Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Extremal optimization for optimizing kernel function and its parameters in support vector regression

Peng CHEN[†1], Yong-zai LU[2]

([1]*Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China*)

([2]*Department of Automation, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: pengchen@sjtu.edu.cn

Received Apr. 20, 2010; Revision accepted June 25, 2010; Crosschecked Mar. 4, 2011

Abstract: The performance of the support vector regression (SVR) model is sensitive to the kernel type and its parameters. The determination of an appropriate kernel type and the associated parameters for SVR is a challenging research topic in the field of support vector learning. In this study, we present a novel method for simultaneous optimization of the SVR kernel function and its parameters, formulated as a mixed integer optimization problem and solved using the recently proposed heuristic 'extremal optimization (EO)'. We present the problem formulation for the optimization of the SVR kernel and parameters, the EO-SVR algorithm, and experimental tests with five benchmark regression problems. The results of comparison with other traditional approaches show that the proposed EO-SVR method provides better generalization performance by successfully identifying the optimal SVR kernel function and its parameters.

Key words: Support vector regression (SVR), Extremal optimization (EO), Parameter optimization, Kernel function optimization
doi:10.1631/jzus.C1000110 **Document code:** A **CLC number:** TP181

1 Introduction

Based on statistical learning theory and the structural risk minimization principle, one of the most critical statistical learning solutions, called the support vector machine (SVM), was first proposed by Vapnik to solve pattern recognition problems (Vapnik, 1995; Burges, 1998). In the last decade, SVM based algorithms have been developed rapidly and employed in many real-world applications, such as handwriting recognition, identification, bioinformatics, classification, and regression.

High generalization capability and global optimal solution constitute the major advantages of SVM over other machine learning approaches. However, the performance of SVM strongly depends on the embedded kernel type (Ali and Smith, 2006) and the associated parameter values (Min and Lee, 2005; Jeng, 2006; Hou and Li, 2009). Therefore, the SVM kernel function and its parameters should be selected carefully on a case-to-case basis as different functions and parameters may lead to widely varying performance. Up to now, a lot of kernels have been proposed by researchers, but there are no effective guidelines or systematic theories concerning how to choose an appropriate kernel for a given problem. The empirical search for the SVM kernel and parameters through a trial-and-error approach has proven to be often time consuming, imprecise, and unreliable (Lorena and de Carvalho, 2008; Zhang *et al.*, 2010).

New parameter optimization techniques for SVM have been proposed and investigated by researchers in recent years. Among them, the most commonly used methods are the grid search (Hsu *et al.*, 2004) and computational intelligence (CI) (Engelbrecht, 2007). The former is time consuming (Avci, 2009) and can adjust only a few parameters (Friedrichs and Igel, 2005). In contrast, the CI methods have shown high suitability for constrained nonlinear optimization problems, and can avoid local

minima inherently.

To date, the optimization of support vector regression (SVR) parameters based on CI methods has been realized in many studies and applications (Friedrichs and Igel, 2005; Mao *et al.*, 2005; Pai and Hong, 2005; Wu *et al.*, 2007; Saini *et al.*, 2010; Avci, 2009; Tang *et al.*, 2009; Wu, 2010; Zhang *et al.*, 2010). However, there has been relatively limited research published on how to determine an appropriate SVR kernel function automatically (Howley and Madden, 2005; Thadani *et al.*, 2006), and few of the existing works were devoted to the simultaneous optimization of the SVR kernel and its associated parameters (Wu *et al.*, 2009). Therefore, this paper proposes a novel SVR tuning algorithm based on extremal optimization (EO), to improve the predictive accuracy and generalization capability. EO is a recently proposed general-purpose heuristic algorithm, with the superior features of self-organized criticality (SOC), non-equilibrium dynamics, and co-evolutions in statistical mechanics and ecosystems. Inspired by far-from-equilibrium dynamics, this approach provides a new philosophy to optimization using non-equilibrium statistical physics and the capability to elucidate the properties of phase transitions in complex optimization problems (Boettcher and Percus, 1999). In this paper, we will, for the first time, apply EO in the simultaneous optimization of the SVR kernel and its parameters. Considering the complexity of SVR tuning, a novel EO-SVR algorithm with a carefully designed chromosome structure and fitness function is proposed to deal with this challenging issue.

2 Problem formulation

In this section, the basic idea of SVR is first introduced in Section 2.1, and then the determination of the optimal SVR kernel function and parameters is formulated into a mixed integer optimization problem in Section 2.2.

2.1 Support vector regression

SVR is a technique to build SVM for regression learning. Suppose we have a set of learning samples $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^n$ represents the input vector and has a corresponding output $y_i \in \mathbb{R}$ for $i=1, 2, \ldots, l$, where l denotes the size of the training data. The SVR regression function can be generally described by

$$f(\boldsymbol{x})=\boldsymbol{w}^{\mathrm{T}}\Phi(\boldsymbol{x})+b, \tag{1}$$

where $\boldsymbol{w}\in\mathbb{R}^n$, $b\in\mathbb{R}$, and Φ denotes a nonlinear transformation from a low dimensional space of $\mathbb{R}^n$ to a high dimensional feature space. The training of SVR is to find $\boldsymbol{w}$ and b values by minimizing the regression risk:

$$R_{\mathrm{reg}}(f)=\gamma\sum_{i=1}^{l}\Gamma(f(\boldsymbol{x}_i)-y_i)+\frac{1}{2}\|\boldsymbol{w}\|^2, \tag{2}$$

where $\Gamma(\cdot)$ is a cost function and γ is a constant that determines penalties to regression errors, and the vector $\boldsymbol{w}$ can be written in terms of data points:

$$\boldsymbol{w}=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\Phi(\boldsymbol{x}_i), \tag{3}$$

where α_i and α_i^* are the (only a few) non-zero Lagrange multipliers, and the corresponding samples are called support vectors. More detailed descriptions for the training process of SVR can be found in Steve (1998).

