Neural Comput & Applic
DOI 10.1007/s00521-013-1392-0

ORIGINAL ARTICLE

Uncertainty evaluation and model selection of extreme learning machine based on Riemannian metric

Wentao Mao · Yanbin Zheng · Xiaoxia Mu · Jinwei Zhao

Received: 4 September 2012 / Accepted: 20 March 2013

Abstract Considering the uncertainty of hidden neurons, choosing significant hidden nodes, called as model selection, has played an important role in the applications of extreme learning machines(ELMs). How to define and measure this uncertainty is a key issue of model selection for ELM. From the information geometry point of view, this paper presents a new model selection method of ELM for regression problems based on Riemannian metric. First, this paper proves theoretically that the uncertainty can be characterized by a form of Riemannian metric. As a result, a new uncertainty evaluation of ELM is proposed through averaging the Riemannian metric of all hidden neurons. Finally, the hidden nodes are added to the network one by one, and at each step, a multi-objective optimization algorithm is used to select optimal input weights by minimizing this uncertainty evaluation and the norm of output weight simultaneously in order to obtain better generalization performance. Experiments on five UCI regression data sets and cylindrical shell vibration data set are conducted, demonstrating that the proposed method can generally obtain lower generalization error than the original ELM, evolutionary ELM, ELM with model selection, and multi-dimensional support vector machine. Moreover, the proposed algorithm generally needs less hidden neurons and computational time than the traditional approaches, which is very favorable in engineering applications.

Keywords Extreme learning machine · Model selection · Multi-objective optimization · Riemannian metric · Uncertainty

1 Introduction

In recent years, extreme learning machines (ELMs), introduced by Huang [1], have become a promising tool in the fields of pattern recognition and regression. As an important branch of neural network, ELMs not only minimize the training error but also seek the smallest norm of output weights [2]. Specifically speaking, while the input weights and hidden layer biases have been generated randomly, ELMs can determine analytically the output weights via a simple matrix inversion procedure. Different from the conventional gradient-based learning algorithms, for example, back-propagation(BP) methods, ELM extends single hidden layer feedforward neural network(SLFN) to "generalized" hidden node case with good generalization performance and very high learning speed [3]. Recently, ELMs have been widely used in solving many real-world problems [4–6].

However, generating hidden neurons in random manner could lead to uncertainty which has aroused wide concern. Here, the uncertainty means the non-optimal or unnecessary input weights and hidden biases generated randomly. Obviously, uncertainty tends to produces redundant hidden nodes in ELM in many cases. Therefore, it is important to select significant hidden nodes for supplying most contribution to improve the generalization performance of ELM. This problem is also called model selection of ELM [7].

W. Mao (✉) · Y. Zheng · X. Mu
College of Computer and Information Engineering,
Henan Normal University, Xinxiang 453007,
People's Republic of China
e-mail: maowt.mail@gmail.com

J. Zhao
State Key Laboratory for Strength and Vibration,
Xi'an Jiaotong University, Xi'an 710049,
People's Republic of China

Published online: 04 April 2013

How to effectively handle uncertainty is a key issue in model selection of ELM. To reach this target, model selection of ELM is generally conducted from two aspects. The first one is optimization strategy which is mainly researched in current study. From the perspective of incremental learning, Feng et al. [8] proposed an error minimized ELM which measured the residual error in an incremental manner. As a follow-up study, Lan et al. [9] introduced a random search method to select most significant nodes. Another kind of typical approach is forward method which selects important hidden nodes from a lot of randomly initialized neurons. Typically, Li et al. [10] used orthogonal least-squares method to evaluate the new neuron's significance without repeating the whole training process. However, in spite of little computational costs, this method is apt to fall into local minima. To uniformly optimize the input weights, biases, and the number of hidden nodes, Zhu et al. [11] proposed a new evolutionary ELM (E-ELM). This algorithm proposed a modified version of differential evolutionary algorithm to find optimal values of input weights and hidden biases while increasing sequentially hidden node. In this algorithm, the fitness function contains RMSE on validation set and the norm of output weight $\|\boldsymbol{\beta}\|$. The second aspect is theoretical evaluation of generalization performance. Based on forward selection, Lan et al. [12] added a refinement stage that used leave-one-out (LOO) error to evaluate the neuron's significance in each backward step. To a certain extent, this method is similar to variable ranking and feature selection. Mao et al. [13] also proposed a simple measure of LOO error via a virtual LOO procedure. This LOO measure can be directly calculated once a training procedure ended. Therefore, the needed computational cost is very low.

However, from the discussion above, the current researches mainly focus on the analysis of generalization performance of whole ELM model instead of specific hidden neurons. Specifically speaking, the above generalization evaluations using RMSE on validation set or LOO error work in a coarse-grained manner, which means, a part of data set cannot provide enough information and microcosmic angle of view to measure the inner structure of ELM accurately. In other words, the above methods cannot provide an accurate statement about the generalization ability and uncertainty of ELM. As a result, the predictive performance of ELM could not be greatly improved, and the optimal number of hidden neurons could not be determined definitively. If the uncertainty of single hidden neuron can be evaluated in microcosmic way, a more efficient ELM model can be obtained. According to the theory of information geometry [14], a nonlinear mapping $\phi(\mathbf{x})$ is a curved submanifold which embeds the input space $S = \{\mathbf{x}\}$ in a high-dimensional Euclidean or Hilbert feature space $F = \{\phi\}$. It provides us a new idea: using Riemannian metric in input space to measure and evaluate the uncertainty of hidden neurons directly. Based on this analysis, this paper proposes a new algorithm for ELM model selection. The contribution of this paper is (1) constructing a new uncertainty evaluation by means of Riemannian metric, and (2) proposing a multi-objective optimization algorithm to find optimal hidden nodes with minimal uncertainty and maximal generalization ability. To our best knowledge, this idea serves as the first attempt to apply information geometry to ELM model selection and uncertainty evaluation.

The rest of this paper is organized as follows. In Sect. 2, a brief review to ELM and a short introduction of Riemannian metric are both given. In Sect. 3, a theoretical proof about uncertainty of hidden neurons in ELM is presented. Section 4 further defines a new uncertainty evaluation and then conducts model selection of ELM via a multi-objective optimization. Section 5 is devoted to numerical experiments, followed by a conclusion of the paper in the last section.

2 Brief summary of ELM and Riemannian metric

2.1 Brief review of ELM

As the theoretical foundations of ELM, Huang et al. [15] studied the learning performance of SLFN on small-size data set and found that SLFN with at most N hidden neurons can learn N distinct samples with zero error by adopting any bounded nonlinear activation function. Then, based on this concept, Huang et al. [3] pointed out that ELM can analytically determine the output weights by a simple matrix inversion procedure as soon as the input weights and hidden layer biases are generated randomly, and then obtain good generalization performance with very high learning speed. Here, a brief summary of ELM is provided.

