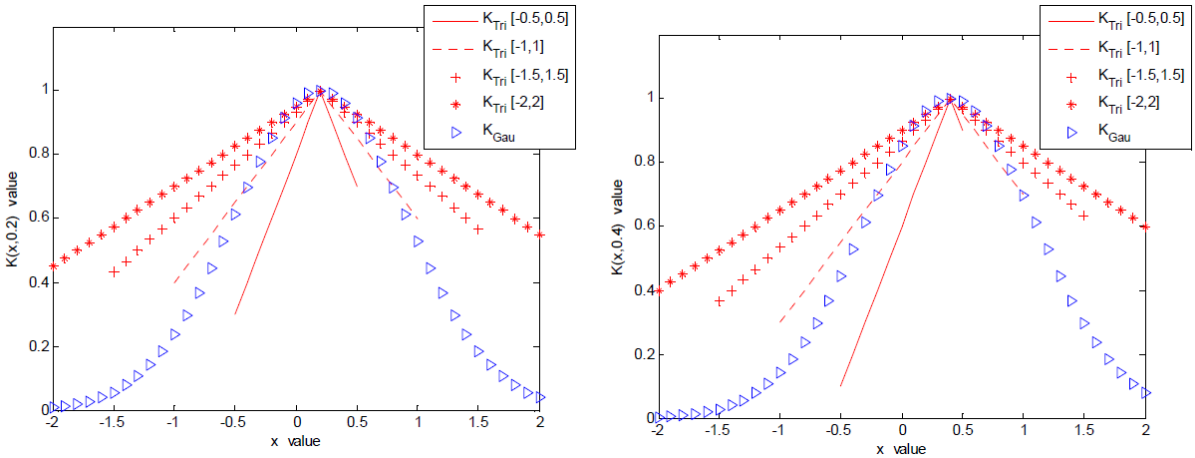
$$\begin{aligned} F[\mathrm{K}](\omega) &= (2\pi)^{-n/2} \int_{R^n} \exp(-i\omega\boldsymbol{x}) K(\boldsymbol{x}) d\boldsymbol{x} \\ &= (2\pi)^{-n/2} \int_{R^n} \exp(-i\omega\boldsymbol{x}) \left(1 - \frac{\|\boldsymbol{x}\|}{\sigma_0}\right) d\boldsymbol{x} \\ &= (2\pi)^{-n/2} \int_{-\infty}^{+\infty} \exp(-i\omega\boldsymbol{x}) \left(1 - \frac{\|\boldsymbol{x}\|}{\sigma_0}\right) d\boldsymbol{x} \\ &= (2\pi)^{-n/2} \left[\int_{-\infty}^{+\infty} \exp(-i\omega\boldsymbol{x}) \times 1 d\boldsymbol{x} - \frac{1}{\sigma_0} \int_{-\infty}^{+\infty} \exp(-i\omega\boldsymbol{x}) \times \|\boldsymbol{x}\| d\boldsymbol{x} \right] \\ &= (2\pi)^{-n/2} \left[2\pi\delta(\omega) + \frac{2}{\sigma_0 \omega^2} \right] \ge 0 \end{aligned} \tag{10}$$

The proof is completed.

[16-19] constructed the weighting function as $K_{Gau}(\boldsymbol{x}, \boldsymbol{z}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{z}\|^2}{d}\right)$, where d denotes the dimension of vector $\boldsymbol{x}$, $\boldsymbol{z}$. In it 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 $K_{Gau}(\boldsymbol{x}, \boldsymbol{z})$ and $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$, Figure 2 shows the graph of 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 Figure 2, where both the vector x and z are one-dimensional ($d = 1$), when z takes a fixed value, the graph of kernel function $K_{Gau}(\boldsymbol{x}, \boldsymbol{z})$ is invariably constant, but the graph of kernel function $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$ changes as the value interval of x changes.



(a) $z = 0.2$ (b) $z = 0.4$

FIGURE 2. The graph of $K_{Gau}(\boldsymbol{x}, \boldsymbol{z})$ and $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$, $x \in [-0.5, 0.5]$, $[-1, 1]$, $[-1.5, 1.5]$ and $[-2, 2]$

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 [19]. Consequently, for the same point (x, z), the similarity in different intervals should be different, the value of $K_{Tri}(x, z)$ increases as the expansion of the interval, while the value of $K_{Gau}(x, z)$ in the four intervals always remain changeless. Therefore, it can be said that the parameter σ_0 of kernel function $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$ is set to retain more distance similarity information of sample data; in addition, its computational cost is very low.

