Function Approximation Based on Twin Support Vector Machines

Chengfu Yang
Computational Intelligence Laboratory
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu 610054, P.R.China
Sichuan University of Arts and Science
Dazhou 635000, P.R.China
chengfu.yang.uestc@gmail.com

Zhang Yi, Lin Zuo
Computational Intelligence Laboratory
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu 610054, P.R. China
{zyi.cilab, linlinzuo}@gmail.com

***Abstract*—A new function approximation algorithm based on Twin Support Vector Machines (TSVM) is presented in this paper. Support Vector Regression (SVR) has been shown to have good robust properties against noise in function approximation, however, the overfitting phenomena cannot be eliminated if the parameters used in SVR are improperly selected, and the selection of various parameters is not straightforward. In this paper, we use the properties of TSVM to solve this problem, that it will generate two nonparallel planes such that each plane is closer to one of the two classes and is as far as possible from the other. The experiments show good performances without additional computing time by using traditional test functions, data with noise and ambiguous training data.**

I. Introduction

The goal of the function approximation is to determine the underlying function $f(.)$ that can denote the relation between the inputs and observations, when given a finite set of training data points, $X = (x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$ with $x_i \in R^n$ and $y_i \in R$. The approximated function must be with high generalization and low overfitting, as well as can predict the outputs for inputs not contained in the training data set. The interpolation and regression, such as linear interpolation, cubic interpolation, cubic splines interpolation and neural networks regression, are the common methods for solving this kind of problem. The shortcoming of the interpolation methods are that they have limited extrapolation abilities and they become more difficult with increasing of the dimensionality of the problem. The Artificial Neural Networks (ANN) approaches [1][11] made progress on the high dimensional regression problem, but the architecture of ANN is difficult to be determined and the learning algorithms easily get into local minima while training.

Recently, an attractive method named Support Vector Machines (SVM) has been used in high-dimensional function approximation. The SVM is based on the Structural Risk Minimization (SRM) principles and statistical learning theories [4], which has the advantages of global minima and well generalization abilities. Large number of experiments showed that it is easy to recognize high-dimension identities using a small bases constructed from the selected support vectors [2]. Normally, the Support Vector Regression (SVR) [3] is often employed to solve the problem of function approximation and regression estimation. The most important advantage for using SVR in function approximation is that the number of free parameters in the function approximation scheme is equal to the number of support vectors which can be obtained by defining the width of a tolerance band [6]. Thus, the function approximation accuracy is only depended on the number of the support vectors, not related to the dimensionality of the input space or other factors as in the cases of multilayer feedforward neural networks. With the application of kernels, the SVR can solve any kinds of function approximation in theory. The Least Squares Support Vector Machine (LSSVM), introduced in [5], is a modification to the standard formulation of SVM[4]. Via the Karush-Kuhn-Tucker conditions, the optimization problem for the LSSVM reduces to the solution of a set of linear equations. This gives us the opportunity to solve large problem without applying iterative chunking. On the other hand, the disadvantage is that the sparseness of the solution is lost. In [7], the function approximation based on LSSVM has been given and good performance were shown in the literature.

The function approximation based on SVM has been shown to have good robust properties against noise. However, overfitting phenomena may still occur when the parameters used in SVM are improperly selected. At the same time, the selection of various parameters is not straightforward [6]. How to control the interference of the noise, outlier and ambiguous points are the hot research topics [6][8][9][11]. This paper proposed a new function approximation algorithm based on TSVM [12] to overcome these shortcoming. The TSVM comes from the generalized eigenvalue Multisurface Proximal Support Vector Machine (MPSVM)[10]. TSVM aims at generating two non-parallel planes such that each plane is closer to one of the two classes and is as far as possible from the other [12]. This property makes TSVM is feasible to be used in function approximation. By regarding the data samples as one class and creating another class relative to the data samples, we can get two planes, one is as close as the data samples according to the property of TSVM which can be regarded as the function approximated from the given data samples. The proposed algorithm gets high generalizations and low computational complexities because of the training principles of TSVM.

