4th WSEAS Int. Conf. on COMPUTATIONAL INTELLIGENCE, MAN-MACHINE SYSTEMS and CYBERNETICS Miami, Florida, USA, November 17-19, 2005 (pp70-75)

Robust Kernel-Based Regression

Budi Santosa
Department of Industrial Engineering
Sepuluh Nopember Institute of Technology
Kampus ITS Surabaya
Surabaya 60111,Indonesia

Theodore B. Trafalis
School of Industrial Engineering
University of Oklahoma
202 West Boyd, Room 124,
Norman, OK 73019, USA

ABSTRACT
In this research, a robust optimization approach applied to support vector regression (SVR) is investigated. A novel kernel based-method is developed to address the problem of data uncertainty where each data point is inside a sphere. The model is called robust SVR. Computational results show that the resulting robust SVR model is better than traditional SVR in terms of robustness and generalization error.

KEY WORDS
Kernel Method, Robust Optimization, Regression, Support Vector Machine, Uncertainty

1 Introduction

Currently, incorporating uncertainty into a mathematical model formulation is an issue of active research in the machine learning community. Lanckriet et al.[10] developed a robust minimax probability machine (MPM) to predict the class of new observations in binary class problems. In their work, the mean and covariance matrix of the data in each class are assumed to belong in some specified set. In [7] the model that incorporates the uncertainty of the data is explored in a different way. The uncertainty of the data is characterized by interval uncertainties of the data within given hyper-rectangles. Following the idea of the minimax probability machine for classification due to Lanckriet et al.[10], in [13] robust minimax probability machine regression is formulated. In their paper regression is formulated as maximizing the minimum probability Ω that the true regression function is within $\pm\epsilon$ of the regression model. Trafalis and Alwazzi [14] proposed a robust support vector machine (SVM) classifier that studies noisy data with bounded errors on the linear model of SVM. Their work investigated how the stability of the solution is affected by the noise of the data. In this research, a robust SVM approach is proposed which can improve the generalization error. The motivation is to increase the margin of separation by introducing noise. Different from the previous work, this research emphasizes how the generalization error improves with the data perturbation. In Trafalis and Alwazzi's [14] approach, the margin of separation decreases with the increase of the noise level and it approaches zero as the radius of the uncertainty sphere becomes equal to the margin. In our case, the margin is increasing as the level of uncertainty is increasing. Street and Mangasarian [12] proved that the generalization error is improved when the training set is learned with less accuracy. They developed a linear model and train with several degrees of tolerances τ to investigate the influence of the noise to the test generalization. Our paper is organized as follows. In section 2, a literature review on robust optimization, robust classification methods and SVR is provided. Section 3 provides a mathematical formulation of the proposed model. In Section 4 computational results are provided and section 5 concludes the paper.

2 Literature Review

2.1 The Kernel Method

Several machine learning algorithms such as the perceptron are developed with the assumption of linearity. Then, the resulting algorithms are limited to linear discriminant functions. Hence, if for example a certain classification problem displays a nonlinear separating surface, algorithms such as the perceptron will not be able to account for this nonlinear behavior. In general, complex real-world problems require more expressive hypothesis spaces than linear functions. Kernel methods [11] offer an alternative solution by mapping a data point x in the input space into a higher dimensional feature space F through a feature map φ such that $\varphi : x \mapsto \varphi(x)$. Therefore the point x in the input space becomes $\varphi(x)$ in the feature space.

Unfortunately, very often the function $\varphi(x)$ is not available, can not be computed, or does not even exist. However, the dot product of two vectors can be computed, both in the input and feature space. In other words, while $\varphi(x)$ might not be available, the dot product $< \varphi(x_1), \varphi(x_2) >$ can still be computed in the feature space. In order to employ the kernel method, it is necessary to express the separation constraints in terms of inner products of the data vectors x_i. Consequently, the constraints describing the classification problem have to be reformulated, such that solely dot products are used. In the new space the dot product $< . >$ becomes $< \varphi(x), \varphi(x)' >$. A nonlinear kernel function, $k(x, x')$, can be used to substitute the dot product $< \varphi(x), \varphi(x)' >$. Then in the higher dimensional feature space, we can construct a linear decision function that represents a nonlinear function in the input

space. The following nonlinear kernel functions are usually used in the SVM literature [9]:

1. polynomial: $(x^T x_i + 1)^p$,
2. radial basis function (RBF): $exp(-\frac{1}{2\sigma^2}\|x - x_i\|^2)$,

The best kernel function which one can use to substitute for the dot products in the feature space depends on the data; usually one has to use cross-validation methods [8] to select the best kernel function.

