SUPPORT VECTOR MACHINE FOR REGRESSION AND APPLICATIONS TO FINANCIAL FORECASTING

Theodore B. Trafalis* **and Huseyin Ince****
School of Industrial Engineering, University of Oklahoma
220 W. Boyd, Suite 124, Norman, Oklahoma 73019
*trafalis@ecn.ou.edu; **ince@ou.edu

Abstract
The main purpose of this paper is to compare the support vector machine (SVM) developed by Vapnik with other techniques such as Backpropagation and Radial Basis Function (RBF) Networks for financial forecasting applications. The theory of the SVM algorithm is based on statistical learning theory. Training of SVMs leads to a quadratic programming (QP) problem. Preliminary computational results for stock price prediction are also presented.

1. Introduction

The objective of this paper is to use support vector machines for regression applied to forecast stock prices. Recently, the support vector machine technique has attracted many researchers in optimization and machine learning areas.

SVM algorithm developed by Vapnik [4] is based on statistical learning theory. In classification case [3,4,7,8], we try to find an optimal hyperplane that separates two classes. In order to find an optimal hyperplane, we need to minimize the norm of the vector w, which defines the separating hyperplane. This is equivalent to maximizing the margin between two classes. In the case of regression [1,5,6,9,10], the goal is to construct a hyperplane that lies "close" to as many of the data points as possible. Therefore, the objective is to choose a hyperplane with small norm while simultaneously minimizing the sum of the distances from the data points to the hyperplane. Both in classification and regression, we obtain a quadratic programming problem where the number of variables is equal to the number of observations.

2. Statistical Learning Theory

The task of learning from the data in the case of two-class pattern classification, can be formulated in the following way:

Given a set of decision functions

$$\{f_\lambda(x): \lambda \in \Lambda\}, \quad f_\lambda : \Re^n \to \{-1,1\} \qquad (1)$$

where Λ is set of abstract parameters.

Suppose $(x_1,y_1), (x_2,y_2), \ldots, (x_l, y_l)$ such that $x \in \Re^n$, $y \in \{-1,1\}$ are taken from an unknown distribution P(x,y). We want to find f_λ^* which minimizes the expected risk, $R(\lambda)$ which is defined as

$$R(\lambda) = \int \frac{1}{2} |f_\lambda(x) - y| dP(x,y)$$

where $f_\lambda(x)$, $\{f_\lambda(x): \lambda \in \Lambda\}$ are called hypothesis and hypothesis space, respectively.

The problem is that the distribution function P(x,y) is unknown. Therefore, we could not compute the expected risk. On the other hand, we have examples from the distribution P(x,y). Therefore, we can compute a stochastic approximation of $R(\lambda)$ which is called empirical risk. Specifically,

$$R_{emp}(\lambda) = \frac{1}{2l} \sum_{i=1}^{l} |f_\lambda(x_i) - y_i|.$$

0-7695-0619-4/00 $10.00 © 2000 IEEE

$R_{emp}(\lambda)$ is a fixed number given a particular choice of λ and a particular training set $\{x_i,y_i\}$. The quantity $(1/2)|f_\lambda(x) - y|$ is called loss. It can only take the values 0 and 1. The theory of uniform convergence in probability provides bound on the deviation of the empirical risk from the expected risk. A typical uniform VC bound, which holds with probability 1-η, has the following form [4].

$$R(\lambda) \le R_{emp}(\lambda) + \sqrt{\frac{h(\ln\frac{2l}{h}+1)-\ln(\frac{\eta}{4})}{l}}, \qquad (2)$$

where h is the VC dimension of f_λ, and l is the number of examples.

Structural risk minimization (SRM) developed by Vapnik tries to solve the problem of choosing an appropriate VC dimension. One can see that a small value of the empirical risk does not necessarily imply a small value of expected risk.

The principle is based on the observation that, in order to make the expected risk small, both sides of equation (2) should be small. Therefore, VC dimension and empirical risk should be minimized at the same time.

In order to do this, we need a nested structure of hypothesis space $H_1 \subset H_2 \subset H_3 \subset ... \subset H_n \subset$....with the property that $h(n) \le h(n+1)$, where $h(n)$ is the VC dimension of the H_n. So, we need to solve the following

$$\min_{H_n} \left(R_{emp}(\lambda) + \sqrt{\frac{h(n)}{l}} \right). \qquad (3)$$

problem:

SRM principle is well founded mathematically. According to [8], it can be difficult to implement this principle since the VC dimension of H_n may be difficult to compute and even if one computes the VC dimension, it is not easy to solve the above problem. By using the SVM approach, we can handle this problem. SVM algorithm minimizes the bound on the VC dimension and the number of training errors at the same time.

