Pattern Recognition Letters 45 (2014) 205–210

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Robust regression with extreme support vectors ☆

Wentao Zhu [a], Jun Miao [a], Laiyun Qing [b,*]

[a] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 22 October 2013
Available online 30 April 2014

Keywords:
Extreme Support Vector Regression
Extreme Support Vector Machine
Extreme support vectors
Extreme Learning Machine

ABSTRACT

Extreme Support Vector Machine (ESVM) is a nonlinear robust SVM algorithm based on regularized least squares optimization for binary-class classification. In this paper, a novel algorithm for regression tasks, Extreme Support Vector Regression (ESVR), is proposed based on ESVM. Moreover, kernel ESVR is suggested as well. Experiments show that, ESVR has a better generalization than some other traditional single hidden layer feedforward neural networks, such as Extreme Learning Machine (ELM), Support Vector Regression (SVR) and Least Squares-Support Vector Regression (LS-SVR). Furthermore, ESVR has much faster learning speed than SVR and LS-SVR. Stabilities and robustnesses of these algorithms are also studied in the paper, which shows that the ESVR is more robust and stable.

1. Introduction

Regression is an important topic for machine learning. Classification is the special case of regression, in which the outputs are in the set of {0,1}. Many regression approaches have been proposed, such as Support Vector Regression (SVR) [18] and least squares regression. However, these methods have some drawbacks, e.g., slow learning speed, poor generalization and low robustness [11].

Extreme Learning Machine (ELM) is a successful single hidden layer feedforward neural network for both classification and regression [11]. It has a good generalization with an extremely fast learning speed. Some desirable advantages can be found in ELM, such as extremely fast learning speed and good computational scalability. The essence of ELM is that the hidden layer parameters need not be tuned iteratively and the hidden layer's output connection weights can be simply calculated by least squares optimization [10]. ELM has attracted a great number of researchers and engineers recently for their theoretical and application works [13,12,22]. However, the traditional ELM may encounter ill-posed problems and it is difficult to choose appropriate hidden parameters to avoid such problems [14].

Extreme Support Vector Machine (ESVM) [15] is a kind of single hidden layer feed forward network developed from ELM and Support Vector Machine (SVM). It has not only the same advantages as ELM, such as extremely fast learning speed and that hidden layer parameters can be randomly generated, but also a better generalization than traditional ELM on classification tasks due to its output bias term and regularization scheme. It is a special form of regularization networks [5] derived from SVM. ESVM can be also viewed as an approximation method of SVM. Such approximation leads to fast learning speed. Due to these properties, a lot of researches have been conducted on ESVM [8,6,19,17]. However, ESVM model cannot be applied to multi-class classification and regression tasks directly.

In this paper, the Extreme Support Vector Regression (ESVR) model, a novel single hidden layer feedforward neural network, is proposed for regression tasks based on ESVM. Inspired by ESVM, our ESVR model is a fast approximation method of ϵ-SV regression [18]. Some experimental results show that the ESVR model has a quite good generalization with a high learning speed. Moreover, the proposed ESVR model is quite robust and stable for regression.

This paper is organized as follows. The ESVM algorithm is briefly reviewed in Section 2. Basic ESVR and kernel ESVR are proposed in Section 3. Performances of ESVR compared with ELM, Support Vector Regression (SVR) and Least Squares-Support Vector Regression (LS-SVR) are verified in Section 4. The experiments about the stabilities and robustnesses of such methods are studied in Section 4 as well.

2. Extreme Support Vector Machine

We here briefly review the model of Extreme Support Vector Machine (ESVM). Similar to ELM, ESVM [15] is a kind of single hidden layer feedforward neural network as illustrated in Fig. 1. The input $\mathbf{x} \in \mathbf{R}^m$ is transformed to a feature space by the activation

☆ This paper has been recommended for acceptance by G. Moser.

* Corresponding author. Tel.: +86 10 6260 0522.
E-mail address: lyqing@ucas.ac.cn (L. Qing).

http://dx.doi.org/10.1016/j.patrec.2014.04.016
0167-8655/

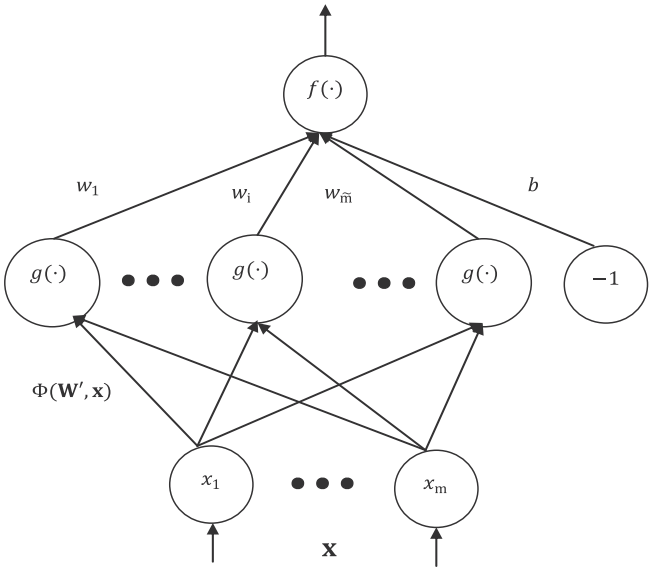

Fig. 1. The structure of ESVM network.

function $\Phi(\mathbf{W}', \mathbf{x})$, which the hidden layer parameters $\mathbf{W}'$ from input layer can be generated randomly. Then the output layer solves a regularized least squares problem in the feature space, where the regularizations are put on both $\mathbf{w}$ and b.