2.2 Support Vector Machines

A well known method in machine learning to find an optimal classifier (hyperplane) between two sets of points is the so called SVM [15]. This method has attracted people in the machine learning and optimization community because of its impressive performance in generalization error of unseen data. In this method one seeks the best hyperplane among many possible hyperplanes to separate two sets of patterns. The optimal hyperplane is the one that is located mid-way between the two classes. This hyperplane is orthogonal to the shortest line connecting the convex hulls of the two classes. Seeking the best hyperplane is equivalent to maximizing the margin between the two classes. If $wx_1 + b = +1$ is on the supporting hyperplane of class $+1$ ($wx_1 + b = +1$) and $wx_2 + b = -1$ is on the supporting hyperplane of class -1 ($wx_2 + b = -1$), the margin between the two classes can be computed by computing the distance between the supporting hyperplanes of those classes. Specifically, the margin is computed as follows $(wx_1 + b = +1) - (wx_2 + b = -1) \Rightarrow w(x_1 - x_2)) = 2 \Rightarrow \left(\frac{w}{\|w\|}(x_1 - x_2)\right) = \frac{2}{\|w\|}$. The mathematical formulation of the SVM optimization problem for the linear separable case is given as

$$\min \frac{1}{2}\|w\|^2 \quad \text{Subject to} \quad y_i(wx_i + b) \geq 1,\ i = 1, .., \ell. \tag{1}$$

In the case of linear non-separable problems, the formulation of the SVM optimization problem is given as

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell} t_i \quad \text{Subject to} \quad y_i(wx_i + b) + t_i \geq 1,\ t_i \geq 0,\ i = 1, .., \ell. \tag{2}$$

By this formulation one wants to maximize the margin of separation of two classes by minimizing $\|w\|^2$ [9]. One needs to minimize the misclassification errors that are described by the slack variables t_i while maximizing the margin. The slack variable t_i is used to handle the case of infeasibility of hard constraints $y_i(wx_i + b) \geq 1$ by penalizing points that do not satisfy the hard constraints. To minimize such deviations, we penalize those through a regularization constant C. The vector w is the normal to the separating hyperplane: $wx + b = 0$. The constant b determines its location relative to the origin. To address the problem of nonlinearity that frequently occurs in real world problems, one can utilize kernel methods. The dual formulation of problem (2) is expressed in the feature space:

$$\min\ \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{\ell} \alpha_i \quad \text{Subject to} \quad 0 \leq \alpha_i \leq C, i = 1, ..\ell \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0, \tag{3}$$

where k is the kernel function described in section 2.1. The formulation in (3) is a linearly constrained quadratic programming. Training SVM is equivalent to solving the above convex optimization problem. Therefore the solution of SVM is unique (under the assumption that k is positive definite) and globally optimal, unlike other networks' training [9] which is equivalent to a nonconvex optimization problem with the danger of obtaining local optima solutions. Let $f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^* k(x_i, x) + b^*$. The resulting optimal classifier is $g(x) = sign(\sum_{i=1}^{\ell} y_i \alpha_i^* k(x, x_i)) + b^*$, where $\alpha_i^*, i = 1, .., \ell$ are the optimal solutions of problem (3) and b^* is chosen so that $y_i f(x_i) = 1$ for any i with $C > \alpha_i^* > 0$ [6]. The points x_i for which $\alpha_i^* > 0$ are called *support vectors* and represent the training data points that are needed to represent the optimal decision function.