The rest of the paper is organized as follows: Section II briefly dwells on TSVM and also introduces the SVR for the comparisons in the next sections. In the Section III, the function approximation system based on TSVM is detailed. Some experimental results are shown in Section IV and the conclusions is included in Section V.

Some notations about this paper: All vectors are column vectors unless transposed to a row vector by a prime superscript '. The $x'y$ or $x{\cdot}y$ denotes the scalar (inner) product of the vectors x and y in the n-dimensional real space R^n. The $\|x\|$ denotes the 2-norm of x. For a matrix $B \in R^{m\times n}$, B_i is the ith row of B and $B_{.j}$ is the jth column of B. A column vector of ones of arbitrary dimension will be denoted by e and the identity matrix of arbitrary order will be denoted by I. For the matrices $A \in R^{m\times n}$ and $B \in R^{n\times k}$, a kernel $K(A,B)$ maps $R^{m\times n} \times R^{n\times k}$ into $R^{m\times k}$.

II. Twin Support Vector Machines

A. Linear and Nonlinear TSVM

In this subsection, we give a brief outline of linear and nonlinear TSVM [12]. First, some notations will be used in this section are concluded in here. The data points belonging to class 1 and -1 are represented by matrices $A_{m1\times n}$ and $B_{m2\times n}$, respectively. The $m1$ is the number of data samples belonging to the class 1 and the $m2$ is the number of the data samples belonging to the class 2. The n is the dimension of the data points.

The TSVM is similar to MPSVM [10] in spirit that they all obtain nonparallel planes around which the data points of the corresponding class get clustered. However, the TSVM has the formulation of a typical SVM, except that only opposite patterns appear in the constraints of the optimization problem. These properties imply that we have many advantages on carrying out the TSVM and saving computing time.

The linear TSVM classifier is to solve the following pair of quadratic programming problems.

$$\begin{cases} \min\limits_{(w_1,b_1,q_2)} & \frac{1}{2}\|A\cdot w_1 + e_1\cdot b_1\|^2 + c_1 e_2\cdot q_2 \\ s.t. & -(B\cdot w_1 + e_2\cdot b_1) + q_2 \geq e_2,\ \ q_2 \geq 0 \end{cases} \quad (1)$$

$$\begin{cases} \min\limits_{(w_2,b_2,q_1)} & \frac{1}{2}\|B\cdot w_2 + e_2\cdot b_2\|^2 + c_2 e_1\cdot q_1 \\ s.t. & (A\cdot w_2 + e_1\cdot b_2) + q_1 \geq e_1,\ \ q_1 \geq 0 \end{cases} \quad (2)$$

where c1, c2>0 are parameters and e1 and e2 are vectors of ones of appropriate dimensions.

The algorithm finds two hyperplanes, TSVM1 for class 1 as $x^T w_1 + b_1 = 0$ and TSVM2 for class 2 as $x^T w_2 + b_2 = 0$ in (1) and (2). And the data points are classified according to which hyperplane the given point is closest to. A simulation for linear TSVM is as Fig. 1.

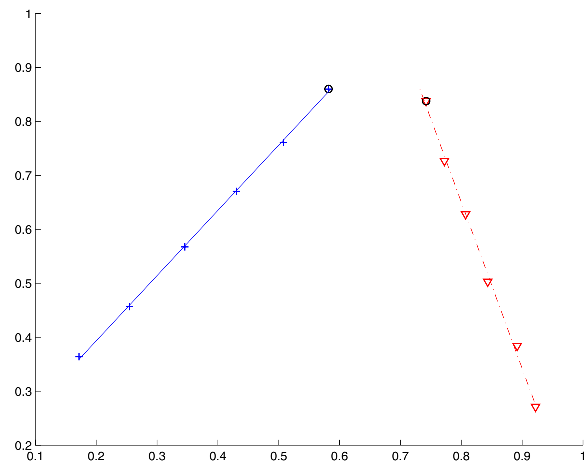


Fig. 1. TSVM for the linear distribution data points with the linear kernel. The data points with circle is support vectors. Two nonparallel planes are close to the respective data samples.