The goal of ESVM is to find an approximate decision boundary of SVM: $\mathbf{w}^T\mathbf{x} - b = \pm 1$, where $\mathbf{w}, b$ are the orientation and the relative location of the decision boundary respectively. The model of ESVM is obtained by replacing the inequality constraints in the traditional SVM model with the equality constraint as

$$\begin{aligned} &\min_{(\mathbf{w},b,\xi)\in\mathbf{R}^{\tilde{m}+1+n}} \frac{C}{2}\|\xi\|_{L_2}^2 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}\right\|_{L_2}^2 \\ &s.t.\ \mathbf{D}(\Phi(\mathbf{W}', \mathbf{A})\mathbf{w} - b\mathbf{e}) + \xi = \mathbf{e}, \end{aligned} \tag{1}$$

where C is the regularization parameter which controls the tradeoff between allowable errors and the minimization L_2 norm of the hidden layer's output weights and bias, ξ is the slack variable of the model, and $\mathbf{D}$ is the diagonal square matrix with the element of 1 or -1 denoting the labels. In Eq. (1), $\mathbf{A}$ is the data sample matrix of size $n \times m$, where each row is one sample $\mathbf{x}$, and $\Phi(\mathbf{W}', \mathbf{A}) = (\Phi(\mathbf{W}', \mathbf{x}_1), \ldots, \Phi(\mathbf{W}', \mathbf{x}_n))^T$. $\Phi(\mathbf{W}', \mathbf{x}) : \mathbf{R}^m \to \mathbf{R}^{\tilde{m}}$ is the feature mapping function in the hidden layer, where m is the dimension of the input data, $\tilde{m}$ is the number of hidden nodes. The most commonly used mapping function is sigmoid function, that is, $\Phi(\mathbf{W}', \mathbf{x}) = sigmoid(\mathbf{W}'^T\mathbf{x})$, $\mathbf{W}'$ is a matrix of size $m \times \tilde{m}$ and can be generated randomly. $\mathbf{e}$ is a vector of size $n \times 1$ which is filled with 1 s.

The model is a quadric programming optimization problem. However, the solution of the model is simply equivalent to calculate the following expression according to [15] as

$$\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix} = \left(\frac{\mathbf{I}}{C} + \mathbf{E}_\Phi^T\mathbf{E}_\Phi\right)^{-1}\mathbf{E}_\Phi^T\mathbf{D}\mathbf{e}, \tag{2}$$

where $\mathbf{E}_\Phi = [\Phi(\mathbf{W}', \mathbf{A}), -\mathbf{e}] \in \mathbf{R}^{n\times(\tilde{m}+1)}$.

ESVM reaches a better generalization than ELM in almost all classification cases and achieve comparable accuracies to SVM [15]. Due to its simple analytical solution, ESVM learns at a quite fast speed. Additionally, the activation function can be constructed explicitly and ESVM has a unified algorithm for both linear and nonlinear mapping function [15]. However, the diagonal label matrix $\mathbf{D}$ must be constructed in the above ESVM model. Besides, $\mathbf{D}$ must be with the element of 1 or -1 in the above deduction, which means that the ESVM model cannot be applied to either multi-class classification or regression tasks directly.

3. Extreme Support Vector Regression

In this section, we extend ESVM from the classification task to regression task. A novel robust model, Extreme Support Vector Regression (ESVR), and a kernel ESVR are proposed.

3.1. Basic Extreme Support Vector Regression

Inspired by the ESVM, ESVR replaces the inequality constraints of the standard ϵ-SV regression with the equality constraint [18,4]. We add the L_2 norm constraints on both $\mathbf{w}$ and b, while Support Vector Regression (SVR) and Least Squares-Support Vector Regression (LS-SVR) [20] have such constraints on $\mathbf{w}$ only. That is one of the reasons that SVR and LS-SVR provide suboptimal solutions compared with that of ESVR [9]. However, different from ESVM, the diagonal target output matrix need not be constructed. The model of ESVR is constructed as

$$\begin{aligned} &\min_{(\mathbf{w},b,\xi)\in\mathbf{R}^{\tilde{m}+1+n}} \frac{C}{2}\|\xi\|_{L_2}^2 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}\right\|_{L_2}^2 \\ &s.t.\ \Phi(\mathbf{W}', \mathbf{A})\mathbf{w} - b\mathbf{e} - \mathbf{y} = \xi, \end{aligned} \tag{3}$$

where $\mathbf{y}$ is the output vector of the sample data matrix $\mathbf{A}$. If $\mathbf{w}$ and b have been obtained, the testing process is to calculate $\hat{y} = \mathbf{w}^T\Phi(\mathbf{W}', \mathbf{x}) - b$ to get the output of the sample.