3.2. **The construction of generalized Hermite Dirichlet kernel.** Hermite polynomial [23] is a kind of orthogonal polynomials with respect to the weighting function e^{-x^2} between the intervals $(-\infty, +\infty)$, which is defined as:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}\left(e^{-x^2}\right) = \sum_{k=0}^{\frac{n}{2}} (-1)^k \frac{n!}{k!(n-2k)!} (2x)^{n-2k}, \quad n = 0, 1, 2, \ldots \tag{11}$$

It satisfies the orthogonal relationship:

$$\int_{-\infty}^{+\infty} e^{-x^2} H_m(x) H_n(x) dx = \begin{cases} 0, & m \neq n \\ 2^n n! \sqrt{\pi}, & m = n \end{cases} \tag{12}$$

and has a recursive relationship:

$$H_0(x) = 1, \quad H_1(x) = 2x \tag{13.1}$$

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), n = 1, 2, \ldots \tag{13.2}$$

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 and Polynomial kernel etc.). For this purpose, let the scalar variable x be replaced by row vector $\boldsymbol{x}$, and x^{n+1} be substituted as follows correspondingly:

$$x^{n+1} \to \begin{cases} \left(\boldsymbol{x}\boldsymbol{x}^T\right)^{n+1/2}, & n = 2k+1 \\ \left(\boldsymbol{x}\boldsymbol{x}^T\right)^{n/2} \boldsymbol{x}, & n = 2k \end{cases}, \quad k = 0, 1, 2, \ldots \tag{14}$$

where $\boldsymbol{x}^{\mathrm{T}}$ is the transpose of the $\boldsymbol{x}$.

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

$$\begin{aligned} &H_0(\boldsymbol{x}) = 1, \quad H_1(\boldsymbol{x}) = 2\boldsymbol{x} \\ &H_{n+1}(\boldsymbol{x}) = 2\boldsymbol{x}H_n^{\mathrm{T}}(\boldsymbol{x}) - 2nH_{n-1}(\boldsymbol{x}), \quad n = 1, 2, \ldots \end{aligned} \tag{15}$$

By using generalized Hermite polynomial, this paper defines generalized nth order Hermite Dirichlet kernel [16] as:

$$K_{Hem}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{n} H_i(\boldsymbol{x}) H_i^{\mathrm{T}}(\boldsymbol{z}) \tag{16}$$

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

$$\begin{aligned} \iint_{\boldsymbol{X}\times\boldsymbol{X}} K(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{x}) f(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{z} &= \iint_{\boldsymbol{X}\times\boldsymbol{X}} \left[\sum_{i=1}^{n} H_i(\boldsymbol{x}) H_i^{\mathrm{T}}(\boldsymbol{z})\right] f(\boldsymbol{x}) f(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{z} \\ &= \sum_{i=1}^{n} \iint_{\boldsymbol{X}\times\boldsymbol{X}} H_i(\boldsymbol{x}) H_i^{\mathrm{T}}(\boldsymbol{z}) f(\boldsymbol{x}) f(\boldsymbol{z}) d\boldsymbol{x} d\boldsymbol{z} \end{aligned}$$

$$= \sum_{i=1}^{n} \left(\int_{\boldsymbol{X}} H_i(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} \right) \left(\int_{\boldsymbol{X}} H_i^{\mathrm{T}}(\boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} \right)$$
$$= \sum_{i=1}^{n} \left(\int_{\boldsymbol{X}} H_i(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} \right)^2 \geq 0 \tag{17}$$

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

3.3. **Triangular Hermite kernel extreme learning machine.** According to Property 2.1, a new KELM kernel called triangular Hermite kernel is constructed which is the multiplication of triangular kernel and generalized Hermite Dirichlet kernel, which is defined as:

