Support Function Machines

Jiuzhen Liang

School of Information Technology, Jiangnan University
1800 Lihu Road, Wuxi, Jiangsu Province, CHINA 214122
jz.liang@yahoo.com.cn

Abstract. This paper proposes a novel model of support function machine (SFM) for time series predictions. Two machine learning models, namely, support vector machines (SVM) and procedural neural networks (PNN) are compared in solving time series and they inspire the creation of SFM. SFM aims to extend the support vectors to spatiotemporal domain, in which each component of vectors is a function with respect to time. In the view of the function, SFM transfers a vector function of time to a static vector. Similar to the SVM training procedure, the corresponding learning algorithm for SFM is presented, which is equivalent to solving a quadratic programming. Moreover, two practical examples are investigated and the experimental results illustrate the feasibility of SFM in modeling time series predictions.

Keywords: support vector machine; learning algorithm; support function; procedure neural networks; time series predictions

1 Introduction

There has been more than ten years since support vector machines(SVM), the most popular machine learning model, was invented by V. Vapnik [1]. This decade discerned fast development of research on SVM both in theory and application. SVM as a general classifier or regression function has shown its great fascination than other models, such as neural networks, especially when samples are sparse and the established model by SVM has good(may be the best) generalization capability. SVM is recently of increasing interest more and more due to its promising empirical performance compared with other learning techniques [2]. Instead of using empirical risk minimization(ERM), which is commonly used in traditional learning, SVM is founded on structural risk minimization(SRM). ERM only minimizes the error occurred to training data whilst SRM minimizes an upper bound of the generalization error. This enables SVM to generalize well. The basic principle of SVM is to map the input space to a high-dimensional feature space using kernel techniques. A linear discriminant analysis is then formulated in the feature space to maximize the margin between two classes so as to maximize the generalization ability. Moreover, a discriminant analysis process is conducted based on a set of support vectors which are selected automatically from training data [3].

For time series predictions, SVM has been utilized as a regression function [4]. But while preparing samples for SVM, all functions which are dispersed in a certain interval of time, have to be transferred to spacial vectors. So essentially, SVM still performs functions that map static vectors from one space to another. Recently, procedural neural networks (PNN) was proposed for spatiotemporal modeling [7]. PNN is a temporal neural networks model which aims to simulate time series predictions. Different from the classic neural networks (NN) in which neurons can not change state with respect to time, PNN combines the spatial and temporal information together, namely neurons process information both from space and time simultaneously. Based on PNN, some other models and learning strategies can be modified to simulate time series as well. In the past years, PNN models, properties and learning algorithms have been approached, such as complex number procedure neural networks [5], functional procedure neural networks [6], approximation ability to functional function [8] and trainings [9]. Naturally, the two models, SVM and PNN, inspire the following motivations: Can we find such a model, which shares the form with SVM, has the function similar to PNN and simulates time series better? Can we minimize the generalization error of PNN by introducing such support functions? This paper will investigate these problems and try to construct an efficient model for time series predictions.

The rest of this paper is organized as follows. In section two, a new model named support function machine (SFM) is established for simulating spatiotemporal problems. Section three deals with the learning algorithm for classification and regression respectively referring to the quadratic maximum problem of SVM, and for classification problems detail learning steps are presented. In section four, two examples, harm forecasts and stock predictions, are investigated. Finally, conclusions are given in section five.

2 Support Function Machines Model

Suppose we have N input patterns, $x_i \in \mathbb{R}^n$ be the i-th input pattern, where n is the number of the input variables, and y_i be the corresponding label of x_i. A SVM model based on the support vectors found through learning is defined as

$$f(\boldsymbol{x}, \alpha) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \alpha_0 \tag{1}$$

where, $K(\boldsymbol{x}, \boldsymbol{x}_i)$ is called a kernel function and $\alpha = (\alpha_0, \alpha_1, \cdots, \alpha_N)$ is the parameter vector needed to be confirmed.

In contrast to SVM, SFM is based on training data pairs ($\mathbf{x}_i(t)$, $\mathbf{y}_i$), in which $t \in [T_1, T_2]$, input patterns $\mathbf{x}_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{in}(t))$ and labels $\mathbf{y}_i = (y_{i1}, y_{i2}, \cdots, y_{im})$, where $i = 1, 2, \cdots, N$. Here m is the dimension of output, T_1 and T_2 are time boundary points. In this paper we only consider the case $n \geqslant 1, m = 1$ which corresponds to a multi-input and mono-output system. Different from SVM, all training data in SFM are vector functions of time which are discrete in interval $[T_1, T_2]$. In real world, sometimes data sampled in $[T_1, T_2]$ are not simultaneously recorded, i.e. in different dimensions different $t \in [T_1, T_2]$ are used as the sampling time points. This makes SVM difficult to simulate, because traditionally in SVM all the components of a vector must be captured simultaneously.