When the kernel tricks are applied in the formulation of TSVM, we can get the algorithm of nonlinear TSVM named KTSVM. According to the [12], the KTSVM is to solve the following pair of quadratic programming problems.

$$\begin{cases} \min\limits_{(w_1,b_1,q_2)} & \frac{1}{2}\|K(A,C^T)w_1 + e_1\cdot b_1\|^2 + c_1 e_2\cdot q_2 \\ s.t. & -(K(B,C^T)w_1 + e_2\cdot b_1) + q_2 \geq e_2,\ \ q_2 \geq 0 \end{cases} \quad (3)$$

$$\begin{cases} \min\limits_{(w_2,b_2,q_1)} & \frac{1}{2}\|K(B,C^T)w_2 + e_2\cdot b_2\|^2 + c_2 e_1\cdot q_1 \\ s.t. & (K(A,C^T)w_2 + e_1\cdot b_2) + q_1 \geq e_1,\ \ q_1 \geq 0 \end{cases} \quad (4)$$

where c1, c2>0 are parameters and e1 and e2 are vectors of ones of appropriate dimensions. $K(,)$ is the selected kernel and $C = [A; B]$ which dimension is $(m1 + m2) \times n$.

The algorithm finds two hyperplanes, KTSVM1 for class 1 as $K(x^T, C^T)w_1 + b_1 = 0$ and KTSVM2 for class 2 as $K(x^T, C^T)w_2 + b_2 = 0$ in (3) and (4). And the data points are classified according to which hyperplane the given point is closest to. A simulation for nonlinear TSVM is as Fig. 2.

B. Support Vector Regression

In this subsection, we will describe the Support Vector Regression (SVR) [3] in order to compare it with our work. The SVR is a very good tool for function approximation. However, as say it in the previous section, the SVR has many shortcoming in the application of function approximation. The most serious is the overfitting problem when the parameters are not properly selected. Some literatures are focus on this topic and get many developments in this field [6][9][11].

Given the training data set $T = (x_1, y_1), ..., (x_m, y_m) \in (X \times Y)^m$, $x_i \in X = R^n$, $y_i \in Y = R$, $i = 1, ..., m$, selecting

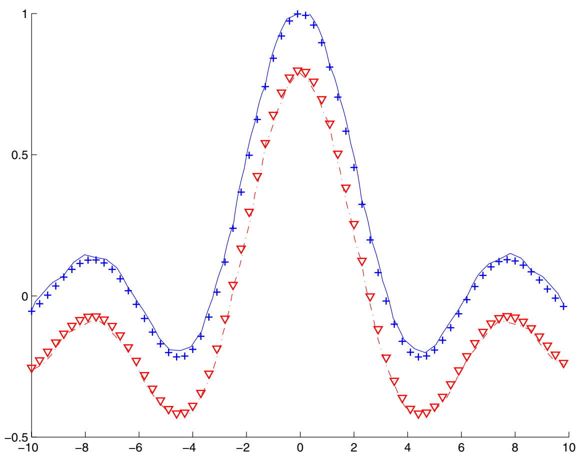


Fig. 2. TSVM for the nonlinear distribution data points with the Exponential RBF kernel $K(u,v) = \exp(\sqrt{((u-v)*(u-v)')}/(2*\sigma^2))$. Here the $\sigma = 4$. Two nonparallel supper planes are close to the respective data samples.

the suitable ε, C and the kernel $K(x,x')$, the SVR algorithm is to solve the following optimization problem.

$$\begin{cases} \min\limits_{\alpha^* \in R^{2m}} & \frac{1}{2}\sum\limits_{i,j=1}^{m}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i,x_j) + \\ & \varepsilon\sum_{i=1}^{m}(\alpha_i^* + \alpha_i) - \sum_{i=1}^{m} y_i(\alpha_i^* - \alpha_i) \\ s.t. & \sum\limits_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0, \\ & 0 \le \alpha_i^*, \alpha_i \le \frac{C}{m}, i = 1,2,...,m. \end{cases} \quad (5)$$

And we can get the optimization results from the (5): $\bar{\alpha} = (\bar{\alpha}_1, \bar{\alpha}_1^*, ..., \bar{\alpha}_m, \bar{\alpha}_m^*)^T$. Then, the decision-function is as (6).

$$f(x) = \sum_{i=1}^{m}(\bar{\alpha}_i^* - \bar{\alpha}_i)K(x_i,x) + \bar{b} \quad (6)$$

and

$$\bar{b} = y_i - \sum_{i=1}^{m}(\bar{\alpha}_i^* - \bar{\alpha}_i)(x_i \cdot x_j) \pm \varepsilon. \quad (7)$$

III. Function Approximation Based on Twin Support Vector Machines

In this section, the steps of function approximation system based on TSVM are proposed. And some details in the programming are depicted. The framework of the system is as Fig. 3.