2.3 Support Vector Regression

By the introduction of Vapnik's ϵ-insensitive loss function, SVM has been generalized for function approximation or regression [11]. Established on the unique theory of Structural Risk Minimization principle to estimate a function by minimizing an upper bound of the generalization error, SVM is shown to be very resistant to the over-fitting problem, eventually achieving high generalization performance. Suppose we have been given ℓ training data, $(x_i, y_i),\ i = 1, .., \ell$ with input data $x = \{x_1, x_2.., x_\ell\} \subseteq \Re^N$ and corresponding outputs $y = \{y_1, .., y_\ell\} \subseteq \Re$. By support vector regression, one wants to find a function $f(x)$ that has at most ϵ deviation from the actual target y_i for all training data. Suppose we have the following function as a regressor:

$$f(x) = w^T \varphi(x) + b, \tag{4}$$

where $\varphi(x)$ denotes a point in the high dimensional feature space F which is the mapping of a point x in the input space. The coefficients w and b are estimated by minimizing the reg-

ular risk function defined in equation (5).

$$\min \frac{1}{2}\|w\|^2 + C\frac{1}{\ell}\sum_{i=1}^{\ell} L_\epsilon(y_i, f(x_i)) \quad (5)$$

Subject to

$$y_i - w\varphi(x_i) - b \leq \epsilon$$
$$w\varphi(x_i) - y_i + b \leq \epsilon, i = 1, .., \ell,$$

where

$$L_\epsilon(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \epsilon & |y_i - f(x_i)| \geq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The term $\|w\|^2$ is called the regularization term. Minimizing $\|w\|^2$ will make a function as flat as possible, thus playing the role of controlling the function capacity. The second term is the empirical error measured by the ϵ-insensitive loss function. Using the idea of ϵ-insensitive loss function [15], one should seek to minimize the norm of w in order to accomplish good generalization properties for the regressor f. Therefore, we have to solve the following optimization problem in the primal weight space:

$$\min \frac{1}{2}\|w\|^2 \quad (7)$$

Subject to

$$y_i - w\varphi(x_i) - b \leq \epsilon$$
$$w\varphi(x_i) - y_i + b \leq \epsilon, i = 1, .., \ell$$

We assume that there is a function f that approximates all pairs (x_i, y_i) with precision ϵ. In this case, we assume that the problem is feasible. In the case of infeasibility, where some points might deviate from $f \pm \epsilon$, one can introduce slack variables t, t^* to cope with infeasible constraints of the optimization problem. Then, the above problem can be formalized as [15]:

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(t_i + t_i^*) \quad (8)$$

Subject to

$$y_i - w^T\varphi(x_i) - b - t_i \leq \epsilon, i = 1, .., \ell$$
$$w\varphi(x_i) - y_i + b - t^*{}_i \leq \epsilon, i = 1, .., \ell$$
$$t_i, t^*{}_i \geq 0,$$

The constant $C > 0$ determines the trade off between the flatness of function f and the amount up to which deviations larger than ϵ are tolerated. Any deviation more than ϵ will be penalized with C. Figure 1 depicts the situation graphically. Only the points outside the shaded region contribute to the cost insofar, as the deviations are penalized in a linear fashion. In Support vector Regression (SVR), ϵ is equivalent to the approximation accuracy placed on the training data points. A small ϵ corresponds to a large slack variable $t_i^{(*)}$ and high approximation accuracy. On the contrary, a large ϵ corresponds to a small slack variable $t_i^{(*)}$ and low approximation accuracy. According to equation (8), a large slack variable will make the empirical error having a large impact relatively to the regularized term. In SVR, support vectors are the training data points lying on or outside the ϵ-bound of the decision function. Therefore, the number of support vectors decreases as ϵ increases. Finally, by introducing Lagrange multipliers and exploiting the optimality constraints, the decision function is explicitly given as:

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)K(x_i, x) + b, \quad (9)$$

where $K(x_i, x)$ is defined through the kernel function k.

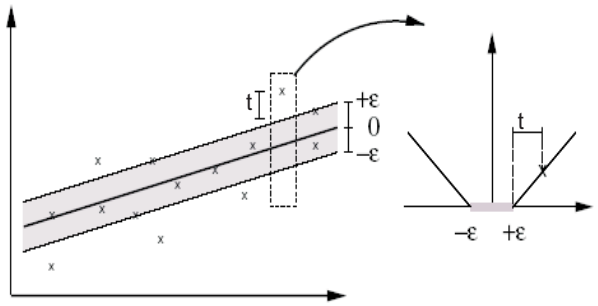


Figure 1. ϵ-insensitive loss function.