Substituting Eq. (3) into Eq. (1), the general form can be rewritten as

$$\begin{aligned} f(\boldsymbol{x}) &= \left(\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\Phi(\boldsymbol{x}_i)\right)^{\mathrm{T}}\Phi(\boldsymbol{x})+b \\ &= \sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(\boldsymbol{x}_i,\boldsymbol{x})+b. \end{aligned} \tag{4}$$

The dot product in Eq. (4) can be replaced by kernel function $k(\boldsymbol{x}_i, \boldsymbol{x})$, which provides an elegant way of dealing with nonlinear mapping in the feature space, thereby avoiding all difficulties inherent in higher dimensions. There are several commonly used kernel types in SVR: linear, polynomial, radial basis function (RBF), and multilayer perceptron (MLP), as listed in Eqs. (5)–(8).

1. Linear kernel function:

$$k(\boldsymbol{x}_i, \boldsymbol{x})=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}. \tag{5}$$

2. Polynomial kernel function:

$$k(\boldsymbol{x}_i, \boldsymbol{x})=(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}+t)^d. \tag{6}$$

3. Radial basis function (RBF):

$$k(\boldsymbol{x}_i, \boldsymbol{x})=\exp(-\|\boldsymbol{x}_i-\boldsymbol{x}\|^2/(2\sigma^2)). \quad (7)$$

4. Multilayer perceptron (MLP) function:

$$k(\boldsymbol{x}_i, \boldsymbol{x})=\tanh(s\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}+t^2). \quad (8)$$

Herein, t, d, σ, and s are adjustable parameters. To design an effective SVR model, the kernel function and parameters must be chosen carefully.

2.2 Optimization of the SVR kernel function and its parameters

In this study, the selection of the optimal kernel function and parameters is formulated into an optimization problem and then solved using EO. The kernels and parameters described in Section 2.1 are first encoded into a chromosome, and a predefined cost function J is used to assess the performance of candidate solutions during the optimization. The performance is assessed in the standard way: by learning different SVRs with the training dataset and evaluating them on an independent validation dataset, as described in Eq. (9):

$$J = \underset{\prod \text{Validation set}}{f}(\text{kernel type, kernel parameters}), \quad (9)$$

where J is a function of 'kernel type' (discrete variable) and 'kernel parameters' (continuous variables). In this study, the mean square error (MSE) in Eq. (10) is selected as the cost function:

$$\begin{aligned} J &= \underset{\prod \text{Validation set}}{f}(\text{kernel type, kernel parameters}) \\ &= \underset{\prod \text{Validation set}}{\text{MSE}}(\text{kernel type, kernel parameters}) \\ &= \frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2, \end{aligned} \quad (10)$$

where N denotes the size of the validation data, y_i represents the actual output, and $\hat{y}_i$ is the SVR predicted value.

Based on the problem formulation described above, our goal is to employ the optimized procedure to explore a finite subset of the possible values and determine the kernel type and associated parameters that can minimize the cost function J:

$$\min \underset{\prod \text{Validation set}}{f}(\text{kernel type, kernel parameters})$$

subject to: (11)

ranges of parameter values and kernel types.

3 Hybrid extremal optimization for the support vector regression kernel function and its parameters

In this section, the development of the proposed EO-SVR method is described in detail. The basic conception of EO is briefly introduced. Then the detailed issues of applying EO to the SVR kernel and parameters optimization are discussed, including chromosome structure, the fitness function, and EO-SVR workflow.

3.1 Extremal optimization

The EO proposed by Boettcher and Percus (1999) is derived from the fundamentals of statistical physics and self-organized criticality (SOC) (Bak *et al.*, 1987) based on the Bak-Sneppen (BS) model (Bak and Sneppen, 1993), which simulates far-from-equilibrium dynamics in statistical physics. SOC states that large interactive systems evolve naturally to a state where a change in one single of their elements may lead to avalanches or domino effects that can reach any other element in the system. For an optimization problem with n decision variables, EO proceeds as follows (Boettcher and Percus, 2000):

1. Initialize a configuration S at will, and set $S_{\text{best}}=S$.

2. For the 'current' solution S:

(1) Evaluate the fitness for each decision variable (component) $\boldsymbol{x}_i$;

(2) Rank all the components by their fitness and find the component with the 'worst fitness';

(3) Choose one solution S' in the neighborhood of S, such that the worst component $\boldsymbol{x}_j$ must change its state;

(4) Accept $S=S'$ unconditionally;

(5) If $F(S)<F(S_{\text{best}})$, set $S_{\text{best}}=S$.

3. Repeat Step 2 as long as desired.

4. Return S_{best} and $F(S_{\text{best}})$.

Generally speaking, EO is particularly applicable in dealing with large complex problems with rough landscape, phase transitions passing 'easy-hard-easy' boundaries, or multiple local optima. It is

less likely to be trapped in local minima than traditional gradient-based algorithms. Having benefited from its generality and ability to explore complicated configuration spaces efficiently, EO and its derivatives have been successfully applied in solving multi-objective combinatorial hard benchmarks and real-world optimization problems (Chen MR *et al.*, 2007; Chen YW *et al.*, 2007; Lu *et al.*, 2007; Chen and Lu, 2008).