3. Mathematical Formulation of Support Vector Regression

At first, we consider the linearly separable case, then soft margin support vector regression will be explained and finally, the nonlinear case will be discussed. Specifically, the ε-insensitive support vector regression will be used for predicting stock prices. In the ε- insensitive support vector regression, our goal is to find a function $f(x)$ that has an ε deviation from the actually obtained target y_i for all training data and at the same time is as flat as possible. Suppose $f(x)$ takes the following form:

$$f(x) = wx + b \qquad w \in X, b \in \Re. \qquad (4)$$

Then, we have to solve the following problem:

$$\begin{aligned} &\min \quad \frac{1}{2}\|w\|^2 \\ &\text{Subject to} \qquad (5) \\ &\qquad y_i - wx_i - b \le \varepsilon \\ &\qquad wx_i + b - y_i \le \varepsilon \end{aligned}$$

In the case where the constraints are infeasible, we introduce slack variables ξ_i,ξ^*_i. This case is called soft margin formulation, and is described by the following problem.

$$
\begin{aligned}
&min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\
&Subject\ to \\
&\quad y_i - wx_i - b \le \varepsilon + \xi_i \\
&\quad wx_i + b - y_i \le \varepsilon + \xi_i^* \\
&\quad \xi_i, \xi_i^* \ge 0 \\
&\quad C \succ 0
\end{aligned}
\tag{6}
$$

where, C determines the trade-off between the flatness of the $f(x)$ and the amount up to which deviations larger than ε are tolerated. This is called ε-insensitive loss function $|\xi|_\varepsilon$ and is described by

$$
|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{if } |\xi| \succ \varepsilon \end{cases} \tag{7}
$$

By constructing the Lagrangian function, we formulate the dual problem. Specifically,

$$
\begin{aligned}
L = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}\lambda_i(\varepsilon_i + \xi_i - y_i + wx_i + b) \\
- \sum_{i=1}^{l}\lambda_i^*(y_i + \varepsilon_i + \xi_i^* - wx_i - b) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*),
\end{aligned}
\tag{8}
$$

and $\lambda_i, \lambda_i^*, \eta_i, \eta_i^* \ge 0$

At the optimal solution, we have:

$$
\begin{aligned}
&\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)x_i = 0 \\
&\frac{\partial L}{\partial b} = \sum_{i=1}^{l}(\lambda_i^* - \lambda_i) = 0 \\
&\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \eta_i = 0 \\
&\frac{\partial L}{\partial \xi_i^*} = C - \lambda_i^* - \eta_i^* = 0
\end{aligned}
\tag{9}
$$

We obtain the dual problem by substituting (9) into (8). Specifically, the dual problem is as follows

$$
\begin{aligned}
&max \quad -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\lambda_i - \lambda_i^*)(\lambda_j - \lambda_j^*)x_i x_j - \varepsilon\sum_{i=1}^{l}(\lambda_i + \lambda_i^*) + \sum_{i=1}^{l} y_i(\lambda_i - \lambda_i^*) \\
&\text{Subject to} \quad \sum(\lambda_i - \lambda_i^*) = 0 \\
&\quad \lambda_i, \lambda_i^* \in (0, C).
\end{aligned}
\tag{10}
$$

Solving (9) for $\boldsymbol{w}$, we have

$$
\begin{aligned}
&w^* = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)x_i, \text{and by (4)} \\
&f(x) = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)x_i x + b^*.
\end{aligned}
\tag{11}
$$

We compute the optimal value of b from the complementary slackness conditions. Specifically,

$$\lambda_i(\varepsilon+\xi_i-y_i+w^*x_i+b)=0$$
$$\lambda_i^*(\varepsilon+\xi_i^*+y_i-w^*x_i-b)=0 \qquad 12)$$
$$(C-\lambda_i)\xi_i=0$$
$$(C-\lambda_i^*)\xi_i^*=0.$$