The main idea is based on the properties of TSVM that is, when we get two planes with TSVM, each data point will be as close as to one plane and as far as to another plane. Then when the TSVM is used in function approximation, the data samples are regarded as one class and another class is generated from the data samples. The plane which is closer to the data samples is the function underlying the data samples.

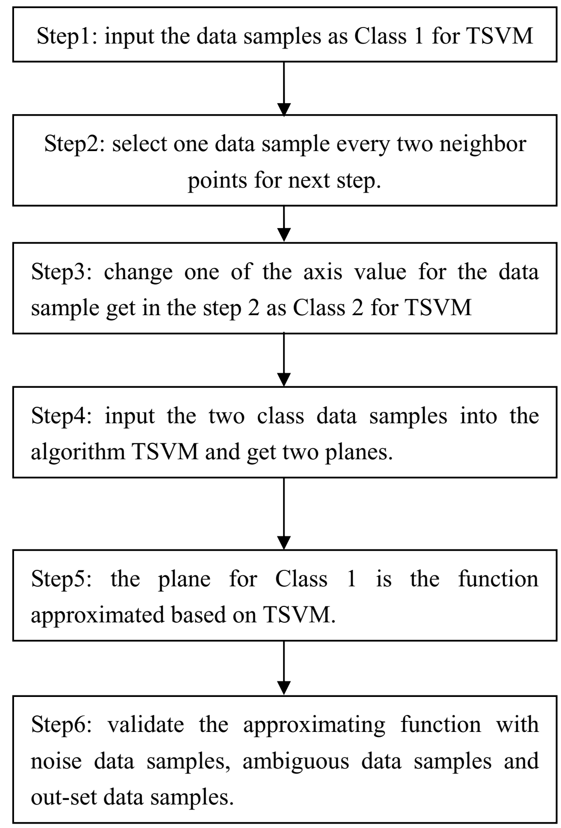


Fig. 3. The function approximation system based on TSVM.

How to generate the data points of class $2th$ for TSVM according to the given data samples is a very pivotal problem in the proposed system. In order to depress the interference of the noise data points, we obtain one point in every two neighbor data samples. This also increases the speed of TSVM. And after changing one of the axis value of fetching out data samples, it will be the data sample of class $2th$ in TSVM. The method of changing axis value is also important for the system to get more accurate estimation results. In this paper, we adopt the method of changing the axis value along one direction, such as y axis direction. For example, in the experiment of multi-value function approximation, the method is along the direction of the gradient decreasing to get the class 2th data points.

The kernel used in this work is Exponential RBF, $K(u,v) = \exp(\sqrt{((u-v)*(u-v)')}/(2*\sigma^2))$. A large of experiments show that this kernel is very suitable to the function approximation system based on TSVM. However, the kernel is also sensitive to the isolated data points.

To the noise data points and ambiguous data points, if they are close to the supper plane in a given threshold, we can get the function value of them using the supper plane. To the data points not included in the training set, we call them out-set data points, the function values are get from the supper plane directly.

IV. Experimental Results

In this section, some experiments are proposed to validate our system. The main function used to be approximated is $sinc(x) = \frac{sin(x)}{x}$, $x \in [-10, 10]$ which are called in-set data samples and the others x are called out-set data points. The kernel used in TSVM is Exponential RBF, $K(u, v) = \exp(\sqrt{((u-v)*(u-v)')}/(2*\sigma^2))$ with $\sigma = 4$ or $\sigma = 1$.