We denote $\boldsymbol{x}(t)$ as a general vector function and $\boldsymbol{x}_i(t)$ as the i-th support function (corresponding to the i-th support vector in SVM). Then a SFM classifier based on the support functions can be defined as

$$\mathbf{y} = f(\boldsymbol{x}(t), \alpha) = \sum_{i=1}^{N} \alpha_i y_i K(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) + \alpha_0 \tag{2}$$

which describes a functional relation between the vector function $\boldsymbol{x}(t)$ and the vector $\boldsymbol{y}$. The structure of SFM is much similar to SVM as in Fig. 1. Here, we omit the first subscripts of input vector $\boldsymbol{x}(t) = (x_1(t), x_2(t), \cdots, x_n(t))$ which denote the sequence numbers of samples.

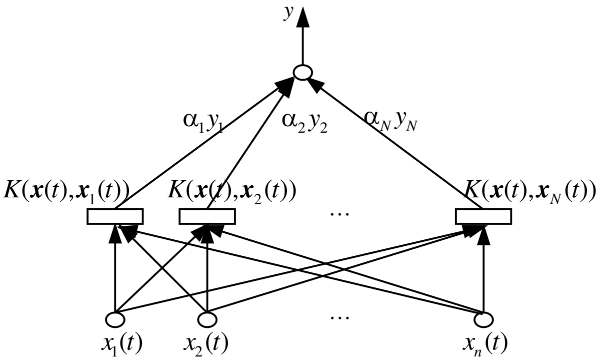

Fig. 1: Support function machine model

A major revision is then made to determine the kernel function using functional similarity for recognizing different attributes. The kernel functional function of the i-th support function is defined as

$$K(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) = \exp(\beta \frac{s(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) - b_i}{b_i}) \tag{3}$$

where $s(\boldsymbol{x}(t), \boldsymbol{x}_i(t))$ is a pair-wise similarity between function $\boldsymbol{x}(t)$ and $\boldsymbol{x}_i(t), b_i$ is the maximum similarity associated with the i-th support function and β is a constant. It can be seen that the term in the exponent function, which can be denoted as

$$c(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) = \frac{s(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) - b_i}{b_i} \tag{4}$$

is in general non-positive.

In this study, we revise the form of bio-SVM [3] and promote SFM, which can be regarded as the generalization of bio-SVM. Actually, there are many ways to define the similarity of two functions and each similarity corresponds to a distance between two functions, for example, in Euclid space

$$s(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) = \frac{1}{T_1 - T_2} \int_{T_1}^{T_2} \frac{\boldsymbol{x}(t) \cdot \boldsymbol{x}_i(t)}{\|\boldsymbol{x}(t)\| \|\boldsymbol{x}_i(t)\|} \mathrm{d}t \tag{5}$$

in which '$\cdot$' denotes the inter-product of two vectors, $\|\boldsymbol{x}(t)\|$ denotes a norm of the vector, while $\boldsymbol{x}(t)$ and $\boldsymbol{x}_i(t)$ are continuous in $[T_1, T_2]$. If $\boldsymbol{x}(t)$ is a non-numerical series function, for instance, a sequence of discrete data $\boldsymbol{x}(t)|_{t\in[T_1,T_2]} = \{x_{ljk}|j = 1,\cdots,n; k = 1,\cdots,n_j\}$ and $\boldsymbol{x}_i(t) = \{x_{ijk}|j = 1,\cdots,n; k = 1,\cdots,n_j\}$, here, $l, i = 1,\cdots,N$ and n_j is the hits number of component $x_{ij}(t)$ in $[T_1, T_2]$, the similarity can be defined as

$$s(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) = \frac{\|\boldsymbol{x}(t) \cap \boldsymbol{x}_i(t)\|}{\|\boldsymbol{x}(t) \cup \boldsymbol{x}_i(t)\|} \tag{6}$$

where $\|A\|$ denotes the size of the set A.

In this paper, for the case of numerical samples in the Euclid space and $\boldsymbol{x}_i(t_k) = \{x_{ijk}|j = 1,\cdots,n\}$, we define the corresponding vector function similarity as

$$s(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) = \frac{1}{M}\sum_{k=1}^{M} \frac{\boldsymbol{x}(t_k) \cdot \boldsymbol{x}_i(t_k)}{\|\boldsymbol{x}(t_k)\|\|\boldsymbol{x}_i(t_k)\|} \tag{7}$$

in which M denotes the sampling scale in the temporal domain.

