Margin Variations in Support Vector Regression for the Stock Market Prediction

YANG, Haiqin

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Department of Computer Science & Engineering

June, 2003

Margin Variations in Support Vector Regression for the Stock Market Prediction

submitted by

YANG, Haiqin

for the degree of Master of Philosophy

at the Chinese University of Hong Kong

Abstract

Support Vector Regression (SVR) has been applied successfully to financial time series prediction recently. In SVR, the ε-insensitive loss function is usually used to measure the empirical risk. The margin in this loss function is fixed and symmetrical. Typically, researchers have used methods such as cross-validation or random selection to select a suitable ε for that particular data set. In addition, financial time series are usually embedded with noise and the associated risk varies with time. Using a fixed and symmetrical margin may have more risk inducing bad results and may lack the ability to capture the information of stock market promptly.

In order to improve the prediction accuracy and to consider reducing the downside risk, we extend the standard SVR by varying the margin. By varying the width of the margin, we can reflect the change of volatility in the financial data; by controlling the symmetry of margins, we are able to reduce the down-side risk. Therefore, we focus on the study of setting the width of the margin and also the study of its symmetry property.

For setting the width of margin, the Momentum (also including asymmetrical margin control) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models are considered. Experiments are performed on two indices: Hang Seng Index (HSI) and Dow Jones Industrial Average (DJIA) for the Momentum method and three indices: Nikkei225, DJIA and FTSE100, for GARCH models, respectively. The experimental results indicate that these methods improve the predictive performance comparing with the standard SVR and benchmark model. On the study of the symmetry property, we give a sufficient condition to prove that the predicted value is monotone decreasing to the increase of the up margin. Therefore, we can reduce the predictive downside risk, or keep it zero, by increasing the up margin. An algorithm is also proposed to test the validity of this condition, such that we may know the changing trend of predictive downside risk by only running this algorithm on the training data set without performing actual prediction procedure. Experimental results also validate our analysis.

摘要

近年來，支持矢量回歸模型（Support Vector Regression, 簡稱 SVR）已經成功的應用於金融時序預測領域。 在 SVR 中，對 ε 不敏感的損失函數通常被用於測量經驗誤差。 該函數中的邊界是固定並且是對稱的。 研究人員通常採用某些方法，例如交叉確認的方法或隨機選取的方法為某個特殊的數據集選取一個適合的 ε 值。 另外，金融時序通常內嵌著噪聲，而且相應的風險隨時間變化。 使用一個固定和對稱的邊界可能有更大的風險導致壞的結果，並且缺少及時捕獲市場信息的能力。

爲了增加預測的準確性和爲了考慮減少下行風險，我們通過改變邊界拓展標準的 SVR。 通過改變邊界的寬度，我們能夠反映金融數據波動性的變化趨勢；通過控制邊界的對稱性，我們能夠減少下行風險。 因此，我們集中於邊界寬度設定的研究和邊界對稱性的研究。

對於邊界寬度的設定，我們考慮了動量（該方法還包括了不對稱邊界的控制）和一般自我回歸條件異質性模型（Generalized Autoregressive Conditional Heteroskedasticity，簡稱 GARCH)。對於動量模型，我們分別對兩支指數：恆生指數和道瓊斯工業平均指數做了實驗。 對於 GARCH 模型，我們分別對三支指數：日經 225 指數，道瓊斯工業平均指數和富時 100 指數做了實驗。 實驗結果表明跟標準的 SVR 和基準模型相比較，這些方法將提高預測的性能。在對稱性方面的研究，我們提出一個條件證明當上邊界增大的時候，被預測的值會單調減少。 因此，我們可以通過增加上邊界的值使預測的下行風險減少或让它保持為零。 於是，我們提出了一個算法去检验該條件的正確性。 這樣，我們可以僅僅通過在訓練數據集上運行該算法，而不需要進行真正的預測實驗，就可能知道預測的下行風險的變化趨勢。 實驗結果同樣驗證了我們的分析。

Acknowledgment

There are many people that I would want to thank. At first, I am grateful to my both supervisors: Prof. Irwin King and Prof. Chan, Laiwan. Prof. Irwin King suggested me to study this topic – Support Vector Machines (SVMs) and honed my presentation and writing skills. Especially the experience of helping me to write the first published paper is very important. Prof. Chan made me to know how to begin to do research, helped me to stride out of the first step from my turtle shell and asked me to change the margin in SVR. Another professor I want to thank is Prof. Xu, Lei. Although I know I learn little technical things from his course, Learning Theory and Computational Finance, I have taken in the methodology of research and applied them in my personal work. From his course, I began to be aware of what is research about, what are the differences between studying a new course and doing a new research topic and the strategies of doing these two things well.