3.2 Extremal optimization based support vector regression kernel and parameters optimization

As described in Section 2, the simultaneous optimization of the SVR kernel function and its parameters falls into a mixed integer optimization problem, in which the discrete variable (kernel type) and the continuous variables (associated parameters) are parts of the same optimization problem. Traditional real coded EO algorithms are designed for the optimization of continuous variables and are unsuitable for these kinds of problems. In this work, a carefully designed fitness function is proposed to deal with the mixed integer optimization problem, in which the real valued gene (component) representing the SVR kernel type is rounded to the nearest integer for fitness calculation. Thus, all the genes can be treated as real values in EO mutation/representation. The benefit of this approach is that our proposed EO-SVR can operate independently of the variable types. There is no need for the additional design of a mixed chromosome representation and mutation operations. The EO based SVR kernel and parameters optimization is developed as follows.

3.2.1 Chromosome structure

In the proposed EO-SVR optimization procedure, the kernel types and associated parameters can be represented by a real coded chromosome without considering the variable types (discrete or continuous). The SVR kernel and associated parameters are directly coded into the chromosome as $S=\{\mathrm{KT}_i, P(1)_i, P(2)_i, P(3)_i\}$, where KT_i, $P(1)_i$, $P(2)_i$, and $P(3)_i$ denote the kernel type and associated kernel parameters, respectively. The chromosome structure of our proposed EO-SVR is shown in Fig. 1.

The SVR kernel functions and associated parameters to be optimized in this work are listed in Table 1.

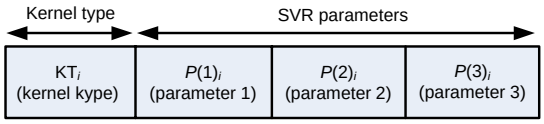

Fig. 1 Chromosome structure of the proposed EO-SVR (extremal optimization - support vector regression)

Table 1 Kernel types (KT) and associated parameters

KT value	KT	$P(1)$	$P(2)$	$P(3)$
0	LKF	γ		
1	PKF	γ	d	t
2	RBFK	γ	σ	
3	MLPK	γ	s	t

LKF: linear kernel function; PKF: polynomial kernel function; RBFK: radial basis function kernel; MLPK: multilayer perceptron kernel. $P(1)$, $P(2)$, and $P(3)$ are the associated kernel parameters

One of the main issues in the simultaneous optimization of SVR kernels and parameters is to deal with the different search spaces of parameter values for various kernel types. Therefore, we map all SVR parameters ($P(1)_i$, $P(2)_i$, and $P(3)_i$) to continuous values between 0 and 1 during EO evolution. When the fitness needs to be calculated, the parameters $P(j)_i$ (j=1, 2, 3) are converted to actual values $P(j)_{\mathrm{real}}$ based on the upper bound $P_{\mathrm{UB}}(j)_i$ and the lower bound $P_{\mathrm{LB}}(j)_i$ associated with kernel types (KT_i), as described in Section 3.2.2.

3.2.2 Fitness function

The EO-SVR in this work minimizes the cost function J in Eq. (10) to establish an efficient SVR model. As shown in Fig. 1, the chromosome in the optimized procedure is represented by a real-valued string of genes:

$$S=\{\mathrm{KT}_i, P(1)_i, P(2)_i, P(3)_i\}. \tag{12}$$

As mentioned previously, the real-valued variable (KT_i) is converted into an integer representing the kernel type before fitness evaluation, and the kernel parameters ($P(1)_i$, $P(2)_i$, and $P(3)_i$) are transformed to actual values, as described in Eq. (13):

$$\begin{aligned}\overline{S} &= \{\mathrm{ROUND}(\mathrm{KT}_i),\ \mathrm{CONVERT}(P(1)_i),\\ &\quad \mathrm{CONVERT}(P(2)_i),\ \mathrm{CONVERT}(P(3)_i)\}\\ &= \{\mathrm{KT}_{\mathrm{integer}},\ P(1)_{\mathrm{real}},\ P(2)_{\mathrm{real}},\ P(3)_{\mathrm{real}}\},\end{aligned} \tag{13}$$

where ROUND is a function that rounds to the nearest integer, and 'CONVERT' is a function that maps $P(1)_i$, $P(2)_i$, and $P(3)_i$ from [0, 1] to actual variable regions:

$$\begin{aligned} P(j)_{\text{real}} &= \text{CONVERT}(P(j)_i) \\ &= P_{\text{LB}}(j)_i + (P_{\text{UB}}(j)_i - P_{\text{LB}}(j)_i)P(j)_i, \end{aligned} \tag{14}$$

where j=1, 2, 3, and $P_{\text{LB}}(j)_i$ and $P_{\text{UB}}(j)_i$ are the variable bounds satisfying $P_{\text{LB}}(j)_i \le P(j)_{\text{real}} \le P_{\text{UB}}(j)_i$.

As described in Eq. (10), to solve the SVR optimization problems, the global fitness can be defined as

$$\text{Fitness}_{\text{global}}(\bar{S}) = \underset{\prod \text{Validation set}}{\text{MSE}}(\bar{S}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2. \tag{15}$$

Unlike GA, which works with a population of candidate solutions, EO evolves a single solution and makes local modification to the worst component. Each component needs to be assigned with a quality measure (i.e., fitness) called 'local fitness'. In this work, the local fitness λ_k is defined as an improvement in global fitness $\text{Fitness}_{\text{global}}$, made by the mutation imposed on the kth component of the best-so-far chromosome $\bar{S}$:

$$\begin{aligned} \lambda_k &= \text{Fitness}_{\text{local}}(\bar{S}'_k) = \Delta \underset{\bar{S}\to\bar{S}'_k}{\text{Fitness}}_{\text{global}}(\cdot) \\ &= \text{Fitness}_{\text{global}}(\bar{S}) - \text{Fitness}_{\text{global}}(\bar{S}'_k). \end{aligned} \tag{16}$$

3.2.3 EO-SVR workflow

Fig. 2 shows the workflow of our proposed EO-SVR method. The kernel function and associated parameters are optimized by EO-SVR with the initial chromosome randomly generated. By always performing mutations on the worst component and its neighbors successively, the component in EO can evolve itself towards the global optimal solution generation by generation. The EO-SVR terminates when a predefined number of iterations is reached. Then the optimal SVR kernel and parameter values can be finally determined.