Given a set of *i.i.d* training samples $\{(\mathbf{x}_1, \mathbf{t}_1), \ldots, (\mathbf{x}_N, \mathbf{t}_N)\} \subset \mathbb{R}^n \times \mathbb{R}^m$, standard SLFNs with $\tilde{N}$ hidden nodes are mathematically formulated as [3]:

$$\sum_{i=1}^{\tilde{N}} \beta_i h_i(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \beta_i g_i(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \mathbf{o}_j, \quad j = 1, \ldots, N \tag{1}$$

where $g(x)$ is activation function, $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is input weight vector connecting input nodes and the ith hidden node, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{im}]^T$ is the output weight vector connecting output nodes and the ith hidden node, b_i is bias of the ith hidden node. Huang et al. [3] have rigorously proved that then for N arbitrary distinct samples and any $(\mathbf{w}_i, b_i)$ randomly chosen from $\mathbb{R}^n \times \mathbb{R}^m$ according

to any continuous probability distribution, the hidden layer output matrix $\mathbf{H}$ of a standard SLFN with N hidden nodes is invertible and $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| = 0$ with probability one if the activation function $g : \mathbb{R} \mapsto \mathbb{R}$ is infinitely differentiable in any interval. Then, given $(\mathbf{w}_i, b_i)$, training a SLFN equals finding a least-squares solution of the following equation [3]:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{2}$$

where:

$$\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, \mathbf{x}_1, \ldots, \mathbf{x}_{\tilde{N}}) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{\tilde{N}}]^T$$

$$\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^T$$

Considering most cases that $\tilde{N} \ll N, \boldsymbol{\beta}$ cannot be computed through the direct matrix inversion. Therefore, Huang et al. [3] calculated the *smallest norm* least-squares solution of Eq. (2):

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} \tag{3}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$. The generalization performance of SLFN will be improved by minimizing training errors as well as the norm of output weights, the solution $\hat{\boldsymbol{\beta}}$ can pledge the generalization ability of SLFN in the theory [2].

Based on the above analysis, Huang et al. [3] proposed ELM whose framework can be stated as follows:

Step 1. Randomly generate input weight and bias $(\mathbf{w}_i, b_i), i = 1, \ldots, \tilde{N}$.

Step 2. Compute the hidden layer output matrix $\mathbf{H}$.

Step 3. Compute the output weight $\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T}$.

Therefore, the output of SLFN can be calculated by $(\mathbf{w}_i, b_i)$ and $\hat{\boldsymbol{\beta}}$:

$$f(\mathbf{x}_j) = \sum_{i=1}^{\tilde{N}} \hat{\beta}_i g_i(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) = \hat{\boldsymbol{\beta}} \cdot h(\mathbf{x}_j) \tag{4}$$

2.2 Brief introduction of Riemannian metric

The pioneer work about information geometry is the study [16] which studied the structure of feature space induced by kernel function from geometrical perspective. Another pioneer work is the study [14] which applied Riemannian metric to construct a data-dependent kernel. The starting point of these works is that the nonlinear mapping function ϕ from Euclidean input space to high-dimensional feature space is a curved submanifold which defines the shape of a surface in this feature space. Therefore, even the kernel mapping ϕ is unknown, we can still measure geodesic distance between two samples in feature space. This channel is Riemannian metric.

Let $\mathbf{z} = \phi(\mathbf{x})$ be the image of sample $\mathbf{x}$ in feature space via mapping function ϕ. Then, a small distance vector $d\mathbf{x}$ in input space can be mapped in feature space as [14]:

$$d\mathbf{z} = \nabla\phi \cdot d\mathbf{x} = \sum_i \frac{\partial}{\partial x_i}\phi(x) d\mathbf{x}_i$$

$d\mathbf{z} = (dz_\alpha)$ is called line element. Then, the squared length of $d\mathbf{z}$ is written as:

$$d\mathbf{z}^2 = \sum_\alpha (dz_\alpha)^2 = \sum_{i,j} g_{ij}(\mathbf{x}) dx_i dx_j \tag{5}$$

where

$$g_{ij}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i}\phi(\mathbf{x})\right) \cdot \left(\frac{\partial}{\partial x_j}\phi(\mathbf{x})\right) \tag{6}$$

In Eq. (6), the dot denotes the summation over index α of mapping ϕ. The $n \times n$ positive-definite matrix $G(x) = (g_{ij}(x))$ is the Riemannian metric tensor induced in input space. $g_{ij}(\mathbf{x})$ in Eq. (6) is defined as Riemannian metric. According to Eq. (5), $g_{ij}(\mathbf{x})$ in the input space shows how a small volume element in the input space is enlarged or reduced in the feature space.

3 Uncertainty evaluation based on Riemannian metric

As stated above, the current researches generally focus on the generalization performance of whole ELM model by calculating LOO error or RMSE on validation set. These studies work macroscopically and have no access to exploit the internal structure of network. In this section, Riemannian metric is introduced to solve this problem. The basic idea is assuming the hidden neurons in ELM locate on a manifold from input space to feature space. Hence, it is possible to evaluate the uncertainty of hidden neurons using Riemannian metric.

This paper mainly deals with regression problem. According to [2, 17], ELM can be optimized via standard regularization like ridge regression, as the following formulation:

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N} \varepsilon_i^2 \\ \text{s. t.} \quad & \varepsilon_i = t_i - \boldsymbol{\beta} \cdot h(\mathbf{x}_i), i = 1, \ldots, N \end{aligned} \tag{7}$$

where C is regularization parameter which controls the trade-off between the training error and generalization ability. Different from support vector machine (SVM), the reason why the above formulation permits the training errors is to eliminate the possible overfitting and further improve the generalization performance [17]. In

geometrical view, the separating hyperplane of ELM basically passes through the origin in the feature space, and the *uncertainty* of hidden neurons essentially indicates how the neurons deviate from this hyperplane.

The basic idea is as follows: in order to decrease the training errors in the feature space without changing the volume of the entire space, it is efficient to reduce volume elements locally in neighborhoods of hidden neurons which are located closely to the decision hyperplane. Considering permitted training errors, we have the following theorem.

Theorem 1 *For regression problem, if the line element* $d\mathbf{z}$ *of ELM's hidden neurons in feature space F decreases, these neurons are closer to the hyperplane* $f(\mathbf{x}) = \boldsymbol{\beta} \cdot h(\mathbf{x})$

Proof For better geometrical illustration, here we assume the ELM mapping $h(\mathbf{x})$ is linear. The following proof can be straightforwardly extended to nonlinear case.

Let $\mathrm{A} = (x_1,y_1)$ and $\mathrm{C} = (x_2,y_2)$ be two hidden neurons in feature space. In the case of linear mapping, the decision hyperplane can be illustrated as Fig. 1. For convenient calculating, we draw a dotted line which is parallel to the line $y = \beta{\cdot}x$ in Fig. 1. Then, AD, denoted by d, is the distance from A and C to hyperplane. AB is parallel to y-axis, and $\mathrm{AB}\perp\mathrm{CE}.\mathrm{AC} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.\mathrm{CE} = |x_1 - x_2|.$ $\mathrm{AE} = |y_1 - y_2|$. According to triangle's property, we have: $\mathrm{AB}{\cdot}\mathrm{CE} = \mathrm{AD}\cdot\mathrm{BC}$, i.e., $\mathrm{AB}{\cdot}|\,x_1 - x_2\,| = \mathrm{BC}{\cdot}d$. According to cosine theorem, we have:

$$\begin{aligned}\mathrm{BC} &= \sqrt{\mathrm{AB}^2 + \mathrm{AC}^2 - 2\mathrm{AB}\cdot\mathrm{AC}\cos\angle\mathrm{BAC}} \\ &= \sqrt{\mathrm{AB}^2 + \mathrm{AC}^2 - 2\mathrm{AB}\cdot\mathrm{AC}\frac{\mathrm{AE}}{\mathrm{AC}}} \\ &= \sqrt{\mathrm{AB}^2 + (x_1 - x_2)^2 + (y_1 - y_2)^2 - 2\mathrm{AB}\cdot|y_1 - y_2|}\end{aligned}$$

Then, we have:

$$\begin{aligned}d^2 &= \frac{AB^2(x_1 - x_2)^2}{\mathrm{AB}^2 + (x_1 - x_2)^2 + (y_1 - y_2)^2 - 2\mathrm{AB}\cdot|y_1 - y_2|} \\ &= \mathrm{AB}^2\left[1 - \frac{(AB - |y_1 - y_2|)^2}{(\mathrm{AB} - |y_1 - y_2|)^2 + (x_1 - x_2)^2}\right]\end{aligned}$$

Obviously, $\mathrm{AB} > \mathrm{AE} = |\,y_1 - y_2\,|$. Therefore, if $(x_1 - x_2)^2$ decreases, the distance from A and C to hyperplane d will decrease correspondingly. Moreover, as line element $d\mathbf{z}$ is the differential of geodesic distance on manifold, there is a positive correlation between $(x_1 - x_2)^2$ and $d\mathbf{z}$. As a result, when $d\mathbf{z}$ decreases, the distance between two neurons will decrease, as well as the distance between neurons and hyperplane.