I also want to thank my colleagues and my playing friends (football teammates). The time in the beginning of my research is the hardest time for me. There are several persons have provided me useful help. They are Precious (Zhang, Wan – a M.Phil. student of CSE[†] , CUHK) and Hu, Xuelei (a Ph.D. student of CSE, CUHK), who have given me beneficial discussions with SVMs, Richard (Sia, Ka Cheung – a M.Phil. student of CSE, CUHK), who has given me helpful suggestions on the familiarizing of technical things. I also want to thank Li, Rui (a M.Phil. student of SEEM[†] , CUHK), who has provided me many examples of mathematical preciseness and basic concepts of optimization in the discussion of the convex optimization. His helpful suggestions let me believe I can prove the results of fixed margin cases in Chapter 6. Xu, Haifeng (a Master student of Mathematics department, Nanjing University) gave me

[†]Computer Science & Engineering
[†]Systems Engineering and Engineering Management

helpful suggestions that made me propose the detective algorithm in Chapter 6. Do, Shizhong (M.Phil. student of Mathematics department, CUHK) has provided me various mathematical concepts. Ma, Ke (a M.Phil. student of IE[†] , CUHK) has given me an example of from industry to academe and made me to do research firmly.

I have to thank my family and my uncles–they provided me the support of spirit. Zhu, Fengfeng (an associate professor of Department of Applied Mathematics in South China University of Technology) calmed down my heart and encouraged me to do good research.

[†]Information Engineering

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Time Series Prediction and Its Problems

Time series prediction, or time series forecasting, takes an existing series of data $x_{t-n}, \ldots, x_{t-2}, x_{t-1}, x_t$ and forecasts any of the future data values $x_{t+1}, x_{t+2}, \ldots$. The goal is to observe or to model the existing data series in order to forecast future unknown data values accurately. Examples of data series include financial data series (stocks, indices, foreign exchange rates, etc.), physically observed data series (sunspots, weather, etc.), and mathematical data series (Fibonacci sequence, integrals of differential equations, etc.). The phrase *time series* generically refers to any data series, irrespective of whether or not the data are dependent on a certain time increment.

Time series prediction has several important applications [11, 1, 10, 22]. For example, forecasting the network flow or identifying the network congestion circumstance based on the previous flow of network [10]. A more useful application is that people hope to profit by applying the time series prediction techniques in the financial markets. Whether this is viable or not is most likely a never-to-be-resolved question.

In this thesis, we focus on a recent model, Support Vector Machine (SVM), which has captured researchers' interest because of its mathematical tractability, geometric interpretation and practical use. We apply the regression model

of this learning machine, Support Vector Regression (SVR), in the financial time series prediction. Different to other traditional regression models, SVR not only minimizes the empirical risk (training error), but also minimizes a term which makes the objective function as flat as possible. Usually, the ε-insensitive loss function is used to measure the empirical risk. This loss function contains a 2ε margin; this margin is fixed and symmetrical. Typically, researchers have used methods such as cross-validation or random selection to select a suitable ε for that particular data set. Financial time series are usually embedded with noise and the associated risk varies with time. Using a fixed and symmetrical margin may have more risk inducing bad results and may lack the ability to capture the stock market information promptly.

In order to improve the prediction accuracy and to consider reducing the downside risk, we extend the standard SVR by varying the margin. By varying the width of the margin, we can reflect the change of volatility in the financial data; by controlling the symmetry of margins, we are able to reduce the downside risk.

1.2 Major Contributions

The main contributions of our work are:

1. We have extended the standard Support Vector Regression (SVR) by varying the margin and have applied it to financial prediction tasks [96, 98]. The original margin in SVR is fixed and symmetrical, which lacks the ability to capture the information of stock market promptly. We extend the margin setting by extending these two characteristics of margin, i. e., fixed margin vs. non-fixed margin, symmetrical margin vs. asymmetrical margin. The resultant models are classified into four categories.

2. Usually, the financial data are noisy and the associated risk is time-varying. By varying the width of margin in SVR, we could reflect the change in volatility of the financial data; by controlling the symmetry of margins, we are able to reduce the downside risk. Standard deviation is a statistical term that provides a good indication of volatility of stock market; the momentum term is used to measure the up and down trend of stock market. Therefore, we propose a novel approach, which combines both characteristics of margins, i. e., by using standard deviation of input $\mathbf{x}$ to set the width of margin and by using a momentum term to control the symmetry of margin in predicting the prices of Hang Seng Index and Dow Jones Industrial Average based on the prices time series. The experimental results show that this actually improves the performance of SVR model in the prediction [98].