Unfortunately, the similarity definition can not guarantee the corresponding kernel function to be semidefinite positive, which is a Mercer's kernel. In fact, it is well known that the exponential of a distance is not, in general, a Mercer's kernel. However, the proposed kernel can always be made semidefinite positive by appropriate choice of parameter (β as in (3)) [14]. In particular, as long as no two samples in the training set are exactly alike, it is always possible to make the kernel matrix diagonally dominant, and therefore semidefinite positive, by making β sufficiently large. Therefore, the positive definiteness of the kernel can usually be checked by evaluating the positive definiteness of the kernel matrix obtained with the data sets.

3 Learning Algorithm

Typically, a learning algorithm for SVM is equivalent to solving a quadratic maximum or minimum problem. This strategy is also valid for SFM except that it is different in computing the kernel function. In the case of classification, the primal objective function (which should be minimized) is

$$E(\alpha) = \frac{1}{2}\sum_{i,j=1}^{N} \alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) - \sum_{i=1}^{N}\alpha_i \tag{8}$$

subject to the box constraint

$$0 \le \alpha_i \le C, \forall i \tag{9}$$

and the linear constraint

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{10}$$

Here C is the restriction bound and can be different in each loop of computing α. In the process of searching optimal α_i, for instance, in the s-th iteration, it takes the form as

$$C(s) = \max_i \{\alpha_i(s)\} + 1 \tag{11}$$

This is a quadratic restrictive optimal problem with respect to α_i and can be solved by a gradient descend strategy, i.e. update α by

$$\alpha_i(s+1) = \min\{C(s), \max\{0, \alpha_i(s) - \Delta\alpha_i(s)\}\} \tag{12}$$

where

$$\Delta\alpha_i(s) = \eta(y_i \sum_{j=1}^{N} \alpha_j(s) y_j K(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) - 1) \tag{13}$$

Here, η is learning rate, $\eta \in (0, 1)$, and typically η is fixed.

While for regression functional function, similar to [10] SFM minimizes functional function as follows.

$$E(\alpha) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(\boldsymbol{x}_i(t), \alpha)|_\varepsilon + \|\alpha\|^2, \tag{14}$$

where $|x|_\varepsilon$ is an ε-insensitive error function defined as

$$|x| = \begin{cases} 0, & \text{if } |x| < \varepsilon \\ |x| - \varepsilon, & \text{otherwise} \end{cases} \tag{15}$$

and the output of SFM has the following form

$$f(\boldsymbol{x}(t), \alpha^*, \alpha) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\boldsymbol{x}(t), \boldsymbol{x}_i(t)) + \alpha_0 \tag{16}$$

Intuitively, α_i^* and α_i are ”positive” and ”negative” Lagrange multipliers (i.e., a single weight) that obey α_i^*, $\alpha_i \geq 0$, $\forall i$, and $\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0$.

Referring to the dual theorem [11], the primal form of equation (16) can be written as

$$\begin{aligned} L(\alpha^*, \alpha) =& \varepsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{N} y_i (\alpha_i^* - \alpha_i) \\ &+ \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\boldsymbol{x}_j(t), \boldsymbol{x}_i(t)), \end{aligned} \tag{17}$$

where the objective function should be minimized with respect to α^* and α, subject to the constraints

$$\begin{cases} \displaystyle\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0 \\ 0 \leq \alpha_i^*, \alpha_i \leq C, \forall i, \end{cases} \tag{18}$$

here the parameter C is the same user-defined constant that represents a balance between the model complexity and the approximation error.

For the case of classification, the algorithm can be described in detail as follows.

Step 1. Initialize all parameters including SFM structure parameters (e.g. α) and learning algorithm parameters (e.g. error precision ε).

Step 2. Choose an appropriate β and compute the kernel matrix by equation (3) until it is semidefinite positive.

Step 3. Compute error function E_1 according to equation (8) for all training samples.

Step 4. Update parameters α by equation (12) and equation (13).

Step 5. Compute error function E_2 by equation (8).

Step 6. If $|E_2 - E_1| < \varepsilon$ stop, otherwise let $E_1 = E_2$, go to step 3.