The lagrangian formula for the model in Eq. (3) is

$$L(\mathbf{w}, b, \xi, \lambda) = \frac{C}{2}\|\xi\|_{L_2}^2 + \frac{1}{2}\left\|\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}\right\|_{L_2}^2 - \lambda^T(\Phi(\mathbf{W}', \mathbf{A})\mathbf{w} - b\mathbf{e} - \mathbf{y} - \xi), \tag{4}$$

where $\lambda \in \mathbf{R}^n$ is the lagrangian multiplier of the model, and λ is also known as support values according to support vector theory. Applying the KKT condition theory [2] to this problem, the solution of ESVR model can be obtained by

$$\begin{aligned} &\frac{\partial L}{\partial \mathbf{w}} = 0 \ \Rightarrow\ \mathbf{w} = \Phi(\mathbf{W}', \mathbf{A})^T\lambda \\ &\frac{\partial L}{\partial b} = 0 \ \Rightarrow\ b = -\mathbf{e}^T\lambda \\ &\frac{\partial L}{\partial \xi} = 0 \ \Rightarrow\ C\xi + \lambda = 0 \\ &\frac{\partial L}{\partial \lambda} = 0 \ \Rightarrow\ \Phi(\mathbf{W}', \mathbf{A})\mathbf{w} - b\mathbf{e} - \xi - \mathbf{y} = 0. \end{aligned} \tag{5}$$

From the third expression of Eq. (5), the support value λ is proportional to the error ξ. That is to say, almost all support values are nonzero values in ESVR. Therefore, there is no traditional support vector concepts in ESVR, or all the data points are support vectors.

We obtain the following equation by substituting the first three expressions in the last expression of Eq. (5) as

$$\left(\frac{\mathbf{I}}{C} + \mathbf{E}_\Phi\mathbf{E}_\Phi^T\right)\lambda = \mathbf{y}, \tag{6}$$

where $\mathbf{E}_\Phi = [\Phi(\mathbf{W}', \mathbf{A}), -\mathbf{e}] \in \mathbf{R}^{n\times(\tilde{m}+1)}$.

Whether the expression, $\frac{\mathbf{I}}{C} + \mathbf{E}_\Phi\mathbf{E}_\Phi^T$, is reversible or not may affect the solution of the above equation. We will discuss them respectively.

In the case of $n < \tilde{m} + 1$, $\frac{\mathbf{I}}{C} + \mathbf{E}_\Phi\mathbf{E}_\Phi^T$ is reversible,

$$\lambda = \left(\frac{\mathbf{I}}{C} + \mathbf{E}_\Phi\mathbf{E}_\Phi^T\right)^{-1}\mathbf{y}. \tag{7}$$

Substituting λ in the first two equations of the KKT conditions, we obtain the analytical solution of $\mathbf{w}$ and b as

$$\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix} = \mathbf{E}_\Phi^T\lambda = \mathbf{E}_\Phi^T\left(\frac{\mathbf{I}}{C}+\mathbf{E}_\Phi\mathbf{E}_\Phi^T\right)^{-1}\mathbf{y}. \tag{8}$$

If $n > \tilde{m}+1$, the above inverse operation of a $n\times n$ matrix will have heavy computational cost. By using Sherman–Morrison–Woodbury formula [7], we can obtain the following expression as

$$\lambda = C\left[\mathbf{I}-\mathbf{E}_\Phi\left(\frac{\mathbf{I}}{C}+\mathbf{E}_\Phi^T\mathbf{E}_\Phi\right)^{-1}\mathbf{E}_\Phi^T\right]\mathbf{y}. \tag{9}$$

Then we obtain the expression for $\mathbf{w}$ and b as

$$\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix} = \mathbf{E}_\Phi^T\lambda = \left(\frac{\mathbf{I}}{C}+\mathbf{E}_\Phi^T\mathbf{E}_\Phi\right)^{-1}\mathbf{E}_\Phi^T\mathbf{y}. \tag{10}$$

From the above discussion, the proposed ESVR model takes advantages of SVR, LS-SVR and ELM. First, ESVR utilizes random projection to increase the dimensions of the original data samples as in ELM. High dimensions may help the following regression task. The random projection also makes the ESVR not over rely on some dimensions of the data, which makes ESVR more robust. Second, ESVR adds the output layer bias, which gets a more flexible model. Third, ridge regression scheme in ESVR provides a more robust solution than traditional ELM. Besides, according to [9], structure risk minimization constraints are added on both $\mathbf{w}$ and b in ESVR, which makes that the solutions of SVR and LS-SVR are just suboptimal solutions of ESVR. Finally, the analytical solution of ESVR is to compute some matrix multiplications, which leads to fast learning speed.