3. We also apply Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, which can reflect the volatility of the financial time series over time, to determine the margin over time for return time series data. This also improves the performance of SVR model [97].

4. After studying the relation between downside risk and fixed margin settings, we give a sufficient (but not necessary) condition to prove that the predictive downside risk can be reduced or kept zero by increasing the up margin. We also propose a detective algorithm to check the validity of this sufficient condition, such that we may know the changing trend of predictive downside risk without running the actual SVR algorithm [97].

1.3 Thesis Organization

This thesis is organized as followed. Chapter 2 presents a procedure of building a time series analysis system and reviews various time series analysis models.

These models are classified into linear models and non-linear models. For linear model, we detail ARIMA models; for nonlinear models, we concentrate on SVMs and GARCH models. The recent model, Support Vector Regression (SVR), for time series analysis is presented in Chapter 3, where our review begins from the regression problem, to the loss function, to the kernel function, which makes a linear SVR model to the non-linear case. Chapter 3 also states the relation between SVR model and other models, Support Vector Classification (SVC) models, Ridge Regression models and Radial Basis Function networks. Chapter 4 addresses the problem that occurs in SVR for the time series prediction and provides a solution by using a general ε-insensitive loss function; the corresponding accuracy metrics and risk measurements for the experiments are also stated in this chapter. Chapter 5 considers the concrete setting of margins in the non-fixed cases and presents the experimental results for two kinds of margin settings by the Momentum method and GARCH models. Chapter 6 studies the downside risk and the asymmetrical margin settings, where we state that if the sufficient condition is valid, we can prove that the predictive downside risk will be reduced or kept zero when the up margin is increased. A detective algorithm is also proposed to check the validity of the condition. Finally, Chapter 7 briefly concludes this thesis and lists some future works about our model.

1.4 Notation

In this thesis, bold typeface will indicate vector or matrix quantities; normal typeface will be used for vector and matrix component and for scalars. The components of vectors and matrices are labeled with Greek indices. The vectors and matrices themselves are labeled with Roman indices.

Chapter 2

Literature Review

A time series is a collection of observations that measures the status of some activities over time [22, 23]. It is the historical record of some activities, with a consistency in the activity and the method of measurement, where the measurement is taken at equally spaced intervals, e. g. day, week, month, etc. In practice, there are various time series and they are used in a wide range of disciplines, from engineering to economics. For example, the air temperatures of a certain city measured in successive days or weeks consists of a series, a certain share prices occurred in successive days, months is another series.

Of all the different possible time series, the financial time series is unusual since it contains several specific characteristics:

Noisy–The financial time series is usually embedded with noise which may be so high that it has a relatively low signal-to-noise ratio. Although this type of noise can be reduced or removed by some techniques, such as smoothing methods or filters, it produces lag problem.

Non-stationary–The second characteristic is non-stationary, i. e., data that do not have the same statistical properties (e. g. mean and variance) at each point in time. This makes the forecasting very difficult. A common technique used to make a series stationary is to difference it. However, for financial time series, making training data set stationary

does not guarantee the stationary of testing data.

Uncertainty–The third characteristic is that both financial theory and its empirical time series contain an element of uncertainty [85], e. g. there are various definitions of asset volatility and the volatility is not directly observable.

An important task of analysis of time series is to predict the future values of the series based on the given observed time series, such as

$$\ldots, x_{t-3}, x_{t-2}, x_{t-1}, ?, ?, \ldots$$

Usually, the financial wellbeing of the whole organization operation depends on the accuracy of the forecast since such information will likely be used to make correlative budget and operative decisions, for example, investment, purchasing, marketing and capital financing. Any significant over-or-under sales forecast error may cause a firm to be overly burdened with excess inventory carrying costs or else to create lost sales revenue through unanticipated item stock. A more useful application is that people hope to profit by applying the time series prediction techniques to the financial markets. Whether this is viable or not is most likely a never-to-be-resolved question.

Before jumping to detail technical models for their own sakes, let's give a brief introduction of the framework of how to build a system to forecast the time series.

2.1 Framework

Forecasting is a necessary input to planning, whether in business or government. Usually, forecasts are generated subjectively and at great cost by group discussion. Modeling a practical problem and doing forecasting can offer an

objective information for future development. The flowchart in Fig 2.1 highlights different phases of such a modeling system. This system contains several functions:

Model Estimation–understand the underlying mechanism generating the time series; this includes describing and explaining any variation, seasonality, trend, etc.

Forecasting Generation–predict the future based on the assumption of "business as usual".

Forecasting Updating–control the system, that is to perform the "what-if" scenarios.