4 Application Examples

4.1 Harm Forecast of Horsetail Pine Worms

Horsetail pine worm is one of major harms to the forest in the southeast China. Although much more has been done in development of techniques to protect forest source, it is necessary to forecast the degree of worm harm in the coming season. The data is provided by the Institute of Jinhua Epidemic Prevention. Each record includes four segments, namely, later age through winter(LA), the first age(FA), the second age(SA), and the anterior age through winter(AA). For each segment there are eight observation fields including light-degree harm (LDH), middle-degree harm (MDH), heavy-degree harm (HDH), total amount of worms(TAW), average amount of worm in one tree (AAW), average amount of tree worm-harmed (AAT), area worm-harmed (AWH) and area of one degree worm-harmed (AOD). In Table 1 we just give two records of the horsetail worm harm affection from 1995 to 1996. Data in one year naturally form one record corresponding to one sample for training SFM. In each sample the four segments correspond to four values for each field in the temporal domain. The seven columns of values (except HDH) compose the input vectors for SFM, and the labels of heavy degree harm (HDH) in the associated seasons are the desire outputs of SFM (1 denotes heavy and -1 denotes not heavy).

Table 1: Two samples of horsetail worm harm

Segments	LDH	MDH	HDH	TAW	AAW	AAT	AWH	AOD
LA	22636	899	-1	23535	2.1	42.2	178581	155046
FA	64798	17867	-1	82665	3.5	37.8	262990	180325
SA	113212	50202	1	163414	3.9	42.3	388334	224920
AA	154515	36068	1	192430	4.5	47.1	411479	219049
LA	115497	13665	1	129162	4.2	47.0	353690	224528
FA	147156	27869	1	175025	4.4	57.6	445607	270582
SA	125226	21048	1	146274	3.0	50.0	435536	289262
AA	63182	280	-1	63462	2.4	53.4	313313	249851

Totally 39 samples are used in this experiment including training and test. Five groups of data are designed and in each group data is randomly divided into two sets: training and test. The experiment results are summarized in table 2.

Table 2: Warm harm heavy degree predictions

Training samples	Test samples	Error samples	Accuracy
20	19	10	47.37 %
26	13	4	69.23 %
30	9	2	77.78 %
32	7	1	85.71 %
34	5	0	100 %

4.2 Stock Price Predictions

It is a practically interesting and challenging topic to predict the trends of a stock price. Fundamental and technical analysis were the first two methods used to forecast stock prices. Various technical, fundamental, and statistical indicators have been proposed and used with varying results. However, no one technique or combination of techniques has been successful enough to consistently ”beat the market”[12]. Much more work has been done on stock markets predictions. We do not discuss more theories on stock market here, and we regard it as a typical example of time series. The stock data comes from Yahoo finance web site[13] in the period from 01/01/2000 to 06/30/2001. Data in one week or five days composes a sample of SFM. For each sample there are five observation fields including the open price, the highest price, the lowest price, the closing price and the stock volume. Table 3 lists two samples of Yahoo stock from 01/03/2000 to 01/14/2000.

Table 3: Ten records of a stock price list

Date	Open	High	Low	Close	Volume
01/03/2000	153.00	153.69	149.19	150.00	22069800
01/04/2000	147.25	148.00	144.00	144.00	22121400
01/05/2000	143.75	147.00	142.56	143.75	27292800
01/06/2000	143.13	146.94	142.63	145.67	19873200
01/07/2000	148.00	151.88	147.00	151.31	20141400
01/10/2000	152.69	154.06	151.13	151.25	15226500
01/11/2000	151.00	152.69	150.63	151.50	15123000
01/12/2000	151.06	153.25	150.56	152.00	18342300
01/13/2000	153.13	154.94	153.00	153.75	14953500
01/14/2000	153.38	154.63	149.56	151.00	18480300

In this experiment we choose 100 samples continuously from the data list, and for each prediction we select 20 samples, which are the closest to the prediction one in date, in training SFM. Each sample is composed of data from five sequential days. The prediction and actual values for open price are plotted in Fig. 2, and so are for close price in Fig. 3.

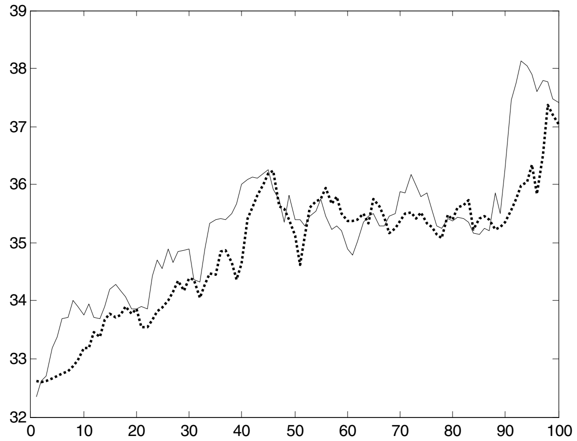

Fig. 2: Predictive and actual value of open price. Here the solid curve denotes the actual price of the stock, the dotted curve indicates the prediction value, the horizontal direction shows the date and the vertical direction presents the stock price.