The first experiment is to see the relationship between the approximating correctness and the number of training data samples. At first, we use about 34 data samples obtained from the function $sinc(x)$ with $x \in [-10, 10]$ to train the TSVM, and we get the supper plane as that are shown in Fig. 4(a). Then, we use about 68 data samples obtained from the function $sinc(x)$ with $x \in [-10, 10]$ to train the TSVM, and we get the supper plane as that are shown in Fig. 4(b). In the validating phase, we use 20 data points in $[-10, 10]$ as in-set testing data, 10 data points in $[-15, -10)$ and 10 data points in $(10, 15]$ as out-set testing data. The testing results are shown in the TABLE I. From the Fig. 4 and TABLE I, we can see that, without the noise data disturbance, the more correct approximation will be obtained with more training data samples.

The second experiment is to see the interference of the noise data samples. In training phase, we use about 68 data samples obtained from the function $sinc(x)$ and about 50 noise data points distributed around the normal data samples. The kernel and its parameters used in TSVM and SVR are same. The Fig. 5(a) shows the results of the TSVM and Fig. 5(b) shows the results of the SVR. From the Fig. 5, we can see the approximation based on TSVM is better than SVR on overfitting problem.

The third experiment is to test the representation of TSVM and SVR on the multi-value function approximation. This kind of function approximations are usually used in controlling fields. Because of the principle of the SVR, multi-value function approximation can't be completed with SVR. But in the practical applications, this kind of function approximation is usually used to track the objective. The result based on TSVM is shown in Fig. 6(a) and the result based on SVR is shown in Fig. 6(b). From the Fig. 6, the multi-value function approximation can be realized using TSVM.

The last experiment focus on the high dimension function approximations based on TSVM. In this test, the $z = sin(\sqrt{x^2+y^2})$ function is be approximated. The kernel of TSVM in this experiment is Exponential RBF, and the parameter σ=1. There are 4489 data samples used to train the TSVM, every which is represented as (x, y, z), $x \in [-10, 10]$, $y \in [-10, 10]$ and $z \in [-10, 10]$. And 1000 in-set data points are used to test the supper plane, 932 data points get right function value. When the 1000 out-set data points are input into the system, about 754 data points get right function value. The Fig. 7 shows the supper plane which is the approximating function based on TSVM. From the results we have get, the function approximation based on TSVM in high dimension case is feasible.

TABLE I
IN-SET AND OUT-SET TESTING FOR FUNCTION APPROXIMATED BASED ON TSVM WITH DIFFERENT TRAINING DATA SAMPLES

accuracy (points)	34 points training	68 points raining
$In-set\ testing$ $20\ points\ in\ [-10, 10]$	17	20
$out-set\ testing$ $10\ points\ in\ [-15, -10)$	6	8
$out-set\ testing$ $10\ points\ in\ (10, 15]$	6	8

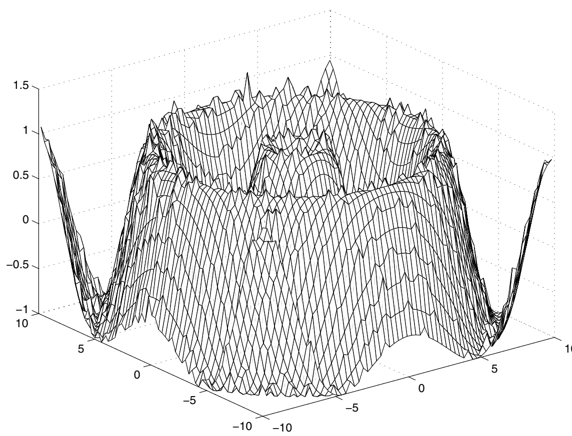


Fig. 7. The function approximation of $z = sin(\sqrt{x^2+y^2})$ based on TSVM is shown. Here the kernel is Exponential RBF with $\sigma = 1$.

V. Conclusion

In this paper, a new function approximation based on TSVM was proposed to solve the overfitting problem in SVR. The basic idea of the approach is to utlize the property of TSVM of generating two nonparallel planes such that each plane is closer to one of the two classes and is as far as possible from the other. In our contribution, only the intuitionistic results were shown. Some theory analysis will be given in our future work.

Acknowledgment

This work was supported by Chinese 863 High-Tech Program Under Grant 2007AA01Z321.