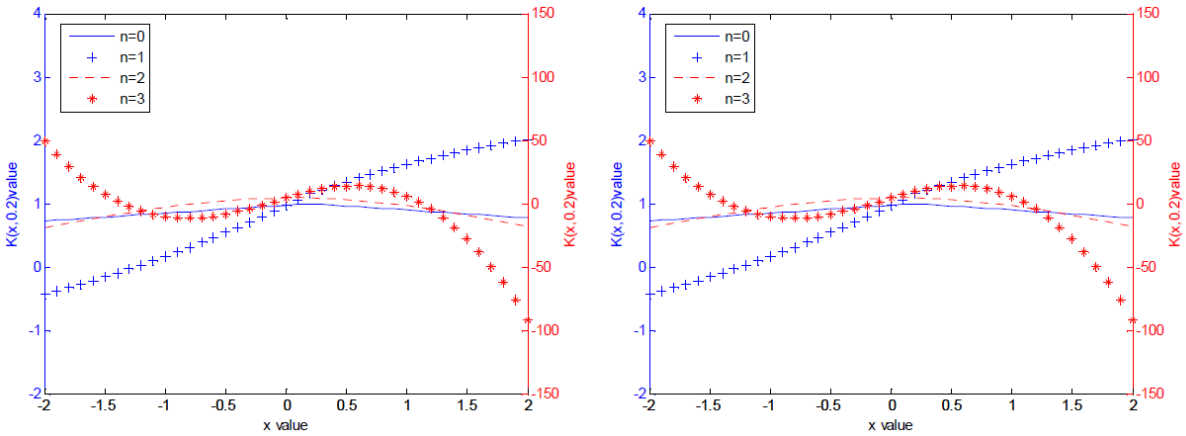
$$K_{Tri\_H}(\boldsymbol{x}, \boldsymbol{z}) = \left(1 - \frac{\|\boldsymbol{x} - \boldsymbol{z}\|}{\sigma_0} \right) \sum_{i=1}^{n} H_i(\boldsymbol{x}) H_i^{\mathrm{T}}(\boldsymbol{z}) \tag{18}$$

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 just chooses natural numbers to its parameter n. Accordingly, it can greatly shorten the time of parameter optimization. Although it has two kernel parameters, they can be determined quickly and easily, which greatly reduces parameter optimization cost.

Figure 3 shows the triangular Hermite kernel output $K_{Tri\_H}(\boldsymbol{x}, \boldsymbol{z})$ up to 3rd order at two different coordinate scales of vertical axis, where $\boldsymbol{x}$ changes within the range of $[-2, 2]$

TABLE 1. List of the triangular Hermite kernel function up to 3rd order

n	Triangular Hermite kernel function $K_{Tri}(\boldsymbol{x}, \boldsymbol{z})$
0	$1 \cdot (1 - \|\boldsymbol{x} - \boldsymbol{z}\|/\sigma_0)$
1	$\left(1 + 4\boldsymbol{x}\boldsymbol{z}^{\mathrm{T}}\right) \cdot (1 - \|\boldsymbol{x} - \boldsymbol{z}\|/\sigma_0)$
2	$\left[1 + 4 \cdot \left(2\|\boldsymbol{x}\|^2 - 1\right)\left(2\|\boldsymbol{z}\|^2 - 1\right) + 4\boldsymbol{x}\boldsymbol{z}^{\mathrm{T}}\right] \cdot (1 - \|\boldsymbol{x} - \boldsymbol{z}\|/\sigma_0)$
3	$\left[1 + 4\left(2\|\boldsymbol{x}\|^2 - 1\right)\left(2\|\boldsymbol{z}\|^2 - 1\right) + 4\boldsymbol{x}\boldsymbol{z}^{\mathrm{T}} + 16\left(2\|\boldsymbol{x}\|^2 - 3\right)\left(2\|\boldsymbol{z}\|^2 - 3\right) \cdot \boldsymbol{x}\boldsymbol{z}^{\mathrm{T}}\right] \cdot \left(1 - \frac{\|\boldsymbol{x}-\boldsymbol{z}\|}{\sigma_0}\right)$



(a) $z = 0.2$ (b) $z = 0.4$

FIGURE 3. The triangular Hermite kernel output $K_{Tri\_H}(\boldsymbol{x}, \boldsymbol{z})$ up to 3rd order: $x \in [-2, 2]$

and $\boldsymbol{z}$ is fixed at a constant value. Figure 3(a) shows the kernel function $K_{Tri\_H}(\boldsymbol{x}, 0.2)$, while Figure 3(b) shows the $K_{Tri\_H}(\boldsymbol{x}, 0.4)$ value, where the 0 and 1st order correspond to the left vertical axis, and the 2nd and 3rd order correspond to the right vertical axis.