The points outside the shaded region are penalized

2.4 Robust Optimization

The robust optimization methodology is a relatively new approach to deal with uncertain data. More recently, the so called robust optimization techniques have been investigated by several authors [2, 1, 4]. Those techniques are more meaningful in formulations with prior bounds on the size of the uncertainties on the data. Specifically, we consider the case where we have data with bounded errors. The solutions coming from robust optimization models are more stable and more appropriate for this kind of uncertainty.

Ben-Tal and Nemirovski [2] proposed the foundation of robust convex optimization based on previous work in robust control. Their assumption is that the data defining a convex optimization problem are not accurately specified, and the only knowledge about those is that they belong to a bounded uncertainty set U. They have shown, that when this set U is an ellipsoidal uncertainty set, then the robust convex program corresponding to some of the most important generic convex problems, such as linear programming, semi-definite programming and others, is a convex optimization problem which can be solved by an efficient algorithm, such as polynomial time interior point methods.

2.5 Robust SVM Classifier

In [14], a robust SVM classifier development is described. This research assumes noisy data with bounded errors on the linear programming (LP) SVM formulation. Specifically, this approach assumes that a data point can be represented through a sphere with a known radius. Accordingly, the supporting hyperplane resulting from the model will be on the boundary of the sphere that contains the data closest to the separating hyperplane (classifier) in one side and on the boundary of the sphere from the separating hyperplane in the other side. In other words, the training data points (represented through the centers of the corresponding uncertainty spheres) can be modified through a new set of data points that are obtained by shifting the points labeled as $+1$ along $-w$ and the points labeled -1 along w, respectively to its boundary of uncertainty (see Figure 2). The optimization problem

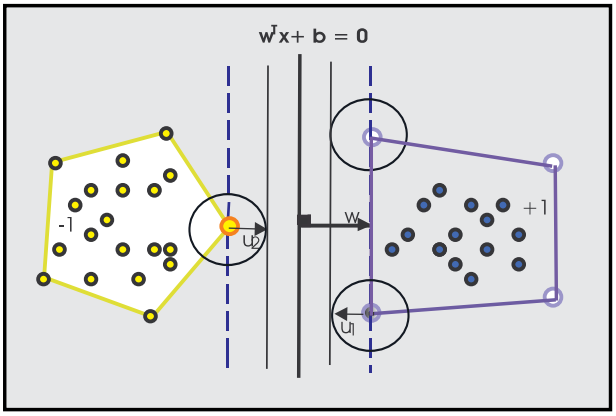


Figure 2. Geometric illustration of robust SVM

using the approach in [14]

formulation is given as

$$\min_{w,b,t} \|w\|_1 + C\sum_{i=1}^{\ell} t_i \qquad (10)$$

Subject to

$$y_i \langle w, \tilde{x_i}\rangle - \sqrt{\eta}\,\|w\| + y_i b + t_i \geq 1$$
$$t_i \geq 0,\ i = 1..\ell,$$

where $\sqrt{\eta}$ is the radius and $\tilde{x_i}$ is the center of the uncertainty sphere. Setting $w = \sum_{i=1}^{\ell} y_i \alpha_i x_i$ and linearizing the objective function, the above problem can be formulated as follows:

$$\min \sum_{i=1}^{\ell} \alpha_i + C\sum_{i=1}^{\ell} t_i \quad (11)$$

Subject to

$$\sqrt{\eta}\sqrt{\alpha^t \tilde{k} \alpha} - y_i \sum_{i=1}^{\ell} y_j \alpha_j k(\tilde{x_j}, \tilde{x_i}) - y_i b - t_i + 1 \leq 0$$
$$t_i \geq 0, \alpha_i \geq 0,\ i = 1, .., \ell,$$

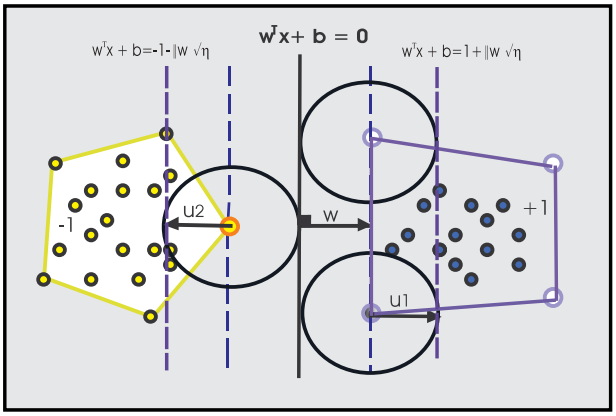


Figure 3. Finding the best classifier for data with uncertainty.