According to Eq. (5), line element $d\mathbf{z}$ can be calculated by Riemannian metric. Therefore, Theorem 1 permits us to evaluate uncertainty of hidden neurons by using Riemannian metric. Here, we give a new definition for uncertainty of ELM.

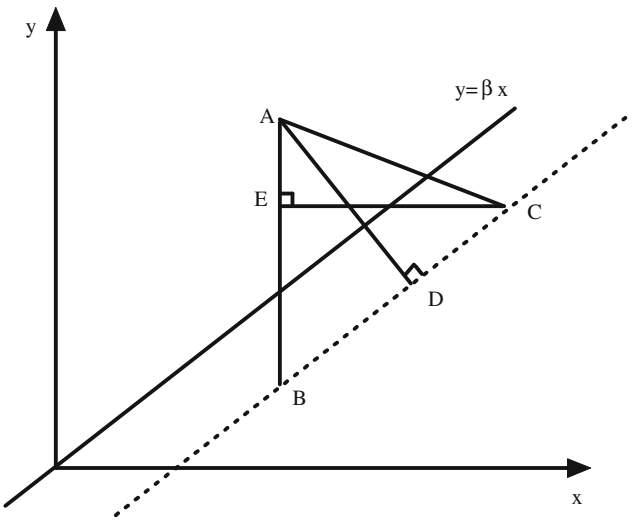


Fig. 1 Sketch diagram of ELM with linear mapping

Definition 1 (*Uncertainty degree*) The uncertainty degree of ELM, denoted by $U(\mathbf{X})$, is defined as:

$$U(\mathbf{X}) = \frac{1}{\tilde{N}}\sum_{l=1}^{\tilde{N}} \|G(\mathbf{x}_l)\|_F \tag{8}$$

where $G(\mathbf{x}) = (g_{ij}(\mathbf{x}))$ is $n \times n$ Riemannian metric tensor, $g_{ij}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i}\phi(\mathbf{x})\right)\cdot\left(\frac{\partial}{\partial x_j}\phi(\mathbf{x})\right)$, $\|\cdot\|_F$ is Frobenius norm, and $\tilde{N}$ is the number of hidden neurons in ELM.

In fact, the distance between neurons and hyperplane can be directly calculated using the distance formula from point to a plane. However, the calculating process is restricted in feature space, and the output variable must be available. On the contrary, the proposed uncertainty degree works in input space, and it does not depend on the output variables. Another advantage is this definition is an easy one to calculate. In Definition 1, $\phi(\cdot)$ can be any nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems [17]. For example, as $\phi(x) = \frac{1}{1+\exp(-x)}$ is widely employed in practice, the partial derivative of $\phi(\cdot)$ as well as $g_{ij}(\mathbf{x})$ can be easily calculated.

4 Model selection based on Riemannian metric

Uncertainty directly affects the generalization ability. Generally speaking, the smaller the uncertainty is, the better generalization performance the ELM tend to have. The ultimate aim of studying uncertainty of ELM is to improve its generalization performance. This section tries to establish a channel from uncertainty to model selection.

4.1 Uncertainty and generalization

The work starts from the following understanding: the uncertainty of ELM can be regarded as the deviation of hidden neurons from boundary surface in geometrical view. Now, we have the following theorem.

Theorem 2 *For regression problems, if hidden neurons are more close to the decision hyperplane* $f(\mathbf{x}) = \boldsymbol{\beta} \cdot h(\mathbf{x})$ *and this hyperplane is more flat, the ELM tends to have better generalization performance.*

Proof The theorem is also proved in linear case for better illustration. Employing the decision hyperplane as boundary surface, all hidden neurons can be divided into two classes. We also use Fig. 1 for demonstration. But here two neurons A and C are placed in the same side. According to Theorem 1, the distance between hidden neurons and hyperplane can be written as:

$$d^2 = \frac{\mathrm{AB}^2(\mathrm{x}_1 - \mathrm{x}_2)^2}{\mathrm{AB}^2 + (x_1 - x_2)^2 + (y_1 - y_2)^2 - 2\mathrm{AB} \cdot |y_1 - y_2|} = \frac{\mathrm{AB}^2}{\frac{(\mathrm{AB} - |y_1 - y_2|)^2}{(x_1 - x_2)^2} + 1}$$

Suppose that all neurons get close to hyperplane at same speed. If d is lessen, $\frac{(\mathrm{AB}-|y_1-y_2|)^2}{(x_1-x_2)^2}$ gets larger. And from Fig. 1, $\mathrm{AB} > \mathrm{AE} = |\, y_1 - y_2 \,|$. Then, the distance between A and C, i.e., $(x_1 - x_2)^2 + (y_1 - y_2)^2$, gets smaller, which means the intra-class distance decreases.

On the other hand, the slope of hyperplane affects the inter-class distance. We also use Fig. 1 to elaborate. In Fig. 1, if the plane becomes more flat, the distance between A and B keeps fixed, but $\angle$DAB gets smaller. Then, $\mathrm{AD} = \mathrm{AB} \cdot cos(\angle \mathrm{DAB})$ gets larger correspondingly, which means the inter-class distance tends to grow larger if the plane becomes more flat. According to Fisher's Rule, with smaller intra-class distance and larger inter-class distance, the generalization ability of ELM tends to be improved.

In fact, Theorem 2 is straightforward. From standard optimization point of view [17], hidden neurons close to hyperplane bring low training errors, and the smaller the norm of $\boldsymbol{\beta}$ is , the better generalization performance of ELM tends to obtain. Note that in geometry, small $\boldsymbol{\beta}$ just leads to flat hyperplane.

Now, we have the following corollary.

Corollary 1 For regression problem, if the uncertainty degree $U(\mathbf{X})$ defined in Eq. (8) and the norm of output weights $\boldsymbol{\beta}$ both get smaller, the generalization performance of ELM will be improved.

Proof By appealing to Theorem 1, Theorem 2, and Definition 1, the statement of the corollary is obviously true.

4.2 Model selection based on multi-objective optimization

To get more effective ELM model, we should minimize the uncertainty degree $U(\mathbf{X})$ and $\|\boldsymbol{\beta}\|$ simultaneously. Note that Corollary 1 also indicates potentially that $U(\mathbf{X})$ and $\|\boldsymbol{\beta}\|$ are somewhat conflicting, which means that the values of $U(\mathbf{X})$ can be increased merely at the cost of worse results of $\|\boldsymbol{\beta}\|$. In multi-objective optimization, a solution is called non-dominated or Pareto-optimal if no objective can be improved without getting worse at least one of the others [18]. As the goal of multi-objective optimization algorithm is to find a set of non-dominated solutions instead of single global best individual as in the single objective problem, the model selection of ELM based on Riemannian metric is naturally transformed into a multi-objective optimization problem.

In this paper, a novel multi-objective PSO, named multi-objective comprehensive learning PSO (MOCLPSO) [18], is chosen as practical realization. Here, a brief summary of MOCLPSO is provided as follows. Based on CLPSO which can use all particles' historical best information effectively, MOCLPSO adopts a crowding distance-based archive maintenance strategy to update the velocity and position of particles. Therefore, diversity of the swarm can be improved to avoid premature convergence efficiently. Its basic idea is described as follows [18]: MOCLPSO uses an external archive B to store the set of non-dominated solutions obtained at each generation and uses an archive A to store the best set found so far, and then compares two sets one by one. If the solution x in B is dominated by a member of A, reject x; If x dominates a subset C of A, then $\mathrm{A} = \mathrm{A} \backslash \mathrm{C}$, $\mathrm{A} = \mathrm{A} \cup \{x\}$.