The architecture of ESVR is a typical structure of single hidden layer feedforward regularization network with a biased output. According to single hidden layer feedforward neural network theory [10], the hidden layer parameters can be generated randomly. Such randomness will be utilized to choose the nonlinear feature mapping function parameters. Then the algorithm of ESVR can be explicitly concluded as Algorithm 1.

Algorithm 1. Training Process of Extreme Support Vector Regression (ESVR)

Input: The training set $\{(\mathbf{x}_i,y_i)|\mathbf{x}_i\in\mathbf{R}^m, y_i\in R, i=1,\ldots,n\}$; The feature mapping function $g(x)$; the dimension $\tilde{m}$ after feature mapping; the regularization parameter C;

Output: The parameters $\mathbf{w}$ and b of the regression model;

1: Randomly generate a hidden layer parameter matrix $\mathbf{W}'\in\mathbf{R}^{\tilde{m}\times(m+1)}$, and obtain the sample matrix $\mathbf{A}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]^T$, target vector $\mathbf{y}=[y_1,\ldots,y_n]^T$ from the training set. The expression of $\Phi(\mathbf{W}',\mathbf{A})$ can be calculated as

$$\Phi(\mathbf{W}',\mathbf{A}) = \begin{bmatrix}\Phi(\mathbf{W}',\mathbf{x}_1)^T\\ \vdots\\ \Phi(\mathbf{W}',\mathbf{x}_n)^T\end{bmatrix} = \begin{bmatrix}g(\mathbf{w}_1'^T\mathbf{x}_1)\cdots g(\mathbf{w}_{\tilde{m}}'^T\mathbf{x}_1)\\ \vdots\ddots\vdots\\ g(\mathbf{w}_1'^T\mathbf{x}_n)\cdots g(\mathbf{w}_{\tilde{m}}'^T\mathbf{x}_n)\end{bmatrix}; \tag{11}$$

2: Generate $\mathbf{E}_\Phi$ by calculating $[\Phi(\mathbf{W}',\mathbf{A}),-\mathbf{e}]$, where $\mathbf{e}$ is a $n\times 1$ vector full of element 1;

3: Compute $\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}$ by expression (8) or (10) with the regularization parameter C;

3.2. Kernel ESVR

We can easily extend ESVR to kernel ESVR. The kernel matrix for ESVM can be defined by Mercer's rule if the feature mapping function $\Phi(\mathbf{W}',\mathbf{x})$ is not given. We define a kernel matrix for ESVM similar to other kernels as

$$\begin{aligned}\Omega_{ESVM} &= \mathbf{E}_\Phi\mathbf{E}_\Phi^T\\ \Omega_{ESVM_{i,j}} &= \Phi(\mathbf{W}',\mathbf{x}_i)\Phi(\mathbf{W}',\mathbf{x}_j)^T+1 = K(\mathbf{x}_i,\mathbf{x}_j)+1,\end{aligned} \tag{12}$$

where $K(\mathbf{x}_i,\mathbf{x}_j)$ is the user defined kernel function.

The decision function of kernel ESVR can be obtained by the following analytical equation as

$$f(\mathbf{x}) = [\Phi(\mathbf{W}',\mathbf{x}),-1]\mathbf{E}_\Phi^T\left(\frac{\mathbf{I}}{C}+\mathbf{E}_\Phi\mathbf{E}_\Phi^T\right)^{-1}\mathbf{y} = \begin{bmatrix}K(\mathbf{x},\mathbf{x}_1)+1\\ \vdots\\ K(\mathbf{x},\mathbf{x}_m)+1\end{bmatrix}^T\left(\frac{\mathbf{I}}{C}+\Omega_{ESVM}\right)^{-1}\mathbf{y}. \tag{13}$$

If the kernel function $K(\mathbf{u},\mathbf{v})$ is given, we need not to choose the feature mapping function $\Phi(\mathbf{W}',\mathbf{x})$, the hidden node number $\tilde{m}$ and hidden parameters $\mathbf{W}'$. Basic ESVR is can also been considered as a special form of kernel ESVR as $K(\mathbf{x}_i,\mathbf{x}_j)=\Phi(\mathbf{W}',\mathbf{x}_i)^T\Phi(\mathbf{W}',\mathbf{x}_j)$.

4. Experiments

In this section, the performances of ESVR are compared with ELM, SVR and LS-SVR on some benchmark regression datasets. Robustnesses of these methods are studied as well.

4.1. Experimental environments

All the simulations for ESVR, ELM, SVR and LS-SVR were carried out in MATLAB R2010a environment running on a Xeon E7520, 1.87 GHZ CPU. The codes used for ELM, SVR and LS-SVR were from [11,3,21] respectively.