References

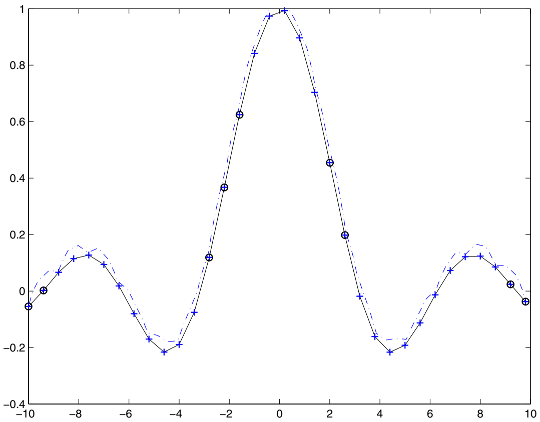
[1] D.S. Chen and R.C. Jain, *A Robust Back-propagation Learning Algorithm for Function Approximation*, IEEE Transactions on Neural Networks, Vol. 5, pp. 467-479, May 1994.

[2] C.J.C. Burges, *A Tutorial on Support Vector Machines for Pattern Recognition*, Data Mining Knowledge Discover, Vol. 2, No. 2, 1996.

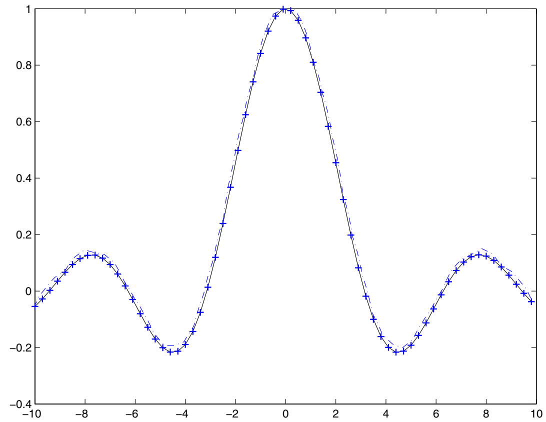
[3] A.J. Smola and B. Schälkopf, *A Tutorial on Support Vector Regression*, Royal Holloway College, London, U.K., Neuro COLT Tech. Rep. TR-1998-030, 1998.

[4] V.N. Vapnik, *The Nature of Statistical Learning Theory*, second ed. New York: Springer, 2000.

[5] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle, *Least Squares Support Vector Machines*, second ed. Singapore: World Scientific Publishing Co., 2002.

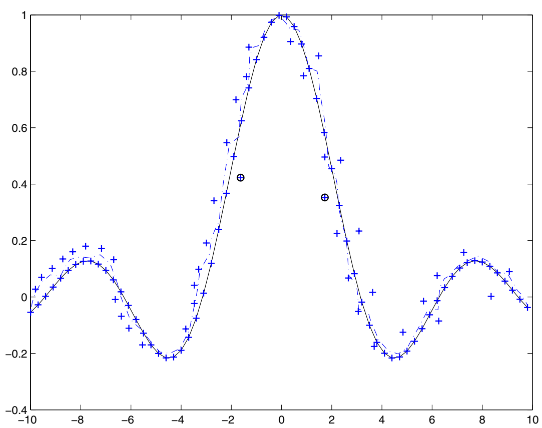


(a) TSVM for sinc function approximation in less samples.

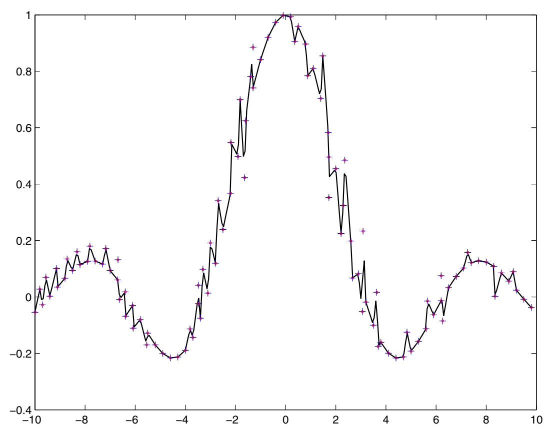


(b) TSVM for sinc function approximation in more samples.