As stated above, two fitness functions in MOCLPSO are the uncertainty degree $U(\mathbf{X})$ shown in Eq. (8) and the norm of output weights $\|\boldsymbol{\beta}\|$, respectively. Same with [11], the individual in the population consists of a set of input weights and hidden biases:

$$\theta = \left[\omega_{11}, \omega_{12}, \ldots, \omega_{1\tilde{N}}, \omega_{21}, \ldots, \omega_{2\tilde{N}}, \ldots, \omega_{N1}, \ldots, \omega_{N\tilde{N}}, b_1, b_2, \ldots, b_{\tilde{N}}\right] \tag{9}$$

Therefore, model selection of ELM transforms into finding the θ^0 which minimize $U(\mathbf{X})$ and $\|\boldsymbol{\beta}\|$ simultaneously with smallest $\tilde{N}$, and can be described as follows:

$$\theta^0 = \arg\min_{\theta} E(\theta) = \arg\min_{\theta}(U(\mathbf{X}, \theta), \|\boldsymbol{\beta}\|) \tag{10}$$

Another point to be given due consideration is how to choose best one from the obtained set of non-dominated solutions. Here, we introduce LOO bound of ELM proposed in [13], as follows:

$$\mathrm{LOO}_{\mathrm{ELM}} = \frac{1}{N}\sum_{i=1}^{N}\left[e_i^{(-i)}\right]^2 \tag{11}$$

where $e_i^{(-i)} = \mathbf{t}_i - f(\mathbf{x}_i)^{(-1)} = \frac{\alpha_i}{\mathbf{P}_{ii}^{-1}}, \alpha_i$ is Lagrange multiplier in Eq. (7), and $\mathbf{P} = \left[\mathbf{K} + \frac{1}{2C}I\right]$.

This LOO bound is very computationally inexpensive as it can be directly calculated after ELM has been trained. So using this bound as choosing criterion cannot add extra computational cost.

The algorithm is described in three steps, as follows:

Step 1. Initializing the population randomly with one hidden neuron. The individual is generated within the range of $[-1,1]$. Once ELM is trained, calculating $U(\mathbf{X})$ and $\|\boldsymbol{\beta}\|$ for each individual from Eqs. (8) and (3), respectively.

Step 2. Applying MOCLPSO to finding the optimal input weights and biases for each sequentially added hidden neurons. After a set of non-dominated solution has been obtained, the solution with lowest value of LOO bound in Eq. (11) is chosen as the best input weights and biases.

Step 3. Add hidden neurons sequentially. Go to Step 2 or stop if the iteration number reaches maximum or the change of fitness values between two consecutive iterations is less than a threshold.

The flowchart of the proposed model selection method is depicted in Fig. 2.

5 Experimental results

In this section, two kinds of data sets are introduced to evaluate the effectiveness of the proposed method. The first includes toy and UCI data sets [19], and the second is real-world engineering data set which is collected from cylindrical shell stochastic vibration system for structural response prediction. The proposed model selection method is compared with the original ELM [3], E-ELM [11], and ELM with LOO model selection [13]. Moreover, multi-dimensional SVM(M-SVM) is also introduced for comparison in the experiment of cylindrical shell data set. For simplicity, the proposed model selection method in this paper is called here U-ELM. ELM with LOO model selection proposed in [13] is called here M-ELM. In ELM, E-ELM, and U-ELM, the sigmoidal function: $\phi(\mathbf{x}) = \frac{1}{1+\exp(-(\mathbf{a}\cdot\mathbf{x}+b))}$ is employed as activation function. The parameters of E-ELM are set as follows: population size (NP), F and CR is set to 200, 1, 0.8, respectively, and the maximum iteration number is 20. In order to compare the generalization error reasonably, M-ELM and U-ELM are set to run the same number of fitness evaluations of E-ELM. The regularization parameter C in M-ELM is set 100. In every experiment, the hidden neurons are sequentially added into network and select the neurons with lowest generalization error as final model. All the experimental results are the mean values of 30 trails. All

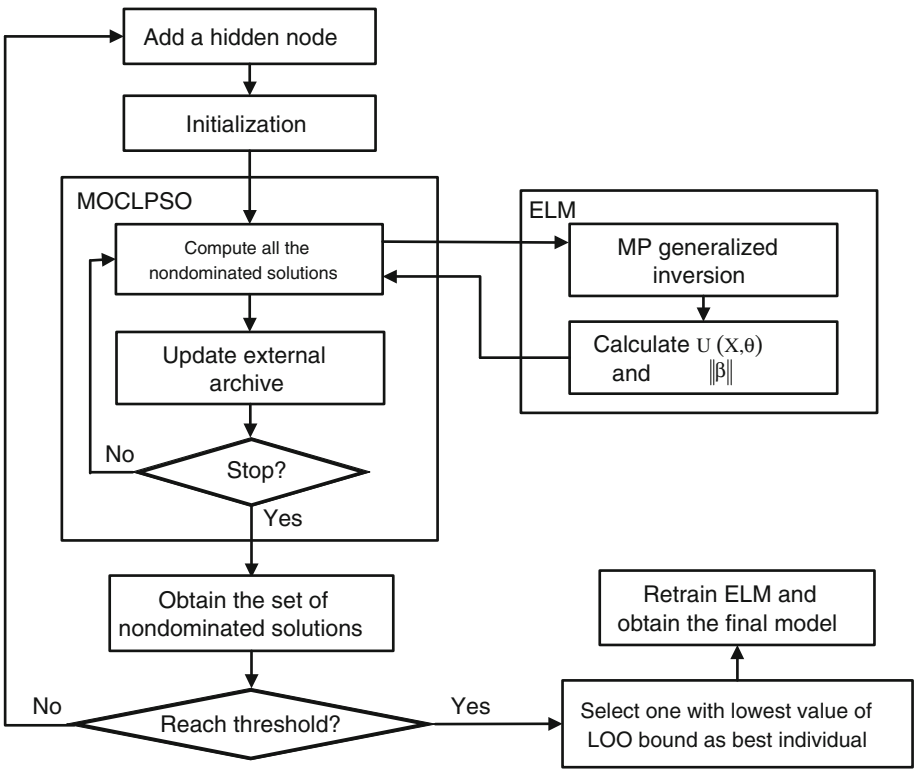


Fig. 2 Flowchart of model selection of ELM based on Riemannian metric

programs are carried out in MATLAB 7.10 environment running in a Core2, 2.66 GHz CPU and 3.37 GB RAM. Each of the input and output variables are rescaled linearly to the range $[-1, +1]$.

5.1 UCI data sets

First, the effectiveness of U-ELM is tested. Here, a noisy sinc function is introduced in favor of illustration comparison. The training set including 100 samples $\{(x_i, y_i)\}$ is generated with x_i drawn from $[-3, +3]$ uniformly, and $y_i = \sin(\pi x_i)/(\pi x_i) + e_i$ where e_i is a Gaussian noise term with zero mean and a variance of 0.1. We take 70 % of the data for training and the remaining for test. Then, U-ELM and ELM are applied on this data set, respectively. Here, the illustrative performance of M-ELM is similar with U-ELM, and the corresponding predictive curve is omitted. The predictive performance is plotted in Fig. 3.

As a priori knowledge, the sinc curve plotted by red line in Fig. 3 tends to have generalization ability. In Fig. 3, the decision curve obtained by U-ELM gets more close to the sinc curve than ELM, which implies the proposed method can greatly improve the generalization performance of ELM. The numerical results also prove the comparison: the training error (RMSE) of U-ELM is 0.0346 while ELM is 0.0687, and the RMSE on test set of U-ELM is 0.0283 while ELM is 0.0504. The number of hidden neurons in ELM is set 15, while this value chosen by U-ELM automatically is 8.