In order to extensively verify the performance of ESVR, ELM, SVR and LS-SVR, twelve datasets of different sizes and dimensions were downloaded from UCI Machine Learning Repository [1] and StatLib library [16]. These datasets can be divided into three categories according to different sizes and feature dimensions. Baskball, Strike, Cloud, and Autoprice are of small sizes and low dimensions. Pyrim, Housing, Bodyfat, and Cleveland are of small sizes and medium dimensions. Balloon, Quake, Space-ga, and Abalone are of large sizes and low dimensions.

In the experiments, the kernel function used was the RBF function $G(\mathbf{a},\gamma,\mathbf{x}) = \exp\left(-\gamma\|\mathbf{x}-\mathbf{a}\|_{L_2}^2\right)$. Three fold cross validation was conducted to select parameters. The regularization factor C and kernel parameter γ of SVR, LS-SVR and kernel ESVR were obtained from the $\log_2$ space from -25 to 25. All the datasets were normalized into $[-1,1]$ before the regression process.

4.2. Performances on benchmark datasets

Experiments between basic ESVR and ELM on the above twelve different benchmarks were carried out. Nonlinear models with sigmoidal additive feature map function $\Phi(\mathbf{a},b,\mathbf{x}) = \frac{1}{1+\exp(-(\mathbf{a}\mathbf{x}+b))}$ were used. Ten rounds of experiments with the same parameters were conducted to obtain an average performance evaluation in each fold due to randomly selected parameters in the hidden layer. Fig. 2 is the testing Root Mean Square Errors (RMSEs) of ESVR and ELM with different number of hidden nodes on six of the twelve real world datasets.

Fig. 2 shows that the testing RMSE of ESVR is lower than that of ELM. The RMSE results of the experiments reveal that the generalization of ESVR is better than that of ELM. The output bias and regularization scheme in the ESVR model mainly contribute to this. Moreover, we can observe that the performances of ELM vary greatly with the number of hidden nodes. ESVR eases such overfitting problem by exploring regularization. It can also be seen

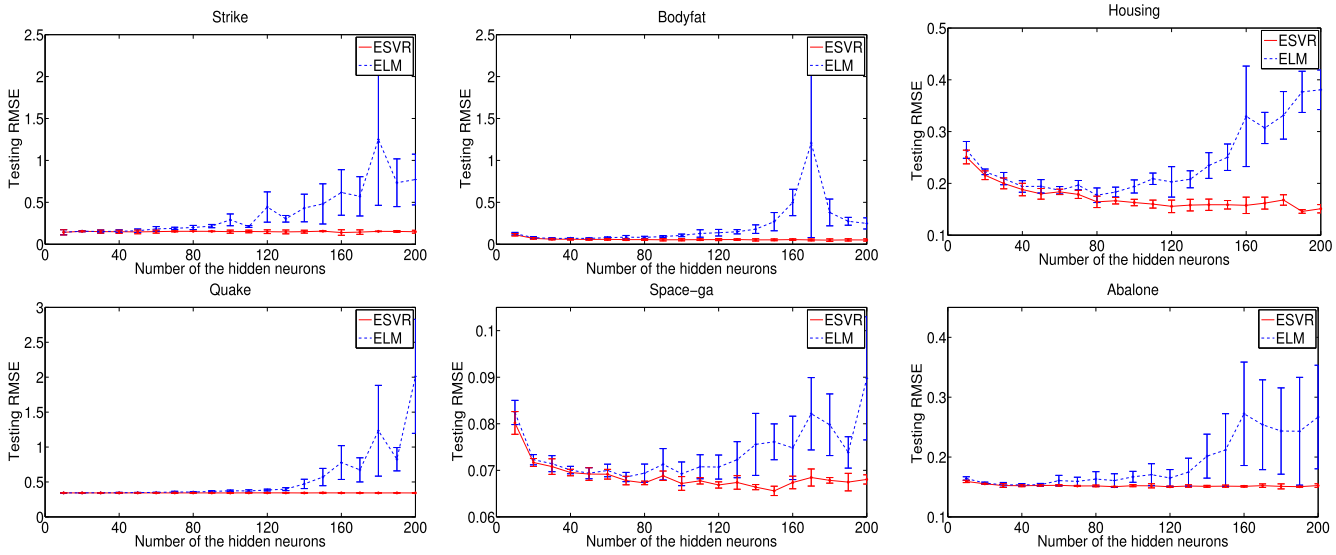

Fig. 2. Test RMSEs of basic ESVR and ELM.

that the standard deviations of ESVR are much less, which mainly because that the regularization term added makes ESVR less sensitive to the different $\mathbf{W}'$ chosen randomly.

The second experiment was conducted to compare basic ESVR, kernel ESVR (KESVR), SVR and LS-SVR. Table 1 shows the results of those methods. Some interested items, including testing RMSE, training time, testing time and the proportion of support vectors in SVR, were recorded as the evaluations of generalization, learning speed and prediction speed respectively. The best testing accuracies and times results for different datasets were emphasized in bold face.