One can make some useful conclusion from the above equations (12). First of all, only samples (x_i,y_i) with corresponding $\lambda_i = C$ lie outside the ε-insensitive tube around f. The set of dual variables can never be nonzero at the same time, i.e.. λ_1, λ^*_1 If λ_i is nonzero, then λ^*_i is zero and vice versa. Finally if λ_i is in $(0,C)$ then the corresponding ξ is zero. Therefore b can be computed as follows.

$$b^* = y_i - w^*x_i - \varepsilon \quad \text{for } \lambda_i \in (0,C)$$
$$b^* = y_i - w^*x_i + \varepsilon \quad \text{for } \lambda_i^* \in (0,C) \qquad (13)$$

Let us look at the non-linear case briefly. First of all, we need to map input space into feature space and try to find a hyperplane in the feature space. Using the trick of kernel functions [4], we have the following QP problem

$$max \qquad -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\lambda_i-\lambda_i^*)(\lambda_j-\lambda_j^*)K(x_i,x_j)-\varepsilon\sum_{i=1}^{l}(\lambda_i+\lambda_i^*)+\sum_{i=1}^{l}y_i(\lambda_i-\lambda_i^*)$$

$$\text{Subject to} \quad \sum(\lambda_i-\lambda_i^*)=0 \qquad (14)$$

$$\lambda_i,\lambda_i^* \in (0,C)$$

At the optimal solution, we obtain

$$w^* = \sum_{i=1}^{l}(\lambda_i-\lambda_i^*)K(x_i), \text{ and}$$
$$f(x)=\sum_{i=1}^{l}(\lambda_i-\lambda_i^*)K(x_i,x)+b, \qquad (15)$$

where K(.,.) is a kernel function.

According to [4], any symmetric positive semi-definite function, which satisfies Mercer's conditions can be used as a kernel function in the SVMs context. Mercer's conditions can be written as,

$$\iint K(x,y)g(x)g(y)dxdy>0, \quad \int g^2(x)dx<\infty, \text{ where} \qquad (16)$$
$$K(x,y)=\sum_{i=1}^{\infty}\alpha_i\psi(x)\psi(y), \qquad \alpha_i \geq 0$$

Polynomial and RBF kernel functions are very common. RBF kernel function is defined as

$$K(x,y)-exp\left(-\frac{(x-y)^2}{2\sigma^2}\right) \qquad (17)$$

Usually we have more than one kernel to map the input space into feature space. The question is which kernel functions provide good generalization for a particular problem. We could not say that one kernel outperforms the others. Therefore, one has to use more than one kernel function for a particular problem. Some validation techniques such as bootstrapping, and cross-validation can be used to determine a good kernel [4]. Even when we decide for a kernel function, we have to decide the parameters of the kernel. For instance, RBF has a parameter σ and one has to decide the value of σ before the experiment. Selection of this parameter is very important. Many algorithms are proposed to solve this problem [2,8,9]. We can divide these algorithms into two groups: (1) using classical nonlinear algorithms such as descent/ascent gradient algorithm, methods of Zoeutendijk [2]; and (2) the

state of art techniques, interior point algorithms especially primal dual path following algorithm [11,12,13]. One can solve moderate sizes of problems with these algorithms. In order to achieve the expected efficiency with large-scale problem, decomposition techniques must be applied [6].

4. Application to Financial Forecasting

Prediction of the economic indicators of a market is very crucial. If one has robust forecasting tools, then he/she will increase the return on investment. The financial forecasting problem is very complicated due to the number of factors that can influence the market. In addition to this, choosing the important factors is also very difficult. Usually, one has to reduce the number of factors in his/her model. This can be done by using some statistical techniques.

IBM, Yahoo and, America Online, daily stock prices data are used as financial forecast application. Data are gathered from the Yahoo's financial web site. Training data are sampled from May 7, 1998 to September 28,1998. Test data are from September 29, 1998 to October 19,1998. In our experimentation, we approximate the following dynamical system.

$$x_t = f(x_{t-1}, x_{t-2}, x_{t-3})$$

where x_t = stock price at time t. A Matlab implementation of the SVR algorithm is used. Two different QP algorithms are used for comparison. Specifically, a primal dual interior point method, and the standard QP algorithm of the optimization toolbox are used. In addition to this, SV regression is compared with other techniques such as backpropagation and RBF networks.