The bounding planes are moved to the boundary of the spheres to obtain maximum margin

where $\tilde{k} = \tilde{k}(x_i, x_j) = y_i y_j < x_i, x_j >$. It is shown that the resulting SVM classifier is *robust* to the noise of the data [14].

3 Robust SVM Formulation

In this section, we develop our model. We begin with the robust support vector machine (R-SVM) for binary classification problems. The next step is extending our models for function approximation or regression problems.

Suppose that we have a set of ℓ samples $\{x_1, x_2, .., x_\ell\}$ and we want a weight vector w and a bias b that satisfies $y_i(wx_i + b) \geq 1$ for all $i = 1, .., \ell$. Recall SVM formulation in 2. Now consider that our data are perturbed. Instead of having the input data point x_i we have $x_i = \tilde{x_i} + u_i$ where u_i is a bounded perturbation with $||u_i|| \leq \sqrt{\eta}$ where η is a positive number, and $\tilde{x_i}$ is the center of the uncertainty sphere where our data point is located. Therefore, the constraints in (2) become

$$\min \frac{1}{2}\alpha^T K \alpha + C\sum_{i=1}^{\ell} t_i \quad (12)$$

$$y_i(< w, x_i > + b) + t \geq 1 \Leftrightarrow \quad (13)$$
$$y_i(< w, \tilde{x_i} > + < w, u_i > + b) + t_i \geq 1, i = 1, .., \ell$$
$$t_i \geq 0$$

Our concern is the problem of classification with respect to two classes. In order to have the best separating hyperplane we try to minimize the dot product of w and u_i in one side of the separating hyperplane (class -1) and maximize the dot product of w and u_i in the other side (class 1) subject to $||u_i|| \leq \sqrt{\eta}$. By this logic we are trying to maximize the distance between the classifier to both points on different sides (see Figure) 3. Therefore, we have to solve the following problem

$$\max < w, u_i > \qquad (14)$$

Subject to $||u_i|| \leq \sqrt{\eta}$

Using Cauchy's Schwarz inequality ($| < w, u > | \leq \|w\| . \|u\| \Rightarrow -\|w\| . \|u\| \leq < w, u > \leq \|w\| . \|u\|$), the maximum of $< w, u_i >$ is equal to $\|w\| . \|u\|$. Hence, referring to (14) the maximum of the dot product of $< w, u_i >$ will be $\sqrt{\eta}\|w\|$. By substituting this maximum value in (12), we have

$$\min \frac{1}{2}\alpha^T K\alpha + C\sum_{i=1}^{\ell} t_i \qquad (15)$$

Subject to

$$\sqrt{\eta}\|w\| - w\tilde{x}_i - b + t_i \geq 1, for\ y_i = -1$$
$$\sqrt{\eta}\|w\| + w\tilde{x}_i + b + t_i \geq 1, for\ y_i = +1$$
$$t_i \geq 0, i = 1, .., \ell$$

If we map the data from the input space to the feature space F,we can represent w in the space F as

$$w = \sum_{i=1}^{\ell} \alpha_i \varphi(\tilde{x}_i) \qquad (16)$$

By substituting w with the above representation and substituting $< \varphi(\tilde{x}), \varphi(\tilde{x})' >$ with $k(\tilde{x}, \tilde{x}')$, we have the following R-SVM formulation:

$$\min \frac{1}{2}\alpha^T K\alpha + C\sum_{i=1}^{\ell} t_i \qquad (17)$$

Subject to

$$\sqrt{\eta}\sqrt{\alpha^T K\alpha} - K_i\alpha - b + t_i \geq 1, for\ y_i = -1$$
$$\sqrt{\eta}\sqrt{\alpha^T K\alpha} + K_i\alpha + b + t_i \geq 1, for\ y_i = +1$$
$$t_i \geq 0$$

where K_i is the 1xℓ vector corresponding to the ith line of the kernel matrix K. Note that we reorder the rows of the matrix K based on the label. It is important to note that most of the time we do not need to know explicitly the map φ. The important idea is that we can replace $< \varphi(x), \varphi(x)' >$ with any suitable kernel $k(x, x')$.