4 Experimental results

In this section, we consider five benchmark regression problems, which cover the approximation of two single-variable functions and three multi-variable functions. As mentioned above, we chose four commonly used SVR kernel types (linear, polynomial, MLP, and RBF) listed in Table 1 as the candidate kernels. The programs were implemented in Matlab, and the experiments were carried out on a Pentium IV E5200 2.5 GHz PC with 2 GB RAM under the WinXP platform.

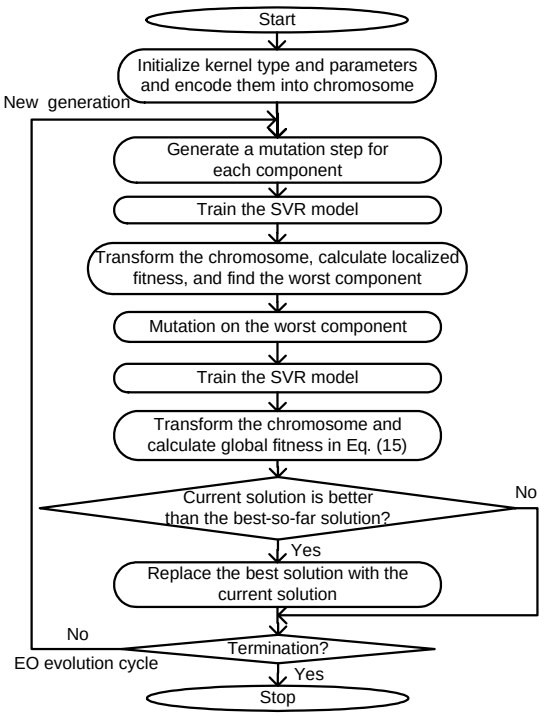

Fig. 2 The optimization process of EO-SVR (extremal optimization - support vector regression)

For comparison, we show the experimental results obtained using EO-SVR, RBF kernel with grid search, and the neural network (NN) model, respectively. The index used for performance comparison is the MSE on the test dataset.

4.1 Approximation of the single-variable function

To verify the effectiveness of the proposed EO-SVR algorithm, we first considered two illustrative problems involving one input and one output. Fig. 3 shows the original function examples and noisy learning examples (including training and validation examples) for those two single-variable functions.

Case 1 We approximate the following simple function (Chuang and Lee, 2011):

$$y = \frac{\sin x}{x}, \quad x \in [-10,\ 10]. \tag{17}$$

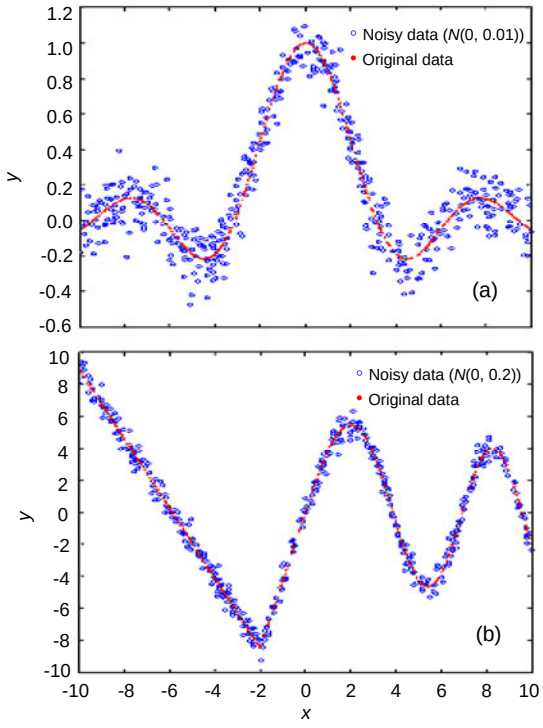

Fig. 3 Original function examples and noisy learning examples for Case 1 (a) and Case 2 (b)

We have 500 randomly generated examples, 200 of which are adopted as training examples, 200 as validation examples, and the remaining 100 as testing examples. The learning samples are disturbed by additive noise ξ with zero mean and standard derivation δ, as described in Eq. (18):

$$\xi=N(0,\ \delta),\ \delta\in\{0,\ 0.002,\ 0.005,\ 0.008,\ 0.010\}. \quad (18)$$

Fig. 4 illustrates the whole evolution process of MSE, the best-so-far result, and the searching dynamics of the kernel type during the EO-SVR optimization with the noise level δ=0.002.

Fig. 5 shows the predictions and the actual values of the output y for test samples. The predictions by EO-SVR show good agreement with the actual outputs under different noise levels.

The performance comparisons among EO-SVR, RBF kernel with grid search, and the NN model for Case 1 are listed in Table 2. The comparisons imply that the predictive accuracies of the traditional SVR with grid search can be improved by simultaneous optimization of the kernel type and parameters.