Finally, this work introduces the kernel $K_{Tri\_H}(\boldsymbol{x}, \boldsymbol{z})$ to KELM algorithm; as a result, the triangular Hermite kernel extreme learning machine algorithm is obtained as follows: Given a training set $\aleph = \left\{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in \mathrm{R}^d, \boldsymbol{t}_i \in \mathrm{R}^m, i = 1, \ldots, N\right\}$, the output function is:

$$f(x) = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^{\mathrm{T}} \left(\frac{\boldsymbol{I}}{\lambda} + \boldsymbol{\Omega}_{ELM}\right)^{-1} \boldsymbol{T} \tag{19}$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 - \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{\sigma_0}\right) \sum_{k=1}^{n} H_k(\boldsymbol{x}_i) H_k^{\mathrm{T}}(\boldsymbol{x}_j)$.

4. **Experiments and Analysis.** In order to test the performance of triangular Hermite kernel extreme learning machine algorithms (Tri_H KELM), this section compares it concerning training accuracy, testing accuracy, parameter optimization time and regression determination coefficient with other various algorithms in bi-spiral dataset and real-world benchmark regression data sets. The SVM cost parameter value is 100. Better test results are given in boldface and the shortest time is in underline on the following tables.

The simulations of [1,4,16-19] on UCI benchmark datasets revealed that the best generalization performance of SVM and ELM with Gaussian Poly kernel is usually achieved in a very narrow range $[0, 10]$ and the SVM with orthogonal polynomial kernel is in the range $[0, 3]$. Hence, 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) [8] extreme learning machine algorithms and triangular Hermite kernel support vector machine algorithm (Tri_H SVM).

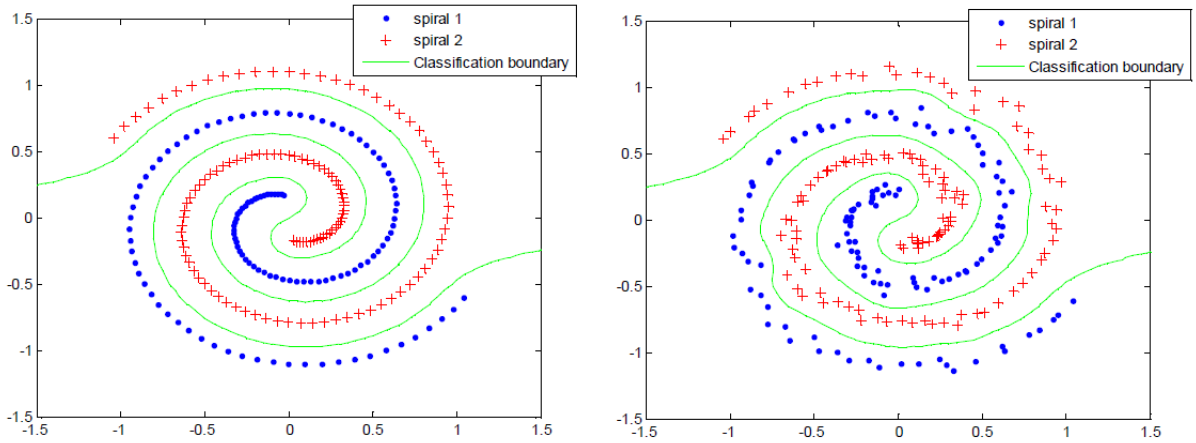
TABLE 2. The various algorithms and the range of its kernel parameters

Algorithms	Kernel Parameter	Range	Step-size
Gauss KELM	σ	0.4∼10	0.2
Poly KELM	n	1∼10	1
Gau_H KELM	n	0∼3	1
Tri_H KELM	n	0∼3	1
Tri_H SVM	n	0∼3	1

The simulations of different algorithms on all the data sets are carried out in MATLAB 7.0.1 environment running in Core(TM) i5-4670K, 3.40-GHZ CPU with 8-GB RAM.

4.1. **Classification performance verification on bi-spiral dataset.** The bi-spiral problem is a typical linear inseparable binary classification problem, which is the touchstone to test the generalization capability of pattern recognition algorithm. In this subsection, the simulations with 2nd order Tri_H KELM have been performed on spiral dataset with noise and no noise respectively. The feature vector of each sample in spiral dataset is 2 dimensional. The dataset has 400 training samples for each experiment, which all contain 200 training samples for each class, while the testing set consists of 200 samples: 100 samples of $(+1)$ class and 100 samples of (-1) class.

The simulation results show that, on spiral dataset with noise, the training accuracy is 100% and the testing accuracy reaches 97%, and when no noise, the training and testing accuracy both achieve 100%.



(a) With no noise (b) With noise

FIGURE 4. The boundary of 2nd order Tri_H KELM on spiral dataset

Figure 4 shows the boundaries of 2nd order Tri_H KELM on spiral dataset with noise and no noise. It can be seen that 2nd order Tri_H KELM can classify two spirals well whether with noise or not.