Secondly, we compare U-ELM with ELM, E-ELM, and M-ELM on five UCI regression data sets [19]. Some statistics of these five data sets are listed in Table 1. These data sets are divided into training, validation, and test set. Note that only E-ELM needs validation set to calculate the validation RMSE as fitness function. The experiments are implemented 30 times with random partitions of data set.

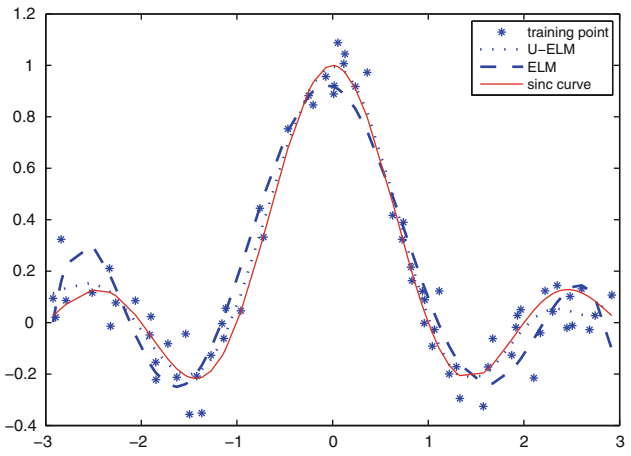


Fig. 3 Decision curves obtained by U-ELM and ELM on noisy sinc data set

Table 1 Specifications of five UCI regression data sets

Name	Attributes	Number of observations		
		Training	Validation	Test
Housing	13	354	76	76
Pyrim	27	51	11	12
Triazines	60	130	26	30
Mpg	7	274	59	59
Bodyfat	14	176	38	38

The ration of random partition is set as the statistics listed in Table 1. The mean and standard error of 30 RMSEs are reported in Table 2.

From Table 2, U-ELM greatly outperforms ELM and E-ELM on all five data sets in terms of test error and gets a minor improvement to M-ELM. The apparent reduction in test error suggests that the proposed method can provide an accurate evaluation of generalization performance rather than LOO bound used in M-ELM and RMSE on validation set used in E-ELM. It is worth noting that the adopted multi-objective optimization strategy could prevent overfitting by minimizing the norm of output weights simultaneously. As a result, the generalization performance of ELM is greatly improved by the proposed model selection method. We also find that ELM sometimes obtain very low training error, and M-ELM and E-ELM both get lower test error than ELM on most data sets. It can be drawn a conclusion that ELM tends to be overfitting in some cases, and model selection will prevent it via introducing the accurate evaluation of generalization ability. In experiments, we also found M-ELM depends on the regularization parameter C. The results show a prefixed value of C will obstruct getting better predictive performance. However, tuning the parameter C will add unnecessary computational time in turn. On the other hand, U-ELM evaluates the generalization ability of ELM by calculating uncertainty degree in input space directly. It does not need to introduce any parameters. The results also indicate using Riemannian metric to evaluate generalization ability of ELM works accurately than using LOO error. To some extents, "microcosmic" manner can provide better performance than "macroscopic" way.

In Table 2, the network architecture is also checked. We find U-ELM tends to get smaller number of hidden neurons than other three algorithm on all five data sets. The comparative results demonstrate using Riemannian metric to measure the uncertainty of hidden neurons can exclude the unnecessary ones and get optimal input weights and biases. Therefore, U-ELM gets more compact networks. Obviously, M-ELM also get smaller number of hidden nodes than ELM and E-ELM in general, which has been proved in previous work [13].

Table 2 Comparison between U-ELM, ELM, E-ELM, and M-ELM using 30 random splits of data sets

Problem	Algorithm	Training time (s)	RMSE		Hidden
			Training	Test	
Housing	ELM	0.0284	2.4782 (1.42e−1)	5.2893 (9.38e−1)	67
	E-ELM	34.764	3.1564 (1.63e−1)	4.2698 (7.54e−1)	35
	M-ELM	28.230	1.9467 (1.30e−1)	1.6548 (1.35e−1)	18
	U-ELM	21.156	1.2240 (1.02e−1)	1.0415 (0.85e−1)	15
Pyrim	ELM	0.0589	0.0781 (1.42e−3)	0.1246 (5.89e−2)	58
	E-ELM	3.4822	0.0528 (3.24e−3)	0.0851 (5.12e−2)	13
	M-ELM	3.5894	0.0614 (1.98e−3)	0.0624 (3.62e−2)	6
	U-ELM	2.2452	0.0681 (1.71e−3)	0.0482 (2.08e−3)	6
Triazines	ELM	0.0581	0.1721 (9.36e−3)	0.1821 (3.72e−2)	54
	E-ELM	25.486	0.1468 (1.24e−3)	0.1682 (2.05e−2)	42
	M-ELM	23.871	0.1268 (1.15e−2)	0.1479 (1.54e−2)	26
	U-ELM	18.526	0.1382 (1.26e−2)	0.1171 (1.45e−2)	21
Mpg	ELM	0.0681	2.8932 (1.54e−1)	2.7112 (4.08e−1)	58
	E-ELM	25.871	2.3214 (1.52e−1)	2.5216 (4.01e−1)	42
	M-ELM	24.264	2.3291 (1.65e−1)	2.2582 (3.68e−1)	32
	U-ELM	20.135	2.3331 (1.26e−1)	1.8264 (2.33e−1)	28
Bodyfat	ELM	0.3587	0.0035 (7.44e−4)	0.0042 (2.85e−3)	48
	E-ELM	12.576	0.0025 (5.21e−4)	0.0037 (2.52e−3)	27
	M-ELM	10.846	0.0028 (6.25e−4)	0.0031 (2.15e−3)	15
	U-ELM	9.5882	0.0019 (5.51e−4)	0.0022 (1.86e−3)	11

Standard error of 30 RMSEs is listed in bracket

Moreover, the computational costs of all four algorithms are tested. From Table 2, we find that U-ELM needs less training time than E-ELM and M-ELM. It is because that with same size of population, the uncertainty evaluation is more easy to calculate than LOO bound in M-ELM and RMSE in E-ELM. From Eq. (8), the uncertainty degree only needs to compute a simple derivative rather a whole training process. Although U-ELM is somewhat computationally expensive compared to original ELM, the training time is still reasonable and acceptable. In addition, we also find that the number of hidden neurons can affect the training time greatly. Obviously, hidden neurons determine the size of individual in E-ELM, M-ELM, and U-ELM and further influence the computational cost of whole algorithm. In spite of many hidden neurons required, ELM only needs very little training time.

5.2 Cylindrical shell vibration data set

In this section, the proposed method is tested in the problem of structural response prediction. This problem plays an increasingly role in the fields of mechanical design and manufacture. Based on the dynamical similarity existing between vibration responses of the specific structure under different boundary conditions, this problem tries to predict the response of a mechanical structure using the response of a same structure under another condition. This process can be viewed as a multi-input multi-output (MIMO) model. Considering the superior performance, ELM is introduced to establish the regression model. The whole process of structural response prediction can be found in our another work [20]. For the sake of paper's integrity, here we give a detailed description about theoretical derivation.