It can be seen form Table 1 that the testing RMSEs of basic or kernel ESVR are the best ones in most cases. These two methods are just different in the different nonlinear kernel functions. These results reveal that, ESVR has a higher accuracy than that of SVR and LS-SVR. The experimental results demonstrate that the approximation method utilized in the ESVR model really works. Furthermore, the training time of ESVR is much less than that of SVR and LS-SVR. The advantage of basic ESVR algorithm about learning speed can be demonstrated explicitly when the sizes of these datasets become larger. The reason is that the solution of ESVR is an analytical equation and the learning process is simply to solve a least square problem. From Table 1, the testing time of SVR is less than that of LS-SVR and kernel ESVR. This is because that the number of support vectors in SVR is much less than the number of samples used by least squares methods in LS-SVR and kernel ESVR. The testing time of basic ESVR is also less because the number of hidden nodes used is much less than the number of samples here. The proportion of support vectors in SVR is recorded as the average percentage that support vectors comprised in the training samples. Because LS-SVR, kernel ESVR and ESVR are least squares based methods, there is no traditional support vector concept, that is, all the training samples are support vectors. Therefore the testing time of LS-SVR is longer. As for ESVR, it does not matter as the prediction is done on the original space rather than the dual space as $\hat{y} = \mathbf{w}^T\Phi(\mathbf{W}', \mathbf{x}) - b$ in testing, which involves only dot productions in $\Phi(\mathbf{W}', \mathbf{x}) = \left(g(\mathbf{w}_1'^T\mathbf{x}), \ldots, g(\mathbf{w}_{\dot{m}}'^T\mathbf{x})\right)^T$ and $\mathbf{w}^T\Phi(\mathbf{W}', \mathbf{x})$. Therefore the testing time is acceptable.

The third experiment was designed to verify the stability of the ESVR, kernel ESVR, SVR and LS-SVR. In the experiment, the performances of these models with different parameters were compared on the Housing dataset from UCI datasets.

The parameters (C, γ) of SVR and LS-SVR and kernel ESVR were selected from -25 to 25 in the $\log_2$ space where C is the regularization parameter and γ is the parameter used in kernel function. The number of hidden nodes, $\dot{m}$, in basic ESVR was selected from 50 to 2050 with the step 50. The parameter C of the nonlinear basic ESVR was selected from -25 to 25 in the $\log_2$ space.

Fig. 3 shows that the results of SVR, LS-SVR, kernel ESVR with Gaussian kernel function and basic ESVR with given sigmoidal additive nodes. We can see that kernel ESVR, SVR and LS-SVR

Table 1
Experimental results of SVR, LS-SVR, kernel ESVR and basic ESVR.

Algorithms	SVR				LS-SVR			KESVR			Basic ESVR		
	Testing RMSE	Training time (s)	Testing time (s)	% SVs	Testing RMSE	Training time (s)	Testing time (s)	Testing RMSE	Training time (s)	Testing time (s)	Testing RMSE	Training time (s)	Testing time (s)
Baskball	0.2567	0.1029	0.0009	84	0.2568	0.0049	0.0017	0.2591	0.0007	0.0005	**0.2478**	0.0172	<**0.0001**
Cloud	0.1729	0.0774	0.0007	72	0.1810	0.0065	0.0028	0.1827	0.0014	**0.0006**	**0.1458**	0.0057	0.0010
Autoprice	0.1381	0.1328	0.0008	61	**0.1359**	0.0072	0.0040	0.1371	0.0021	**0.0007**	0.1585	0.0094	0.0016
Strike	0.1443	0.9707	**0.0028**	44	0.1472	0.0541	0.0098	0.1478	0.0274	0.0081	**0.1395**	0.0417	0.0130
Pyrim	0.2151	0.0336	0.0005	63	0.2159	0.0051	0.0027	**0.2095**	0.0015	0.0005	**0.1963**	0.0104	<**0.0001**
Bodyfat	0.0514	0.0485	**0.0011**	15	0.0502	0.0128	0.0055	**0.0496**	0.0037	0.0014	**0.0490**	0.0260	0.0063
Cleveland	0.4267	0.2690	0.0026	86	0.4333	0.0147	0.0062	**0.4260**	0.0052	0.0021	**0.4164**	0.0130	**0.0005**
Housing	0.1469	0.7729	**0.0035**	59	0.1458	0.0455	0.0102	**0.1441**	0.0162	0.0066	**0.1458**	0.0422	0.0063
Balloon	0.0242	7.8253	**0.0025**	2	0.0099	1.0798	0.0977	**0.0056**	0.4931	0.0971	**0.0095**	0.1224	0.0297
Quake	0.3438	205.7426	0.0983	83	**0.3425**	2.4292	0.1202	0.3427	0.5701	0.1058	0.3431	0.0141	**0.0073**
Space-ga	0.0654	92.1705	0.1039	37	0.0665	2.5293	0.2358	**0.0651**	1.3030	0.1782	0.0656	0.1573	**0.0344**
Abalone	0.1519	250.4772	0.3357	63	**0.1486**	9.6423	0.1486	0.1493	4.8523	0.5162	0.1503	0.1630	**0.0474**

algorithms are sensitive to their parameters as the performances of these algorithms vary greatly with their parameters (C, γ) from the figure. ESVR is quite stable over their parameters $(C, \tilde{m})$. The reason is probably that ESVR utilizes the randomly generated hidden parameters. The combination of randomness and regularization makes the ESVR algorithm much more stable from the view of experimental results.