4.2. **Regression performance comparison on UCI benchmark data sets.** This subsection selects 5 regression cases from UCI Machine Learning Repository, and data is described in Table 3. The simulations of different algorithms on all the regression data sets 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 better the regression performance is. The simulations results (R^2 and the parameter optimization time) are given on Table 4.

TABLE 3. Specifications of regression cases

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

As Table 4 shown, in comparison to several other KELM algorithms, Tri_H KELM obtains the maximum coefficient of determination R^2 in most regression data sets, and for itself, it has all achieved the maximum R^2 with respect to kernel parameter $n \leq 2$. 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, based on Hermite polynomial, a simple novel triangular Hermite kernel extreme learning machine (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. Firstly, the triangular kernel function is constructed, and the validity of it is proved by the Fourier transform of

TABLE 4. Performance comparison of deferent algorithms on regression data sets

Algorithms / Datasets	Gauss KELM		Poly KELM		Gau_H KELM		Tri_H KELM		Tri_H SVM	
	R^2	Parameter Optimization Time(s)	R^2	Parameter Optimization Time(s)	R^2	Parameter Optimization Time(s)	R^2	Parameter Optimization Time(s)	R^2	Parameter Optimization Time(s)
Cloud (C=1)	0.4163 ($\sigma = 2$)	4.3680	0.9646 ($n = 1$)	1.6380	0.9658 ($n = 2$)	0.6540	**0.9709** ($n = 2$)	<u>0.5782</u>	0.9642 ($n = 2$)	5.2008
Autoprice (C=1)	0.8521 ($\sigma = 10$)	0.4356	0.8661 ($n = 2$)	0.2028	0.9183 ($n = 1$)	0.1083	**0.9194** ($n = 1$)	<u>0.0723</u>	0.9057 ($n = 1$)	2.2464
Housing (C=1)	0.8661 ($\sigma = 3$)	1.3572	0.8994 ($n = 3$)	0.5304	**0.9141** ($n = 1$)	0.2340	0.9012 ($n = 1$)	<u>0.2184</u>	0.8969 ($n = 1$)	4.7656
Bodyfat (C=0.1)	0.8936 ($\sigma = 5.6$)	0.5460	0.9791 ($n = 3$)	0.2355	0.9643 ($n = 2$)	0.1417	**0.9865** ($n = 1$)	<u>0.1092</u>	0.9725 ($n = 1$)	2.0455
NIR (C=1000)	0.2784 ($\sigma = 10$)	0.4056	0.9371 ($n = 1$)	$W_{0.1404}$	0.9452 ($n = 0$)	0.0985	0.9614 ($n = 0$)	<u>0.0780</u>	**0.9678** ($n = 0$)	1.1638

translation invariant kernel function. Then we construct a generalized Hermite Dirichlet kernel based on generalized Hermite polynomial, according to Mercer Theorem, it also proves that it can be used as an allowed KELM kernel. Thus, using the product of triangular kernel and generalized Hermite Dirichlet kernel, a mixed kernel function called triangular Hermite kernel function has been constructed and introduced to KELM as a valid kernel. Numerical experiments have been performed with different algorithms (Tri_H SVM, Gauss, Poly, Gau_H and Tri_H KELM) on bi-spiral benchmark data set and a number of real-world benchmark datasets and their results have been compared with Tri_H SVM and Gauss, Poly, Gau_HKELM for regression and binary classification. Comparable 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 indicates its usefulness and effectiveness. Future work will be on the study of Tri_H ELM in its practical applications and multiple-kernel problems [24-26].

Acknowledgment. This work is partially supported by the Program for Liaoning Excellent Talents in University (No. LJQ2014018) and the Scientific Research General Project of Education Department of Liaoning Province, China (No. L2014066). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] G. B. Huang, X. Ding and H. Zhou, Optimization method based extreme learning machine for classification, *Neurocomputing*, vol.74, pp.155-163, 2010.

[2] G. B. Huang, Q. Y. Zhu and C.-K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing*, vol.70, nos.1-3, pp.489-501, 2006.

[3] G. B. Huang, Q. Y. Zhu and C.-K. Siew, Extreme learning machine: A new learning scheme of feed forward neural networks, *Proc. of International Joint Conference on Neural Networks*, pp.985-990, 2004.

[4] G. B. Huang and H. Zhou, Extreme learning machine for regression and multiclass classification, *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.42, no.2, pp.513-529, 2012.