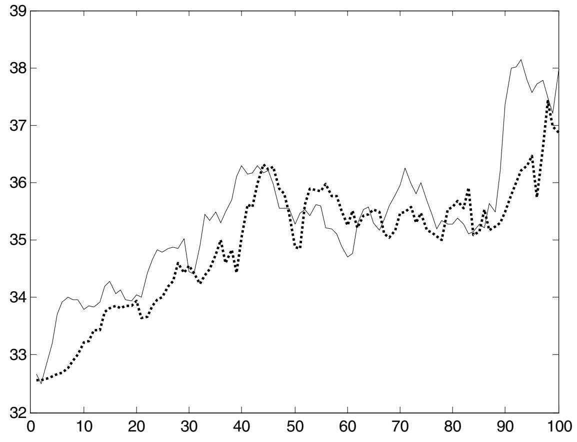

Fig. 3: Predictive and actual value of close price. The meanings of the two curves in this figure are the same as in Fig.2. Here 100 records of daily stock prices are investigated as the test set of SFM model. For the sake of intuition, all the discrete points are connected with line and each of the curve tendency is plotted clearly.

5 Conclusions

The purpose of this paper is to generalize SVM and promote SFM in which input patterns are functions of time. In the case of classifications, sometimes we have to deal with such problems that need to separate vector functions in a function space. While the task of function regression is to simulate time series in a spatiotemporal domain. The main contribution of this paper is to define the similarity between two vector functions and give the kernel function of vector functions. The learning algorithm for SFM is in no discrimination with that of the traditional SVM except the kernel functions. In real world, there are many problems associated with a procedure or varied with respect to time. SFM provides a new attempt to model such time series issues.

Acknowledgement This work is partly supported by CMP laboratory, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University. The author is supported by the Agreement between Czech Ministry of Education and Chinese Ministry of Education. Thanks to Pavol Vlcek for his suggestions on revision. Thanks to the session chair and two referees for their very valuable comments and suggestions.

References

1. Vapnik,V.: The nature of statistical learning theory. Springer, New York.(1995)
2. Scholkopf, B., Sung, K.K., Burges, C.J.C., Girosi, F., Niyogi, P., Poggio, T., Vapnik, V.: Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE Trans. Signal Process. **45**(1997) 2758–2765
3. Yang, Z.R., Chou, K.-C.: Bio-support vector machines for computational proteomics. Bioinformatics. **20**(2004)735–741
4. Vapnik, V.: The support vector method of function estimation. In Suykens, J.A.K. and Vandewalle, J.(eds) Nonlinear Modeling: Advanced Black-Box Techniques. Kluwer, Boston, MA. (1998)55–85
5. Liang, J.Z, Han, J.M.: Complex number procedure neural networks, In: Proc. of the First International Conference on Natural Computation. Part I, Springer. (2005)336-339
6. Liang, J.Z.: Functional procedure neural networks. Dynamic of Continuous Discrete and Impulsive Systems-Series B-Applications & Algorithms 1: Sp. Iss. SI. (2005)27–31
7. Liang, J.Z., Zhou, J.Q., He X.G.: Procedure neural networks with supervised learning. In: the 9th International Conference on Neural Information Processing. Singapore. (2002)523–527
8. He, X.G., Liang, J.Z.: Some theoretic problems of procedure neural networks. Engineering Science in China. **2**(2000)40–44
9. He, X.G., Liang, J.Z., Xu, S.H.: Training and application of procedure neural networks, Engineering Science in China. **3**(2001)31–35
10. Flake, G., Lawrence, S.: Efficient SVM regression training with SMO. Machine Learning. **46**(2002)271–290
11. Nello, C., John, S.T.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press. (2000)
12. Lawrence, R.: Using neural networks to forecast stock market prices. Reports. http://people.ok.ubc.ca/rlawrenc/research/Papers/nn.pdf (1997)
13. http://finance.yahoo.com/q/hp?s=GE&a=00&b=1&c=2007&d=06&e=26&f=2007&g=d (2007)
14. Chan, A., Vasconcelos, N. Moreno, P.J.: A family of probabilistic kernels based on information divergence. Technical Report SVCL-TR-2004-01, June 2004.