By the margin(η), we define the margin of separation when the level of uncertainty is η. Then

$$margin(\eta) = \frac{(1+\|w\|\sqrt{\eta}-b)-(-1-b+\sqrt{(}\eta)\|w\|)}{\|w\|} \qquad (18)$$
$$= \frac{2+2\sqrt{\eta}\|w\|}{\|w\|}$$
$$= \frac{2}{\|w\|} + 2\sqrt{\eta} = margin(0) + 2\sqrt{\eta}.$$

Note that the margin of separation is increasing. In the case of robust optimization formulation [14], margin(η) = margin(0)-$2\sqrt{\eta}$. Now, consider the support vector regression problem in equation (8). Using perturbed data as explained above, we define the robust support vector regression (R-SVR) problem as follows

$$\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(t_i + t^*{}_i) \qquad (19)$$

Subject to

$$y_i - < w, \tilde{x_i} > - < w, u_i > -b - t_i \leq \epsilon, i = 1, .., m$$
$$< w, \tilde{x_i} > + < w, u_i > -y_i + b - t^*{}_i \leq \epsilon, i = 1, .., m$$
$$t_i, t^*{}_i \geq 0$$
$$\forall u_i \in \Re^d, \text{such that} ||u_i|| \leq \sqrt{\eta}.$$

By substituting w with $K_i\alpha$, and the term $< w, u >$ with its maximum value, $\sqrt{\eta}\sqrt{\alpha^T K\alpha}$, we obtain a robust support vector regression (R-SVR) formulation as:

$$\min \frac{1}{2}\alpha^T K\alpha + C\sum_{i=1}^{\ell}(t_i + t^*{}_i) \qquad (20)$$

Subject to

$$y_i - K_i\alpha - \sqrt{\eta}\sqrt{\alpha^T K\alpha} - b - t_i \leq \epsilon, i = 1, .., m$$
$$K_i\alpha + \sqrt{\eta}\sqrt{\alpha^T K\alpha} - y_i + b - t^*{}_i \leq \epsilon, i = 1, .., m$$
$$t_i, t^*{}_i \geq 0$$
$$\forall u_i \in \Re^d, \text{such that } ||u_i|| \leq \sqrt{\eta}.$$

The geometric illustration of robust SVR is given in Fig.4.

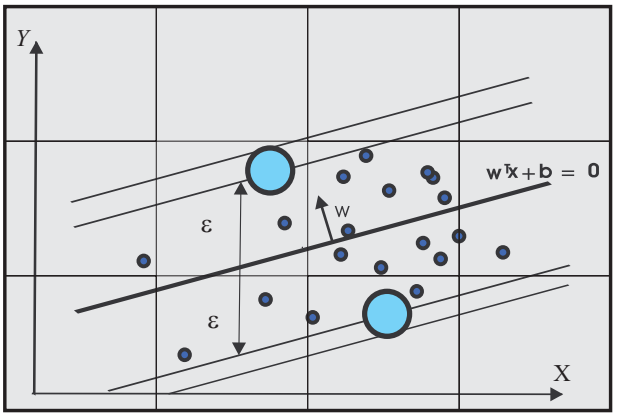


Figure 4. Geometric illustration of robust SVR

4 Computational Results

In this implementation R-SVR is applied in time series Flour price data [5] and Abalone data [3]. Table 1 and Table 3 show the performance of R-SVR with uncertainty on Flour price data and Abalone data. As shown in those tables, applying uncertainty can reduce the MSE of SVR significantly. Table 2 indicates that varying ϵ can improve the performance of SVR.