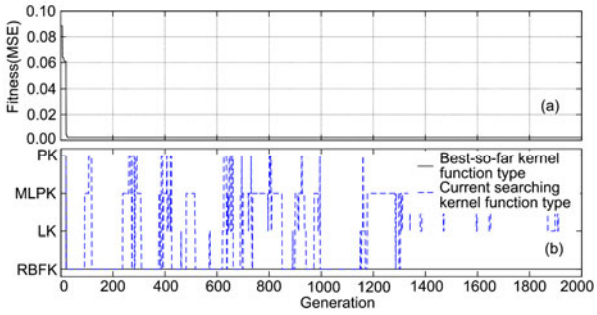

Fig. 4 Evolution process, best kernel type, and searching dynamics during EO-SVR (extremal optimization - support vector regression) optimization (Case 1, δ=0.002)
(a) Optimization process of EO-SVR; (b) Best kernel function type and searching dynamics of the kernel function during EO evolution. PK: polynomial kernel; MLPK: multilayer perceptron kernel; LK: linear kernel; RBFK: radial basis function kernel

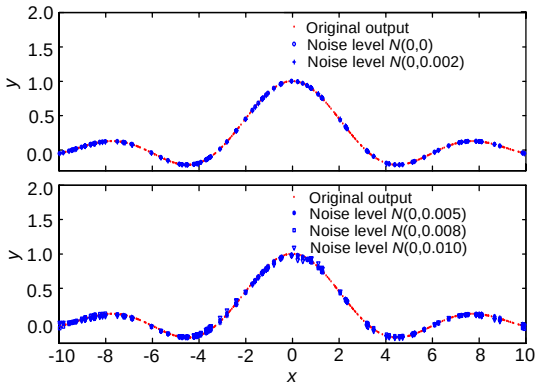

Fig. 5 EO-SVR (extremal optimization - support vector regression) based predictions and actual values for test samples under different noise levels

Case 2 We consider a more complex example (Zhang *et al.*, 2004):

$$y=\begin{cases}-2.186x-12.864, & -10\le x<-2,\\ 4.246x, & -2\le x<0,\\ 10\mathrm{e}^{-0.05x-0.5}\sin\left[(0.03x+0.7)x\right], & 0\le x\le 10.\end{cases} \quad (19)$$

In this case, we randomly generate 500 examples, 200 of which are adopted as training examples, 200 as validation examples, and the remaining 100 as testing examples. The learning samples are disturbed by an additive noise ξ with zero mean and standard derivation δ:

$$\xi=N(0,\ \delta),\ \delta\in\{0,\ 0.02,\ 0.05,\ 0.10,\ 0.20\}. \quad (20)$$

Table 2 Performance comparisons between EO-SVR (extremal optimization - support vector regression) and other conventional approaches (Case 1)

δ	Method	BK	Para 1	Para 2	Para 3	MSE
0	**EO-SVR**	**RBF**	**964.5230**	**0.2695**		**2.8266e-09**
	RBFKGS		671.7264	0.2888		3.9785e-09
	NN					2.3429e-08
0.002	**EO-SVR**	**RBF**	**998.8666**	**0.7968**		**5.8237e-05**
	RBFKGS		262.8475	0.6111		6.1862e-05
	NN					1.5466e-04
0.005	**EO-SVR**	**RBF**	**3.0704**	**0.1745**		**1.3436e-04**
	RBFKGS		2.1483	0.1385		1.8660e-04
	NN					4.4858e-04
0.008	**EO-SVR**	**RBF**	**8.2553**	**0.4723**		**2.1916e-04**
	RBFKGS		4.9925	0.3983		2.3985e-04
	NN					0.0010
0.010	**EO-SVR**	**MLP**	**530.0677**	**3.9864**	**0.6656**	**8.6413e-04**
	RBFKGS		4.8030	0.0960		8.9598e-04
	NN					0.0017

δ: noise level; BK: best kernel; Para: parameter; MSE: mean square error. RBFKGS: RBF kernel with grid search; NN: neural network; RBF: radial basis function; MLP: multilayer perceptron. The best evolved model with the optimal kernel function and associated parameters is highlighted in bold fonts

Table 3 Performance comparisons between EO-SVR (extremal optimization - support vector regression) and other conventional approaches (Case 2)

δ	Method	BK	Para 1	Para 2	Para 3	MSE
0	**EO-SVR**	**RBF**	**997.6446**	**0.0065**		**2.3383e-04**
	RBFKGS		616.9396	0.0236		5.5786e-04
	NN					0.0145
0.02	**EO-SVR**	**MLP**	**1758.8000**	**4.4702**	**2.0033**	**0.0034**
	RBFKGS		616.9396	0.0236		0.0042
	NN					0.0069
0.05	**EO-SVR**	**MLP**	**1140.1000**	**3.7743**	**1.6610**	**0.0045**
	RBFKGS		49.9279	0.0401		0.0061
	NN					0.0256
0.10	**EO-SVR**	**MLP**	**1431.3000**	**4.2647**	**1.8316**	**0.0106**
	RBFKGS		831.0181	0.0965		0.0124
	NN					0.0131
0.20	**EO-SVR**	**MLP**	**1096.3000**	**2.4390**	**1.5260**	**0.0216**
	RBFKGS		16.7501	0.1735		0.0304
	NN					0.0238

δ: noise level; BK: best kernel; Para: parameter; MSE: mean square error. RBFKGS: RBF kernel with grid search; NN: neural network; RBF: radial basis function; MLP: multilayer perceptron. The best evolved model with the optimal kernel function and associated parameters is highlighted in bold fonts

The improvement in generalization performance via EO-SVR is shown in Table 3. Compared with traditional methods, the prediction error can be further reduced by the proposed EO-SVR. Moreover, in many SVR applications, the RBF kernel function is supposed to have a better performance over other kernel functions; however, in this case, the MLP kernel with optimized parameters yields a lower prediction error and better generalization capability.