At last, experiments of robustness were conducted on Baskball dataset. One-dimension or two-dimension gaussian noises with 0

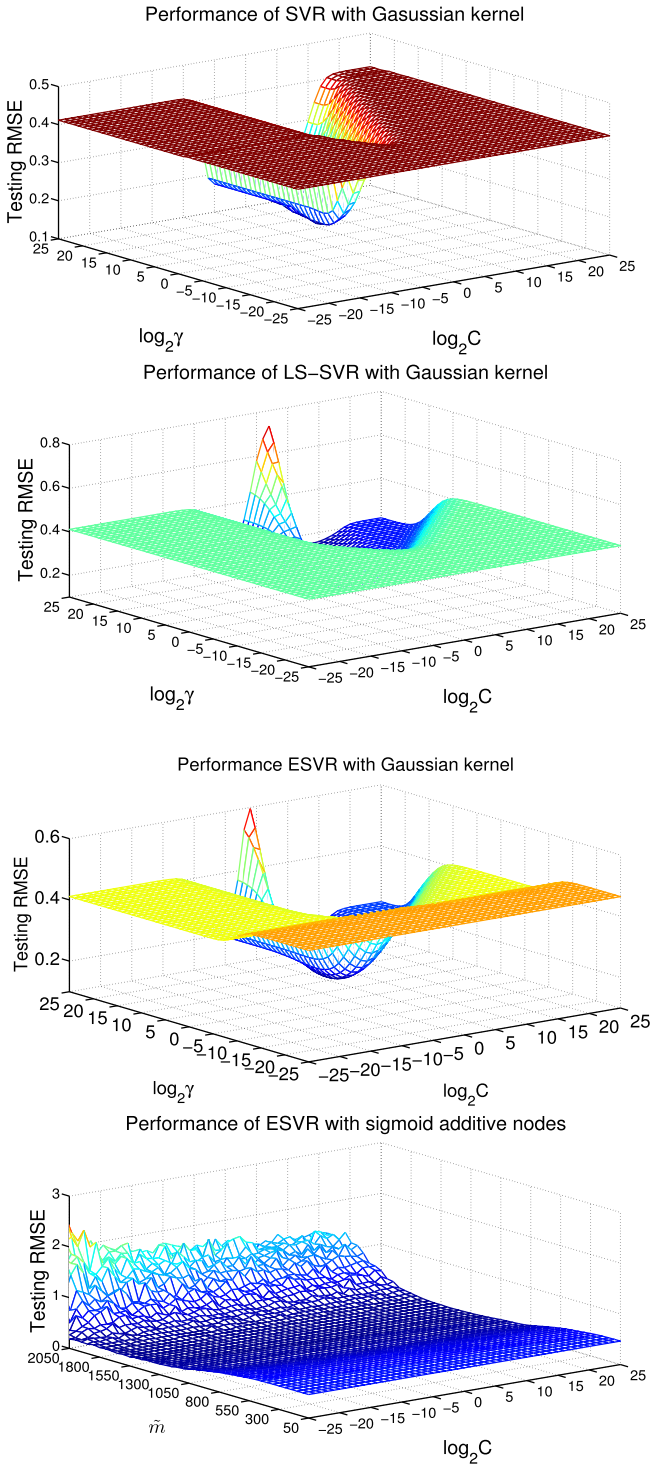

Fig. 3. Performances of SVR, LS-SVR and kernel ESVR with Gaussian kernels are sensitive to parameters (C, γ); performances of basic ESVR with sigmoidal additive nodes are more stable to parameters $(C, \tilde{m})$.