Fig. 4. TSVM with Exponential RBF kernel for $sinc(x) = \frac{sin(x)}{x}$ function approximation. There are less samples for training in left figure, and more samples in right figure. Here the $\sigma = 4$ in Exponential RBF kernel. The black line is the origin sinc function, and the blue dot line is the sinc approximation with TSVM.

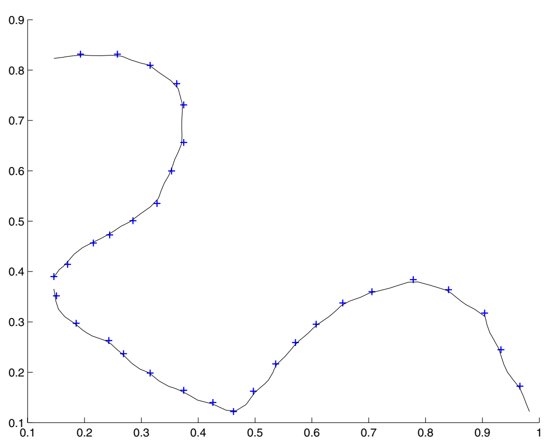


(a) TSVM for sinc(x) function approximation with noise samples disturbances.

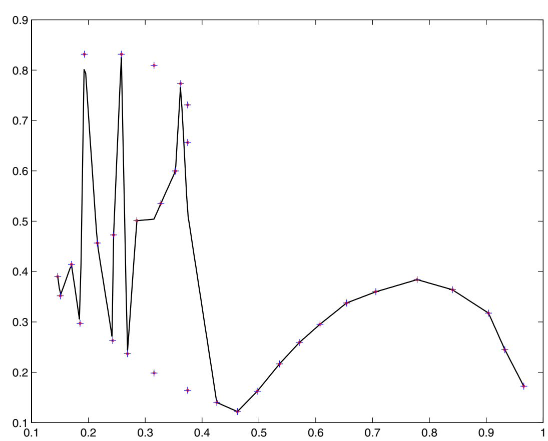


(b) SVR for sinc(x) function approximation with noise samples disturbances.

Fig. 5. Comparison between TSVM and SVR in $sinc(x) = \frac{sin(x)}{x}$ function approximation with noise data samples disturbances. The left figure shows the results using TSVM and the SVR results is shown in right figure. Here the kernel is Exponential RBF with $\sigma = 4$.



(a) TSVM for multi-value function approximation.



(b) SVR for multi-value function approximation.

Fig. 6. Comparison between TSVM and SVR in multi-value function approximation. The left figure shows the results using TSVM and the SVR results is shown in right figure. Here the kernel is Exponential RBF with $\sigma = 4$.

[6] C.C. Chuang, S.F. Su, J.T. Jeng and C.C. Hsiao, *Robust Support Vector Regression Networks for Funtion Approximation with Outliers*, IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1322-1330, November 2002.

[7] B.J. de Kruif and T.J.A. de Vries, *Support-vector-based Least Squares for Learning Non-linear Dynamics*, in Proceedings of the CDC'02, 2002.

[8] J.A. Piñeiro, S.F. Oberman, J.M. Muller and J.D. Bruguera,*High-Speed Function Approximation Using a Minimax Quadratic Interpolator*, IEEE Transactions on Computers, Vol. 54, No. 3, pp. 304-318, March 2005.

[9] S. Ferrari and R.F. Stengel, *Smooth Function Approximation Using Neural Networks*, IEEE Transactions on Neural Networks, Vol. 16, No. 1, pp. 24-38, January 2005.

[10] O.L. Mangasarian and E.W. Wild, *Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 1, pp. 69-74, January 2006.

[11] J.Y. Goulermas, P. Liatsis, X.J. Zeng and P. Cook, *Density-Driven Generalized Regression Neural Networks (DD-GRNN) for Function Approximation*, IEEE Transactions on Neural Networks, Vol. 18, No. 6, pp. 1683-1696, November 2007.

[12] Jayadeva, B. Khemchandani and Suresh Chandra, *Twin Support Vector Machines for Pattern Classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 5, pp. 905-910, May 2007.