This method rests on the following understanding: once a dynamical system has been defined, the relationship between load and response is intrinsic and will be determined as long as the structure and boundary condition are determined. According to dynamic theory, model, boundary condition, and load are three key factors to work upon system response. Once the structure has been constrained with boundary condition, there is only load left to influence response due to the fixity of local physical property of boundary condition as well as nature property of structure. Without loss of generality, the structure and boundary 1 constitute system 1, and the same structure and boundary 2 constitute system 2. Intuitively, if the external loads are restricted to be uniform, the relationship between system 1 and 2 can be exploited only via intrinsic property of two systems, which can be reformulated as the following regression case:

$$x_{1ij} = \frac{\sum_{k=1}^{n} \varphi'_{1ik} \cdot z_{1kj}}{\sum_{k=1}^{n} \varphi'_{2ik} \cdot z_{2kj}} x_{2ij} = \psi_{ij}(t) x_{2ij}$$

in time domain and:

$$x_{1ij}(\omega) = \frac{\sum_{i,j=1}^{n} h_{1ij}(\omega)}{\sum_{i,j}^{n} h_{2ij}(\omega)} x_{2ij}(\omega) = \psi_{ij}(\omega) x_{2ij}(\omega)$$

in frequency domain. Here, $\psi_{ij}(t)$ is a typical regression model between x_{1ij} and x_{2ij}, and more importantly, it only depends on the intrinsic characteristics of two systems. In this section, we do not focus on the dynamic analysis and only provide the flowchart of structural response prediction, as listed in Fig. 4.

In Fig. 4, there are three marks which need to be explained in detail:

1. Loads with different forms are needed to imitate different dynamic environments. To reflect intrinsic characteristic of structure effectively, impact excitation and stochastic white noise are recommended to produce response data for training.
2. Data preprocessing mainly contains noise reduction for experimental data and feature extraction, for example, building frequency data via FFT transformation from time domain.
3. Without loss of generality, the response data of system 1 are set input and the data of system 2 are output. Considering the MIMO mapping model, the sample number equals load number, and the dimensions of input and output both equal the length of time or frequency sampling range.

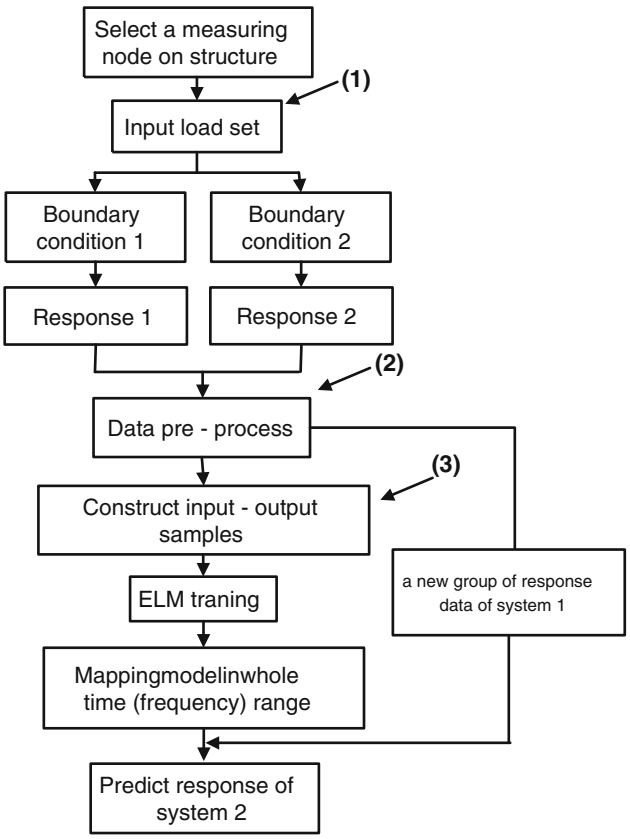


Fig. 4 Framework of structural response prediction based on ELM

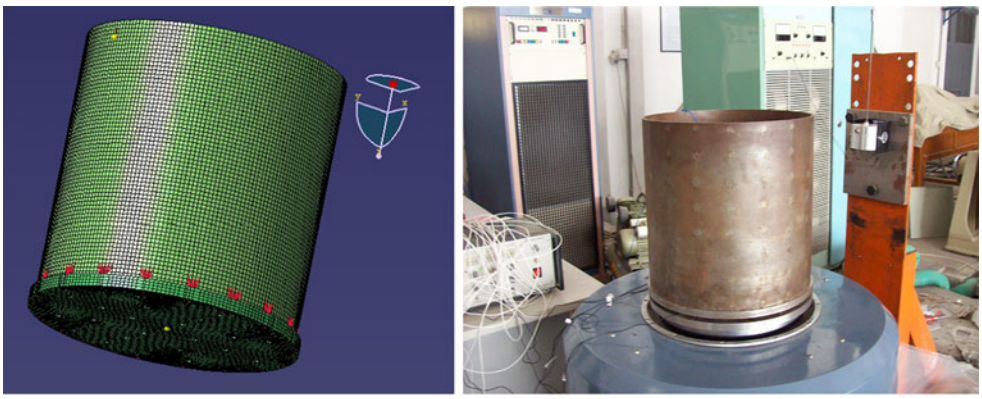
To evaluate the proposed method, we construct different systems which are composed of a cylindrical shell and different clamps. To represent simple and complicated boundary conditions, the shell is assembled by steel and magnesium aluminum (Mg–Al) alloy clamps, respectively. In this study, the vibration response of cylindrical shell constrained by Mg–Al clamp (referred to as system 1) is set to predict by the response of cylindrical shell constrained by steel clamp (system2). The physical characteristics of cylindrical shell and two clamps can be found in the paper [20].

Two dynamic data sets are constructed for test: simulation shock response data in time domain and experimental stochastic response data in frequency domain. A finite element models of cylindrical shell with two different clamps, built in LMS Virtual.Laboratory, are adopted to generate shock response data. And a shaking table stochastic vibration system is also set up to generate stochastic response data. These two experimental models are illustrated in Fig. 5. Note that the concrete dynamic analysis of these two systems can be found in [20].

For better comparison, we add an excellent MIMO modeling algorithm, M-SVR which was proposed by Pérez-Cruz [21]. Gaussian RBF kernel is used and defined as $K(x, x') = \exp(-\frac{\|x-x'\|^2}{2\sigma^2})$. We construct a total 30 groups of load signal to generate acceleration response in which 29 groups are used for training, and another group is for test. We choose the data in entire time or frequency range to construct the MIMO samples.

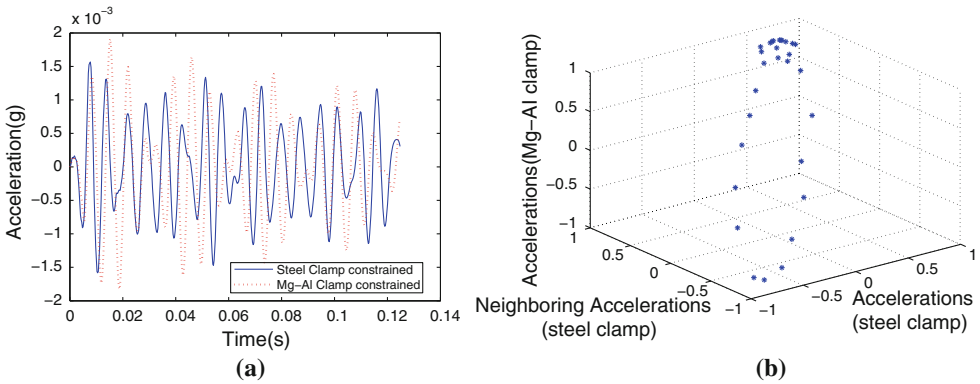
Firstly, we test the effectiveness of U-ELM in time domain. The distributions of acceleration responses under two different boundary conditions in time domain are listed in Fig. 6.