[5] A. Iosifidis, A. Tefas and I. Pitas, Graph embedded extreme learning machine, *IEEE Trans. Cybernetics*, vol.46, no.1, pp.311-324, 2016.

[6] Y. l. He, Z. Q. Geng and Q. X. Zhu, Positive and negative correlation input attributes oriented subnets based double parallel extreme learning machine (PNIAOS-DPELM) and its application to monitoring chemical processes in steady state, *Neurocomputing*, vol.165, no.1, pp.171-181, 2015.

[7] K. D. Brabanter, J. D. Brabanter, J. A. K. Suykens and B. De Moor, Optimized fixed-size kernel models for large data sets, *Computer. Statist. Data Anal.*, vol.54, no.6, pp.1484-1504, 2010.

[8] B. Frenay and M. Verleysen, Using SVMs with randomised feature spaces: An extreme learning approach, *Proc. of the 18th European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp.315-320, 2010.

[9] X. L. Tang and M. Han, Ternary reversible extreme learning machines: The incremental tri-training method for semi-supervised classification, *Knowl. Inf. Syst.*, vol.22, no.3, pp.345-372, 2010.

[10] B. Frenay and M. Verleysen, Parameter-insensitive kernel in extreme learning for non-linear support vector regression, *Neurocomputing*, vol.74, no.16, pp.2526-2531, 2011.

[11] A. A. Mohammed and M. A. Sid-Ahmed, Application of wave atoms decomposition and extreme learning machine for fingerprint classification, *Lect. Notes Comput. Sci.*, vol.6112, pp.246-256, 2010.

[12] R. Minhas, A. Baradarani and S. Seifzadeh, Human action recognition using extreme learning machine based on visual vocabularies, *Neurocomputing*, vol.73, pp.1906-1917, 2010.

[13] V. Malathi, N. S Marimuthu and S. Baskar, Intelligent approaches using support vector machine and extreme learning machine for transmission line protection, *Neurocomputing*, vol.73, pp.2160-2167, 2010.

[14] Y. Lan, Y. C. Soh and G. B. Huang, Two-stage extreme learning machine for regression, *Neurocomputing*, vol.73, pp.3028-3038, 2010.

[15] S. L. Liu, F. Lin, Y. Xiao and H. B. Wang, Robust activation function and its application: Semi-supervised kernel extreme learning method, *Neurocomputing*, 2014.

[16] R. Zhang, W. J. Wang and Y. D. Zhang, Legendre kernel function for support vector classification, *Computer Science*, vol.48, no.36, pp.50-53, 2012.
[17] R. Zhang, W. J. Wang and J. Q. Wang, New set of kernel functions based on Laguerre orthogonal polynomial, *Computer Engineering and Applications*, vol.48, no.36, pp.50-53, 2012.
[18] R. Zhang, H. Gao and L. W. Zhang, A new set of Hemite kernel functions for support vector machine, *Journal of Shanxi University (Nat. Sci. Ed.)*, vol.35, no.1, pp.38-42, 2012.
[19] M. Tian and W. Wang, Research on the properties of orthogonal polynomial kernel functions, *PR & AI*, vol.05, pp.385-393, 2014.
[20] G. S. Wang, Properties and construction methods of kernel in support vector machine, *Computer Science*, vol.6, pp.172-174,178, 2006.
[21] B. Christopher, *Pattern Recognition and Machine Learning*, Springer-Verlag, New York, 2007.
[22] F. Fleuret and H. Sahibi, Scale-invariance of support vector machines based on the triangular kernel, *Proc. of the 3rd International Workshop on Statistical and Computational Theories of Vision*, 2003.
[23] Q. Y. Li, N. C. Wang and D. Y. Yi, *Numerical Analysis*, Tsinghua University Press, Beijing, 2008.
[24] Z. Tang, J. N. Hwang, Y. S. Lin and J. H. Chuang, Multiple-kernel adaptive segmentation and tracking (MAST) for robust object tracking, *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.1115-1119, 2016.
[25] K. H. Lee, J. N. Hwang, G. Okopal and J. Pitton, Ground-moving-platform-based human tracking using visual SLAM and constrained multiple kernels, *IEEE Trans. Intelligent Transportation Systems*, pp.1-11, 2016.
[26] G. V. Karanikolas, G. B. Giannakis, K. Slavakis and R. M. Leahy, Multi-kernel based nonlinear models for connectivity identification of brain networks, *2016 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp.6315-6319, 2016.