5 Conclusions

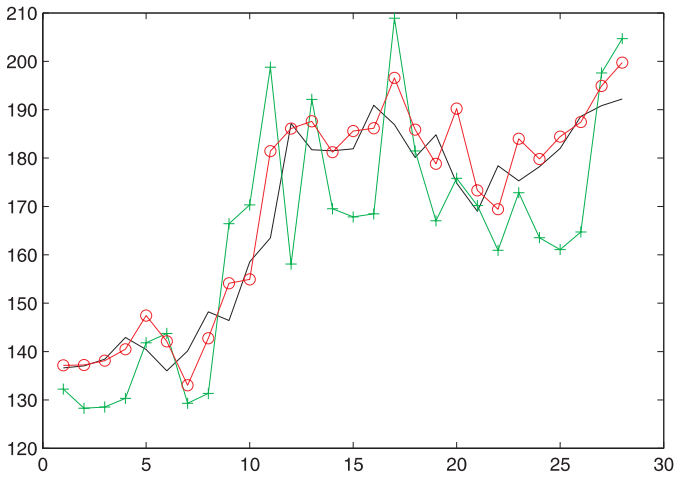
In this work, motivated by the presence of uncertainty in real data,a novel robust support vector regression approach has been developed. The impact of uncertainty on the data to the generalization error was investigated for regression problems. Robust SVR (R-SVR) improved the performance of regular SVR.

Table 1. MSE of R-SVR on flour price data with RBF kernel with η varied, ϵ = 0.0

η	C=1000, $\sigma = 10$
0.0	265.79
0.00001	164.32
0.0001	121.89
0.001	46.33
0.01	381.35

Table 2. MSE of R-SVR on flour price, RBF, C=1000,σ=10,η=0.001

ϵ	MSE
0	46.33
1	45.77
3	41.04
5	47.90
10	98.90



— :actual data, -+- : η = 0, -○- : η = 0.001

Figure 5. Plots of the actual data and R-SVR results with RBF kernel on flour price data

References

[1] A Ben-Tal and A. Nemirovski. Robust solutions to uncertain linear program via convex programming. *Operation Research Letters*, 25(1):1–17, 1996.

[2] A Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operation Research*, 23(4):769–805, 1998.

[3] C.L. Blake and C.J. Merz. UCI Repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[4] S. Boyd, M.S Lobo, and L. Vandenberghe. Application of second-order cone programming. *Linear Algebra and its Applications*, 284:193–226, 1998.

[5] K. Chakraborty, K. Mehrotra, C.K. Mohan, and S. Ranka. Forecasting the behavior of multivariate time series using neural network. *Neural Networks*, 5:961–970, 1992.

[6] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

Table 3. MSE and computation time (CPU time) with different η values on Abalone, training=750 points, testing=250 points, ϵ = 0.0

η	MSE(CPU-time)
0.0	7.76(193.15)
0.001	29.54(32.03)
0.01	21.52(28.16)
0.1	7.56(30.56)
1	32.83(44.14)

[7] L. El Ghaoui, G.R.G. Lanckriet, and G. Natsoulis. Robust classification with interval data. Technical report, Technical Report CSD-03-1279, Division of Computer Science, University of California, Berkeley, 2003. http://robotics.eecs.berkeley.edu/ gert/.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: data mining , inference, and prediction*. Springer-Verlag, New York, 2001.

[9] Simon Haykin. *Neural Network: A Comprehensive Foundation*. Prentice Hall, New Jersey, 1999.

[10] G.R.G. Lanckriet, L. El Ghaoui, C. Bhattacharyya, and M.I. Jordan. A robust minimax approach to classification. *Journal of Machine Learning Research*, 3:555–582, 2002.

[11] B. Schölkopf and A.J. Smola. *Learning with Kernels*. The MIT Press, Cambridge, Massachusetts, 2002.

[12] W.N. Street and O.L. Mangasarian. Improved generalization error via tolerant training. *Journal of Optimization Theory and Applications*, 96(2):259–279, 1998.

[13] Thomas R. Strohmann and Gregory Z. Grudic. Robust minimax probability machine regression. Technical report, Department of Computer Science,University of Colorado, 2003. http://www.cs.colorado.edu/ grudic/publications/.

[14] T.B. Trafalis and S.A. Alwazzi. Robust optimization in support vector machine training with bounded errors. In *Proceedings of the International Joint Conference On Neural Networks, Portland, Oregon*, pages 2039–2042. IEEE Press, 2003.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.