Obviously, there is great difference between two groups of response of two systems which have different boundary conditions. That means the dynamic characteristics of a structure under different boundary conditions are greatly different from the perspective of data analysis. The target of this research is to predict the response under Mg-AI clamp from the response under steel clamp, as shown in the following experimental results. In Fig. 6b, the vibration responses of two different systems are mainly in a ring distribution, which indicates there exists a nonlinear regression relationship between the responses which are excited by different loads under two boundary conditions. Therefore, the proposed method as listed in Fig. 2 is



(a) (b)

Fig. 5 Experimental setup of cylindrical shell with **a** finite element model and **b** stochastic vibration system



(a) (b)

Fig. 6 Acceleration responses under two different boundary conditions in time domain excited with distribution of response excited by **a** a same impact load and **b** 29 groups of impact loads

applicable. Randomly selecting a group of response data for test, the prediction performance of the proposed method on the cylindrical shell with Mg-AI clamp is illustrated in Fig. 7. For comparison, the performances of M-SVR and M-ELM are also illustrated in Fig. 7. Here, the number of neurons in ELM is set 100. In M-SVR, PSO with LOO error is applied to choose the optimal hyper-parameters. The numbers of particle and iteration of PSO are both set 50.

Note that the data in this experiment are simulated data in time domain. Because the data are out of noise, the graphical effects in Fig. 7 all look good. However, in Fig. 7, the prediction performance of the proposed method using U-ELM is better than the other two methods. As shown in Fig. 7c, the predicted response curve is very close to the real response curve, which indicates the benefit of the proposed method in structural response prediction. The other two methods can also get relative good prediction results. However, in Fig. 7a, the predicted curve of M-SVR deviates the true curve notably, and in Fig. 7b, the predicted curve of M-ELM fluctuates at many peaks. The reason comes from the defects of M-SVR and M-ELM when facing MIMO problem. Although M-SVR can effectively establish MIMO regression model, it is sensitive to hyper-parameters in some small-size sample problems. Similar to M-SVR, the performance of M-ELM depends slightly on the regularization parameter. In addition, it has to be proved that cross-validation does not generally work well in small-size sample regression problems [22]. Therefore, in spite of model selection, the generalization performance of M-SVR and M-ELM is still unsatisfactory at some peaks. On the contrary, avoiding calculating LOO error, U-ELM uses Riemannian metric to exploit the property of ELM in microcosmic manner, and hence is good at solving small-size MIMO problems.

Secondly, we test the effectiveness of U-ELM in frequency domain. Similar to time domain, we also provide the distributions of the power spectral densities (PSD) of acceleration responses under two different boundary conditions in frequency domain, as listed in Fig. 8. The sampling frequency of PSD is 4096Hz and frequency interval is 1Hz. Maximum entropy spectral estimation method is utilized to de-noise and smooth the signals.

Clearly, there is obvious dynamic difference in frequency domain between two systems which are composed of a same structure and different two boundary conditions.

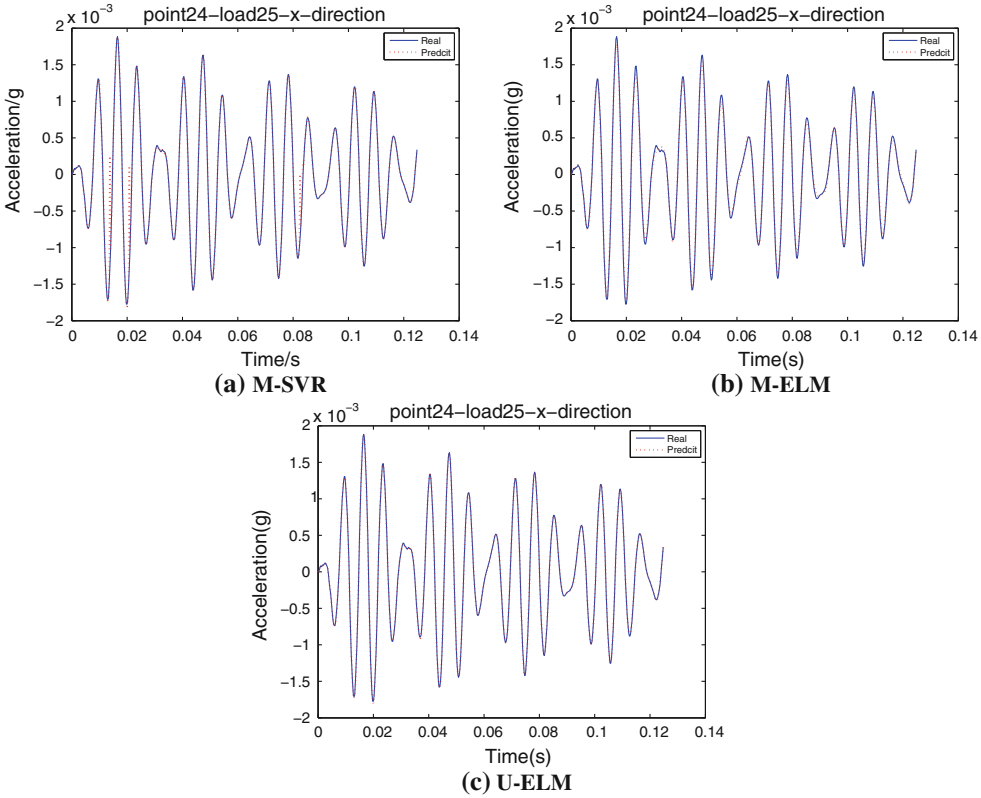


Fig. 7 Real and predicted shock acceleration response of cylindrical shell with Mg-AI clamp by response under steel clamp using the method of **a** M-SVR, **b** M-ELM, and **c** U-ELM

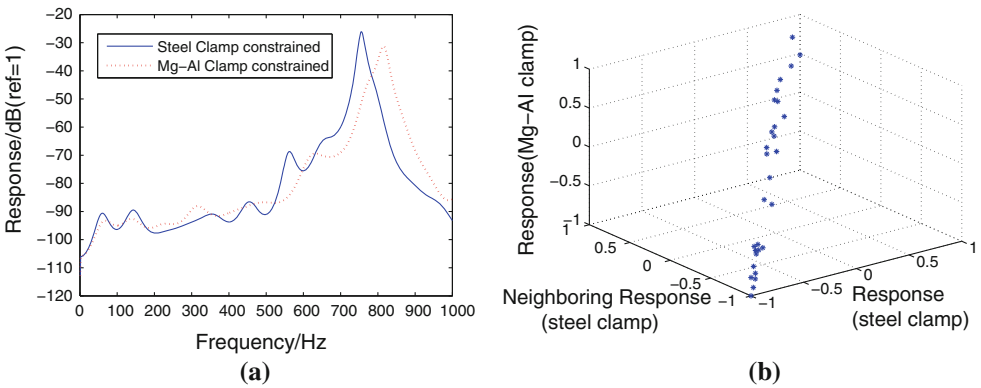


Fig. 8 PSDs of acceleration responses under two different boundary conditions in frequency domain excited with distribution of response excited by **a** a same load and **b** 29 groups of stochastic loads

And the PSDs of two systems' acceleration responses are approximately in a linear distribution. Intuitively, the proposed method is suited to establish prediction model in frequency domain. Randomly selecting a group of response data for test, the prediction performances of the proposed method and two other algorithm, M-SVR and M-ELM, on the cylindrical shell with Mg-AI clamp are illustrated in Fig. 9. The experimental setup is equal to the case in time domain.

Note that the stochastic responses used in this experiment inevitably contain noises. It is obvious that the propose method has better prediction performance than two other methods. More importantly, U-ELM and M-ELM both outperform M-SVR in terms of stability and accuracy. The reason has been discussed above. Although U-ELM gets similar performance with M-ELM, it is still more stable when facing small-size sample problem.

For better comparison, the numerical results of Figs. 7 and 9 are listed in Table 3. It can be observed that the numerical results keep line with illustrative comparison.