4.2 Approximation of the multi-variable function

In this part, the effectiveness of EO-SVR is further validated on three benchmark functions involving multiple inputs and one output. Fig. 6 shows the original function examples and noisy learning examples for Cases 3 and 4.

Case 3 A two-variable function is considered (Chuang and Lee, 2011):

$$y=\frac{\sin\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2}},\quad -5\le x_1, x_2\le 5. \tag{21}$$

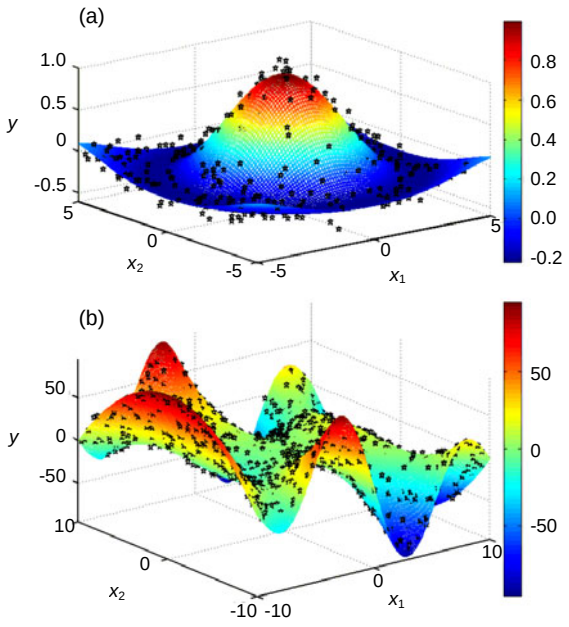

Fig. 6 Original function examples and noisy learning examples ('☆') for Case 3 (noise level $N(0, 0.01)$) (a) and Case 4 (noise level $N(0, 5)$) (b)

We have 700 randomly generated examples, 400 of which are adopted as training examples, 200 as validation examples, and the remaining 100 as testing examples. The learning samples are disturbed by an additive noise ξ with zero mean and standard derivation δ:

$$\xi=N(0, \delta),\ \delta\in\{0, 0.002, 0.005, 0.008, 0.010\}. \quad (22)$$

For comparison, the experimental results by EO-SVR, RBF kernel with grid search, and the NN model for Case 3 are summarized in Table 4.

Table 4 Performance comparisons between EO-SVR (extremal optimization - support vector regression) and other conventional approaches (Case 3)

δ	Method	BK	Para 1	Para 2	MSE
0	**EO-SVR**	**RBF**	**999.9703**	**0.9227**	**6.8732e-08**
	RBFKGS		547.4999	0.8840	1.2564e-07
	NN				1.4118e-04
0.002	**EO-SVR**	**RBF**	**3.1471**	**1.0067**	**1.5906e-04**
	RBFKGS		2.1308	0.7159	1.8197e-04
	NN				4.3281e-04
0.005	**EO-SVR**	**RBF**	**70.5141**	**2.5893**	**1.8091e-04**
	RBFKGS		78.9656	2.7952	1.8725e-04
	NN				2.6633e-04
0.008	**EO-SVR**	**RBF**	**8.4046**	**1.4856**	**4.9897e-04**
	RBFKGS		5.0120	1.2219	5.3518e-04
	NN				0.0018
0.010	**EO-SVR**	**RBF**	**2.3983**	**1.2687**	**6.5814e-04**
	RBFKGS		0.8969	0.7793	6.8268e-04
	NN				0.0017

δ: noise level; BK: best kernel; Para: parameter; MSE: mean square error. RBFKGS: RBF kernel with grid search; NN: neural network; RBF: radial basis function. Para 3 is not needed for all the methods. The best evolved model with the optimal kernel function and associated parameters is highlighted in bold fonts

Case 4 Consider a two-variable function as (Zhang *et al.*, 2004)

$$y=(x_1^2-x_2^2)\sin(0.5x_1),\ -10\le x_1,\ x_2\le 10. \quad (23)$$

We have 700 randomly generated examples, 400 of which are adopted as training examples, 200 as validation examples, and the remaining 100 as testing examples. The learning samples are disturbed by an additive noise ξ with zero mean and standard derivation δ:

$$\xi=N(0, \delta),\ \delta\in\{0, 0.5, 1, 2, 5\}. \quad (24)$$

Table 5 shows the simulation results comparison among the proposed EO-SVR, RBF kernel with grid search, and the NN model for Case 4. It can be seen that the prediction model, which is evolved by the proposed EO-SVM method, can always have satisfactory performances under different noise levels.