There are several issues that we need to consider in the SVR application. First of all, we need to determine some parameters before running the particular algorithm. These parameters are ε, C and kernel specific parameters. We set to $\varepsilon = 0$, $C = 1000000$. Also, we use different values for the kernel parameter. Specifically, $\sigma = 0.95, 2.5, 5, 7.5, 10, 15$. As p increases from 0.5 to 15, we get better results in terms of the prediction. In addition to this, we have very good forecast results for training period when we use small p. On the other hand, we obtain poor results in testing, period. As we increase p, the accuracy of the prediction also increases. This can be seen from the following tables with $\sigma > 10$.

σ	0.95	2.5	5	7.5	10	15
Standard QP	125.5293	1030.204	14.52261	4.705138	5.95556	2.853779
Primal-Dual	36.43152	24.22798	6.845459	5.306733	4.288409	3.260365

Table 1: MSE comparison for AOL stock price (test data)

σ	0.95	2.5	5	7.5	10	15
Standard QP	3350.755	2386.002	20793.76	2957.18	18882.97	11501.51
Primal-Dual	254.4825	158.0462	68.45128	92.02597	20.81991	86.24175

Table 2: MSE comparison for YAHOO stock price (test data).

σ	0.95	2.5	5	7.5	10	15
Standard QP	932.7354	727.0439	2.772304	13.45547	2.772304	3.864605
Primal-Dual	13.5061	6.060193	10.95697	5.82049	4.538986	3.164936

Table 3: MSE comparison for IBM stock price (test data).

The performance of the other two methods, Backpropagation and RBF networks for the test period is shown in the following table.

	MLP (3 Neurons)	MLP (4 Neurons)	MLP (5 Neurons)	RBF (3 Neurons)	RBF (4 Neurons)	RBF (5 Neurons)
AOL	2.3740	2.3803	2.3512	3.6040	3.3461	3.5827
IBM	3.0310	2.8720	2.8020	6.6558	7.4636	7.8581
YAHOO	10.6003	10.6299	11.6819	10.8201	10.3690	17.1500

Table 4: MSE of the Backpropagation and RBF network for different architectures.

5. Conclusion

In this study we have compared the support vector machines for regression with Backpropagation and RBF networks. The support vector machines for regression is a robust technique for function approximation. Preliminary computational results in the MATLAB environment seem quite promising. We currently investigate decomposition interior point optimization method for large scale problems.

Acknowledgment: The present work has been supported by the NSF grant ECS-9978813.

References

[1] N. Ancona, Classification Properties of Support Vector Machines for Regression, *Technical Report*, RI-IESI/CNR-Nr. 02/99.

[2] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, *Nonlinear Programming: Theory and Algorithms*, John Wiley & Sons Inc., New York, 1993.

[3] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, Volume 2, pp. 1-43, Kluwer Academic Publishers, Boston, 1998.

[4] C. Cortes and V. Vapnik, Support Vector Networks, *Machine Learning*, 20, 273-297, 1995.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Inc., New Jersey, 1999

[6] T. Joachims, Making Large-Scale SVM Learning Practical, *Technical Report*, LS-8-24, Computer Science Department, University of Dortmund, 1998.

[7] M.Pontil and A. Verri, Properties of Support Vector Machines, *Technical Report*, Massachusetts Institute of Technology, 1997.

[8] E.E. Osuna, R. Freund and F. Girosi, Support Vector Machines: Training and Applications, *Technical Report*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, AI Memo No. 1602, , 1997.

[9] A.J. Smola and B. Scholkopf, A Tutorial on Support Vector Regression, NEUROCOLT2 Technical Report Series, NC2-TR-1998-030, 1998.

[10] B. Scholkopf, P. Bartlett, A. Smola and R. Williamson, Shrinking the Tube: A New Support Vector Regression Algorithm.

[11] T.B. Trafalis, Primal-Dual Optimization Methods in Neural Networks and Support Vector Machines Training, Working Paper, School of Industrial Engineering, University of Oklahoma, 1999.

[12] R.J. Vanderbei, Interior Point Methods: Algorithms and Formulations, ORSA Journal on Computing, Vol. 6, No. 1, pp. 32-34,1995.

[13] R.J. Vanderbei, LOQO an Interior Point Code for Quadratic Programming, *Technical Report*, Statistics and Operations Research, Princeton University, SOQ-94-15, 1998.

[14] H. White, Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns, in R.R. Trippi, E. Turban (Eds.), *Neural Networks in Finance and Investing*, pp. 315-328, 1993.