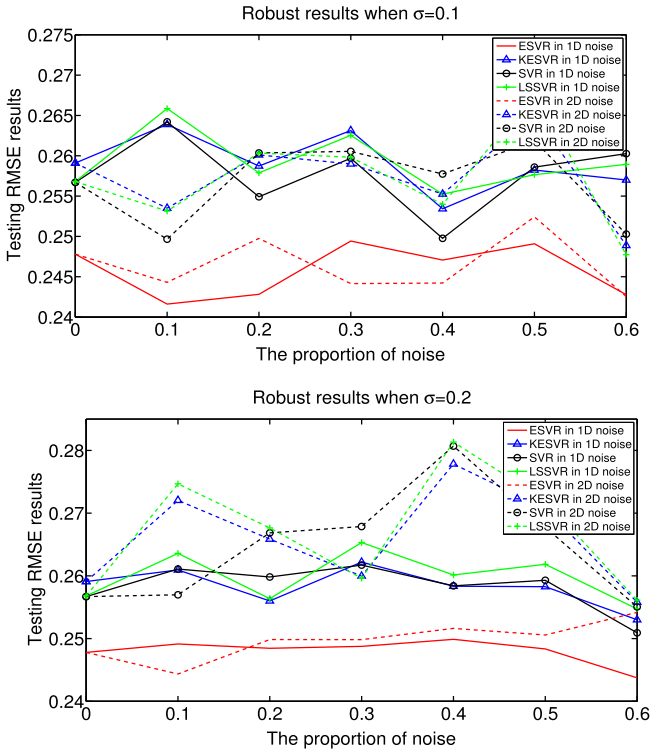

Fig. 4. Robust experimental results among basic ESVR, kernel ESVR, LS-SVR and SVR.

means were added into the first one dimension or the first two dimensions of samples respectively. Experiments were conducted with different variances (0.1 and 0.2) and different proportions of noise (from 0.1 to 0.6). Other experimental settings were the same as before.

Fig. 4 shows that the testing RMSEs of basic ESVR, kernel ESVR, LS-SVR and SVR with different datasets. We can see that the performances of basic ESVR are much more stable (with the least variance against the noise) with the best generalization from the figure. This phenomenon suggests that the ESVR algorithm is more robust. The reason for the phenomenon is that random projection can ease over relying on some dimensions of the data.

5. Conclusions

This paper proposes a novel robust regression method with extreme support vectors-ESVR by taking advantage of ELM and SVR. Inspired by ESVM, ESVR is a novel approximation ϵ-SV regression method based on regularized least squares and it is also a variant of ELM algorithm from the point of view of computation. Moreover, it is easily to incorporate kernel function to ESVR model. The experimental results show that, ESVR has a better generalization performance with much higher learning speed. In addition, ESVR is more stable and robust.

Acknowledgment

This research is partially sponsored by Natural Science Foundation of China (Nos. 61070116, 61070149, 61175115 and 61272320) and President Fund of Graduate University of Chinese Academy of Sciences (No. Y35101CY00).

References

[1] K. Bache, M. Lichman, UCI machine learning repository, 2013. <http://archive.ics.uci.edu/ml>.

[2] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[3] S. Canu, Y. Grandvalet, V. Guigue, A. Rakotomamonjy, Svm and kernel methods matlab toolbox, Perception Systemes et Information, INSA de Rouen, Rouen, France, 2005. <http://asi.insarouen.fr/enseignants/~arakotom/toolbox/index.html>.
[4] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learning 20 (1995) 273–297.
[5] T. Evgeniou, M. Pontil, T. Poggio, Regularization networks and support vector machines, Adv. Comput. Math. 13 (2000) 1–50.
[6] B. Frénay, M. Verleysen, Using svms with randomised feature spaces: an extreme learning approach, in: Proceedings of the 18th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 2010, pp. 28–30.
[7] G.H. Golub, C.F. Van Loan, Matrix Computations, vol. 3, JHUP, 2012.
[8] Q. He, C. Du, Q. Wang, F. Zhuang, Z. Shi, A parallel incremental extreme svm classifier, Neurocomputing 74 (2011) 2532–2540.
[9] G.B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (2010) 155–163.
[10] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Proceedings of IEEE 2004 International Joint Conference on Neural Networks, IEEE, 2004, pp. 985–990.
[11] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[12] H.T. Huynh, Y. Won, Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks, Pattern Recogn. Lett. 32 (2011) 1930–1935.
[13] A. Iosifidis, A. Tefas, I. Pitas, Dynamic action recognition based on dynemes and extreme learning machine, Pattern Recogn. Lett. (2012).
[14] N.Y. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Networks 17 (2006) 1411–1423.
[15] Q. Liu, Q. He, Z. Shi, Extreme support vector machine classifier, in: Advances in Knowledge Discovery and Data Mining, Springer, 2008, pp. 222–233.
[16] M. Mike, Statistical datasets, 1989. <http://lib.stat.cmu.edu/datasets/>.
[17] P.F. Pai, M.F. Hsu, An enhanced support vector machines model for classification and rule generation, in: Computational Optimization, Methods and Algorithms, Springer, 2011, pp. 241–258.
[18] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222.
[19] A. Subasi, A decision support system for diagnosis of neuromuscular disorders using dwt and evolutionary support vector machines, Signal Image Video Process. (2013) 1–10.
[20] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.
[21] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least Squares Support Vector Machines, World Scientific, 2002. http://www.esat.kuleuven.be/sista/lssvmlab/.
[22] D. Yu, L. Deng, Efficient and effective algorithms for training single-hidden-layer neural networks, Pattern Recogn. Lett. 33 (2012) 554–558.