6 Conclusions

In this paper, the problem of model selection for ELM is addressed from the perspective of Riemannian geometry. The target of this paper is to obtain the best generalization

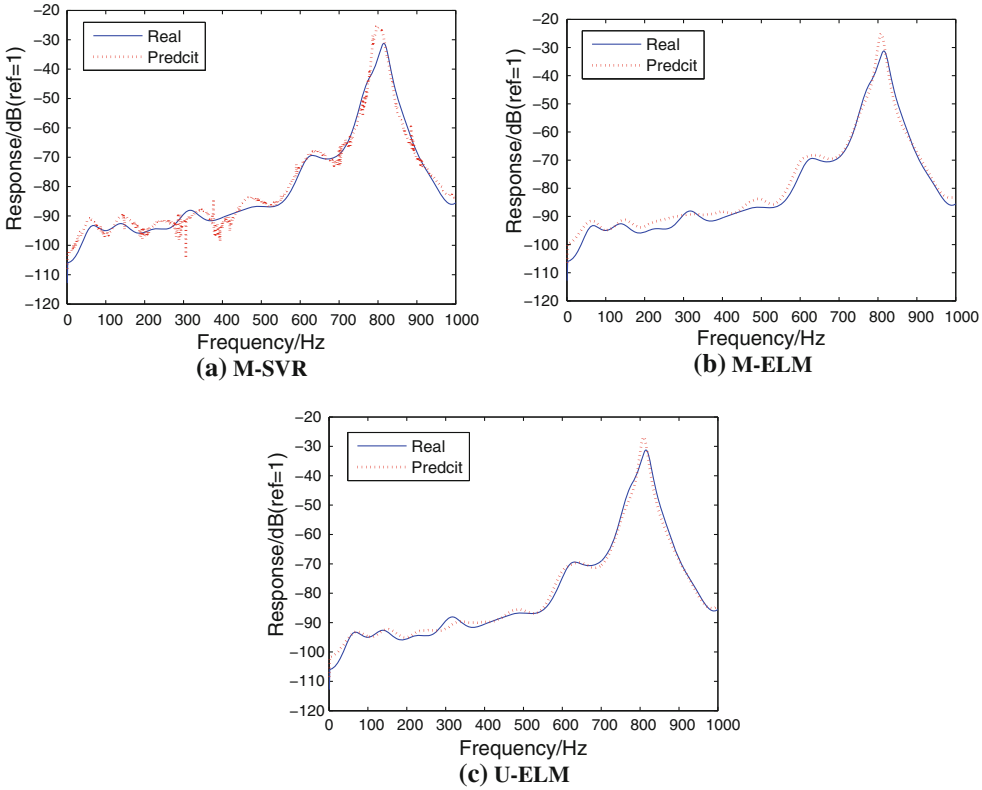


Fig. 9 Real and predicted stochastic vibration response of cylindrical shell with Mg-AI clamp by response under steel clamp using the method of **a** M-SVR, **b** M-ELM, and **c** U-ELM

Table 3 Comparative results of three methods on simulation shock and experimental stochastic data in terms of RMSE, average percentage error (APE), and maximum percentage error (MPE)

	Simulation shock data			Experimental stochastic data		
	M-ELM	M-SVR	U-ELM	M-ELM	M-SVR	U-ELM
RMSE	1.04e−004	4.34e−005	1.17e−005	6.71e−003	4.69e−003	3.18e−003
APE(%)	3.93	12.59	3.59	25.49	23.52	13.45
MPE(%)	226.46	444.47	105.62	248.05	126.24	83.68

performance with compact network. Different from most of the current researches, this paper tries to reach this target by evaluating the uncertainty of hidden neurons. The starting point is considering uncertainty of hidden neurons as the deviation of neurons from decision hyperplane in feature space. Therefore, this paper first proves the uncertainty can be characterized by a form of Riemannian metric and further proves the generalization performance of ELM is closely related to this uncertainty. The defined uncertainty degree has two advantages: it can be calculated very simply in input space, and it provides a way to exploit the inner structure of ELM in microcosmic manner. Finally, this paper utilizes a multi-objective optimization algorithm, MOCLPSO, to minimize two objectives, the uncertainty degree and the norm of output weights, simultaneously. In obtained non-dominant solutions, we choose the one leading to lowest value of LOO bound as best solution. Experimental results on five UCI data sets and a real-life cylindrical shell engineering data set show that the proposed model selection method has faster learning speed and lower generalization error than conventional ELM with model selection, and can greatly improve the generalization performance than original ELM and multi-dimensional SVM. Although a few computational costs are needed additionally, the experimental results still indicate the benefit of model selection for ELM using Riemannian metric. Note that although this paper takes regression problem for example, the whole analysis procedure can be easily extended to classification case.

Acknowledgments We thank the author Suganthan of [18] for the implementation of MOCLPSO. This work was supported by National Natural Science Foundation of China (NO.U1204609,60873104) and Key Scientific and Technological Project of Henan Province, China (No. 122102210086).

References

1. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of 2004 international joint conference on neural networks (IJCNN'2004), Budapest, Hungary, pp 985–990
2. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B 42:513–529
3. Huang GB, Zhu X, Siew C et al (2006) Extreme learning machine: theory and applications. Neurocomputing. 70:489–501
4. Chen FL, Ou TY (2011) Sales forecasting system based on gray extreme learning machine with taguchi method in retail industry. Expert Syst Appl 38:1336–1345
5. Minhas R, Baradarani A, Seifzadeh S, Wu QMJ (2010) Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing 73:1906–1917
6. Mohammed A, Wu QMJ, Sid-Ahmed M (2010) Application of wave atoms decomposition and extreme learning machine for fingerprint classification. Lect Notes Comput Sci 6112: 246–256
7. Huang GB, Wang D, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Lean Cybern 2:107–122
8. Feng G, Huang GB, Lin Q, Gay R (2009) Error minimized extreme learning machine with growth of hidden nodes and incremental learning. IEEE Trans Neural Netw 20:1352–1357
9. Lan Y, Soh YC, Huang GB (2010) Random search enhancement of error minimized extreme learing machine. In: Proceedings of European symposium on artificial neural networks (ESANN 2010), Bruges, Belgium, pp 327–332
10. Li K, Huang GB, Ge SS (2010) Fast construction of single hidden layer feedforward networks. In: Rozenberg G, Bäck T, Kok JN (eds) Handbook of natural computing. Springer, Berlin
11. Zhu QY, Qin AK, Suganthan PN, Huang GB (2005) Evolutionary extreme learning machine. Pattern Recogn 38:1759–1763
12. Lan Y, Soh YC, Huang GB (2010) Two-stage extreme learning machine for regression. Neurocomputing 73:3028–3038
13. Mao W, Tian M, Cao X, Xu J (2013) Model selection of extreme learning machine based on multi-objective optimization. Neural Comput Appl 22(3):521–529
14. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying Kernel functions. Neural Netw 12:783–789
15. Huang GB, Babri HA (1998) Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9:224–229
16. Burges CJC (1998) Geometry and invariance in Kernel based methods. In: Schölkopf B, Burges CJC, Smola AJ (eds) Advances in Kernel methods: support vector learning. MIT Press, Cambridge, pp 89–116
17. Huang GB, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74:155–163
18. Huang VL, Suganthan PN, Liang JJ (2006) Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems. Int J Intell Syst 21: 209–226
19. Newman DJ, Hettich S, Blake CL, Merz CJ (1998) UCI Repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA
20. Mao W, Yan G, Dong L (2012) A novel machine learning based method of combined dynamic environment prediction. J Sound Vib (under review)
21. Sánchez-fernández M, De-prado-cumplido M, Arenas-garcía J, Pérez-Cruz F (2004) SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems. IEEE Trans Signal Process 52:2298–2307
22. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of 14th international conference on artificial intelligence (IJCAI), pp 1137–1143