Table 5 Performance comparisons between EO-SVR (extremal optimization - support vector regression) and other conventional approaches (Case 4)

δ	Method	BK	Para 1	Para 2	MSE
0	**EO-SVR**	**RBF**	**999.9983**	**0.2437**	**0.0649**
	RBFKGS		557.4963	0.2478	0.0946
	NN				0.5883
0.5	**EO-SVR**	**RBF**	**998.0115**	**0.5039**	**0.1722**
	RBFKGS		588.8470	0.4691	0.1875
	NN				0.5972
1.0	**EO-SVR**	**RBF**	**998.0890**	**0.8163**	**0.3354**
	RBFKGS		434.7411	0.7442	0.3920
	NN				0.5844
2.0	**EO-SVR**	**RBF**	**999.3974**	**0.8031**	**0.5222**
	RBFKGS		434.7411	0.7442	0.5471
	NN				1.1521
5.0	**EO-SVR**	**RBF**	**879.9280**	**0.7545**	**0.9555**
	RBFKGS		215.5972	0.5907	1.0473
	NN				2.8953

δ: noise level; BK: best kernel; Para: parameter; MSE: mean square error. RBFKGS: RBF kernel with grid search; NN: neural network; RBF: radial basis function. Para 3 is not needed for all the methods. The best evolved model with the optimal kernel function and associated parameters is highlighted in bold fonts

Case 5 We approximate a widely used three-input nonlinear function (Qiao and Wang, 2008) to verify the effectiveness of the proposed EO-SVR:

$$y=(1+x_1^{0.5}+x_2^{-1}+x_3^{-1.5})^2,\ 1\le x_1,\ x_2,\ x_3\le 6. \quad (25)$$

The training samples consist of 600 randomly generated data. Another 200 examples are independently generated as the validation dataset, and 100 examples as the testing dataset. The learning samples are disturbed by additive noise ξ with zero mean and standard derivation δ, as described in Eq. (26):

$$\xi=N(0, \delta),\ \delta\in\{0, 0.02, 0.05, 0.08, 0.10\}. \quad (26)$$

Table 6 shows the forecasting accuracy on the testing dataset with the proposed EO-SVR, RBF kernel with grid search, and the NN model. Obviously, the developed EO-SVR model yields more appropriate kernel and parameters, thus giving higher predictive accuracy and generalization capability.

Table 6 Performance comparisons between EO-SVR (extremal optimization - support vector regression) and other conventional approaches (Case 5)

δ	Method	BK	Para 1	Para 2	Para 3	MSE
0	**EO-SVR**	**Poly**	**0.5282**	**8.7922**	**4.8795**	**1.0962e-07**
	RBFKGS		559.3278	23.4082		2.5487e-05
	NN					1.7727e-07
0.02	**EO-SVR**	**MLP**	**438.2326**	**0.1241**	**3.3632**	**4.4824e-04**
	RBFKGS		650.2169	19.1805		0.0012
	NN					0.0018
0.05	**EO-SVR**	**MLP**	**21.3813**	**0.0235**	**0.3436**	**8.1356e-04**
	RBFKGS		24.6381	36.8418		0.0021
	NN					0.0064
0.08	**EO-SVR**	**MLP**	**607.0087**	**0.0077**	**0.1461**	**0.0012**
	RBFKGS		13.4786	33.1361		0.0029
	NN					0.0044
0.10	**EO-SVR**	**MLP**	**31.4823**	**0.0396**	**2.2208**	**0.0015**
	RBFKGS		32.6444	38.0491		0.0028
	NN					0.0121

δ: noise level; BK: best kernel; Para: parameter; MSE: mean square error. RBFKGS: radial basis function kernel with grid search; NN: neural network; Poly: polynomial; MLP: multilayer perceptron. The best evolved model with the optimal kernel function and associated parameters is highlighted in bold fonts

4.3 Summary

In this section, we demonstrate the effectiveness of the proposed EO-SVR through experiments on five typical benchmark regression problems at different noise levels. As shown by the experimental results, the optimal kernel and parameters vary with the problems to be solved and the noise levels. The proposed EO-SVR can successfully determine the optimal SVR kernel type and associated parameter values consistently for all five cases, leading to a better generalization performance and a lower forecasting error. Our experimental results also show that the most frequently used SVR kernel function, the RBF kernel, may not be the best choice for nonlinear regression in some cases: the multilayer perceptron (MLP) kernel can offer a better performance if the associated parameters are well tuned.

5 Conclusions

We propose a new SVR tuning method, EO-SVR, for the automatic optimization of SVR kernel type and its parameters, to provide a better generalization performance and a lower forecasting error. The experiments on five benchmark regression problems in various noise environments are carried out to demonstrate the effectiveness of the proposed EO-SVR method. Experimental results show that the EO-SVR algorithm has a high potential of increasing the predictive accuracy when integrating EO with the traditional SVR model. Moreover, the method is robust under different noise levels. Future work includes more applications of the proposed EO-SVR in real world modeling and forecasting problems.

Acknowledgements

The authors would like to thank Supcon, Inc. for their financial support.

References

Ali, S., Smith, K.A., 2006. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, **70**(1-3):173-186. [doi:10.1016/j.neucom.2006.03.004]

Avci, E., 2009. Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithm—support vector machines: HGASVM. *Expert Syst. Appl.*, **36**(2):1391-1402. [doi:10.1016/j.eswa.2007.11.014]

Bak, P., Sneppen, K., 1993. Punctuated equilibrium and criticality in a simple model of evolution. *Phys. Rev. Lett.*, **71**(24):4083-4086. [doi:10.1103/PhysRevLett.71.4083]

Bak, P., Tang, C., Wiesenfeld, K., 1987. Self-organized criticality: an explanation of the 1/*f* noise. *Phys. Rev. Lett.*, **59**(4):381-384. [doi:10.1103/PhysRevLett.59.381]

Boettcher, S., Percus, A.G., 1999. Extremal Optimization: Methods Derived from Co-evolution. Proc. Genetic and Evolutionary Computation Conf., p.825-832.

Boettcher, S., Percus, A., 2000. Nature's way of optimizing. *Artif. Intell.*, **119**(1-2):275-286. [doi:10.1016/S0004-3702(00)00007-2]

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, **2**(2):121-

167. [doi:10.1023/A:1009715923555]

Chen, M.R., Lu, Y.Z., 2008. A novel elitist multiobjective optimization algorithm: multiobjective extremal optimization. *Eur. J. Oper. Res.*, **188**(3):637-651. [doi:10.1016/j.ejor.2007.05.008]

Chen, M.R., Lu, Y.Z., Yang, G.K., 2007. Multiobjective extremal optimization with applications to engineering design. *J. Zhejiang Univ.-Sci. A*, **8**(12):1905-1911. [doi:10.1631/jzus.2007.A1905]

Chen, Y.W., Lu, Y.Z., Chen, P., 2007. Optimization with extremal dynamics for the traveling salesman problem. *Phys. A*, **385**(1):115-123. [doi:10.1016/j.physa.2007.06.014]

Chuang, C.C., Lee, Z.J., 2011. Hybrid robust support vector machines for regression with outliers. *Appl. Soft. Comput.*, **11**(1):64-72. [doi:10.1016/j.asoc.2009.10.017]

Engelbrecht, A.P., 2007. Computational Intelligence: an Introduction (2nd Ed.). John Wiley & Sons, New York.

Friedrichs, F., Igel, C., 2005. Evolutionary tuning of multiple SVM parameters. *Neurocomputing*, **64**(1-4):107-117. [doi:10.1016/j.neucom.2004.11.022]

Hou, S.M., Li, Y.R., 2009. Short-term fault prediction based on support vector machines with parameter optimization by evolution strategy. *Expert Syst. Appl.*, **36**(10):12383-12391. [doi:10.1016/j.eswa.2009.04.047]

Howley, T., Madden, M.G., 2005. The genetic kernel support vector machine: description and evaluation. *Artif. Intell. Rev.*, **24**(3-4):379-395. [doi:10.1007/s10462-005-9009-3]

Hsu, C.W., Chang, C.C., Lin, C.J., 2004. A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University.

Jeng, J.T., 2006. Hybrid approach of selecting hyperparameters of support vector machine for regression. *IEEE Trans. Syst. Man Cybern. B*, **36**(3):699-709. [doi:10.1109/TSMCB.2005.861067]

Lorena, A.C., de Carvalho, A., 2008. Evolutionary tuning of SVM parameter values in multiclass problems. *Neurocomputing*, **71**(16-18):3326-3334. [doi:10.1016/j.neucom.2008.01.031]

Lu, Y.Z., Chen, M.R., Chen, Y.W., 2007. Studies on Extremal Optimization and Its Applications in Solving Real World Optimization Problems. IEEE Symp. on Foundations of Computational Intelligence, p.162-168. [doi:10.1109/FOCI.2007.372163]

Mao, Y., Zhou, X., Pi, D., Sun, Y., Wong, S.T.C., 2005. Parameters selection in gene selection using Gaussian kernel support vector machines by genetic algorithm. *J. Zhejiang Univ.-Sci.*, **6B**(10):961-973. [doi:10.1631/jzus.2005.B0961]

Min, J.H., Lee, Y.C., 2005. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. *Expert Syst. Appl.*, **28**(4):603-614. [doi:10.1016/j.eswa.2004.12.008]

Pai, P.F., Hong, W.C., 2005. Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conv. Manag.*, **46**(17):2669-2688. [doi:10.1016/j.enconman.2005.02.004]

Qiao, J.F., Wang, H.D., 2008. A self-organizing fuzzy neural network and its applications to function approximation and forecast modeling. *Neurocomputing*, **71**(4-6):564-569. [doi:10.1016/j.neucom.2007.07.026]

Saini, L.M., Aggarwal, S.K., Kumar, A., 2010. Parameter optimisation using genetic algorithm for support vector machine-based price-forecasting model in National electricity market. *IET Gener. Transm. Distr.*, **4**(1):36-49. [doi:10.1049/iet-gtd.2008.0584]

Steve, G., 1998. Support Vector Machines Classification and Regression. ISIS Technical Report, Image, Speech, & Intelligent Systems Group, University of Southampton, UK.

Tang, X., Zhuang, L., Jiang, C., 2009. Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization. *Expert Syst. Appl.*, **36**(9):11853-11857. [doi:10.1016/j.eswa.2009.04.015]

Thadani, K., Ashutosh, Jayaraman, V.K., Sundararajan, V., 2006. Evolutionary Selection of Kernels in Support Vector Machines. Int. Conf. on Advanced Computing and Communications, p.19-24. [doi:10.1109/ADCOM.2006.4289849]

Vapnik, V., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York.

Wu, C.H., Tzeng, G.H., Goo, Y.J., Fang, W.C., 2007. A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Syst. Appl.*, **32**(2):397-408. [doi:10.1016/j.eswa.2005.12.008]

Wu, C.H., Tzeng, G.H., Lin, R.H., 2009. A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Syst. Appl.*, **36**(3):4725-4735. [doi:10.1016/j.eswa.2008.06.046]

Wu, Q., 2010. A hybrid-forecasting model based on Gaussian support vector machine and chaotic particle swarm optimization. *Expert Syst. Appl.*, **37**(3):2388-2394. [doi:10.1016/j.eswa.2009.07.057]

Zhang, L., Zhou, W., Jiao, L., 2004. Wavelet support vector machine. *IEEE Trans. Syst. Man Cybern. B*, **34**(1):34-39. [doi:10.1109/TSMCB.2003.811113]

Zhang, X.L., Chen, X.F., He, Z.J., 2010. An ACO-based algorithm for parameter optimization of support vector machines. *Expert Syst. Appl.*, **37**(9):6618-6628. [doi:10.1016/j.eswa.2010.03.067]