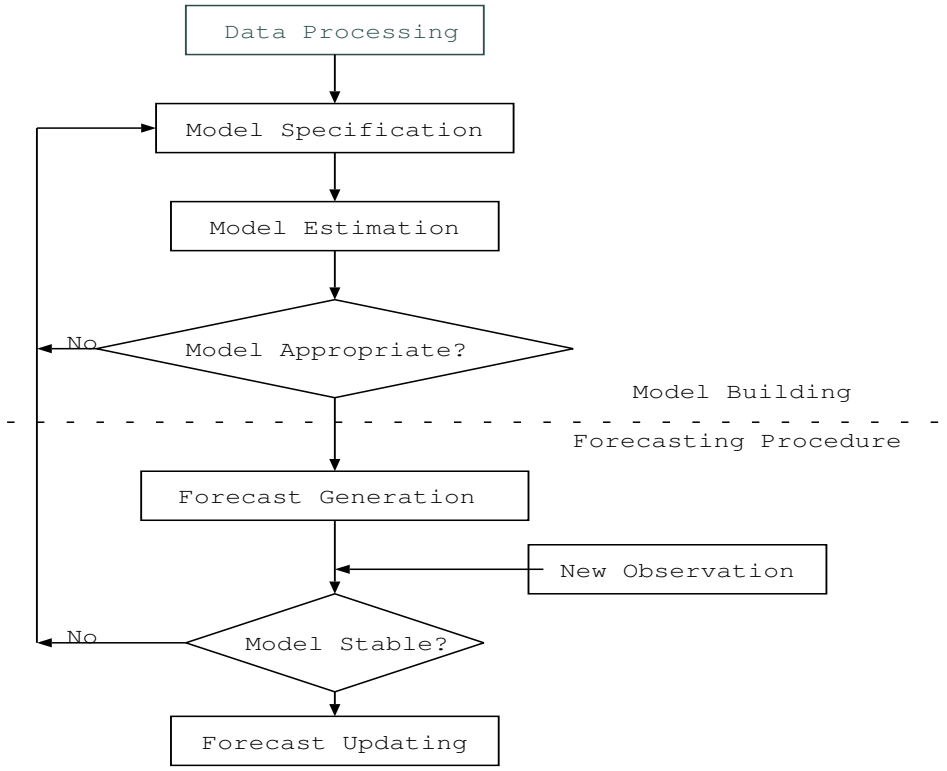


Figure 2.1: Model building and forecasting phases of a forecasting system.

2.1.1 Data Processing

Finding good representation for the data is a crucial and labor intensive task. Depending on different problems, it is necessary to perform some data processes in order to satisfy the requirement of the special models. For instance, we need preprocessing to remove seasonal effect, trend effect or cyclic oscillation in the data. Without performing such preprocessing on the data, we may, for example, incorrectly infer that recent increase patterns will continue indefinitely when actually the increase is simply because it is that time of the year. In the following, we will introduce two methods for data processing: *smoothing* and *differencing*.

Smoothing

Inherited in the collection of data taken over time is the form of random variation. There are some methods for reducing or canceling the effect of this random variation. An often-used technique is "smoothing". This technique, when properly applied, reveals more clearly the underlying trend, seasonal and cyclic components from the original data.

There are two distinct groups of smoothing methods:

1. Simple Moving Average (SMA) – a k-day SMA takes the average of previous k days' values as current day's value.

2. Exponential Moving Average (EMA) – a k-day EMA begins from the first day, $EMA_1 = x_1$, and set the i-day's value as $EMA_i = EMA_{i-1} \times (1 - \frac{2}{k}) + x_i \times \frac{2}{k}$.

For example, given a data series $x_1, x_2, x_3, x_4, \ldots$, after taking SMA with an interval of three, it becomes $[(x_1 + x_2 + x_3)/3]$, $[(x_2 + x_3 + x_4)/3], \ldots$. While using 3-days' EMA, the series becomes x_1, $\frac{1}{3}x_1 + \frac{2}{3}x_2$, $\frac{1}{9}x_1 + \frac{2}{9}x_2 + \frac{2}{3}x_3$, $\frac{1}{27}x_1 + \frac{2}{27}x_2 + \frac{2}{9}x_3 + \frac{2}{3}x_4, \ldots$.

Comparing SMA with EMA, we can conclude the following difference:

1. Taking k-days' SMA will reduce the number of data points in the series by $k-1$, while EMA still retains the same number of original data.
2. EMA gives more weight to the latest data than SMA.
3. EMA reacts faster to recent value changes than SMA.

Figure. 2.2 also illustrates these differences.

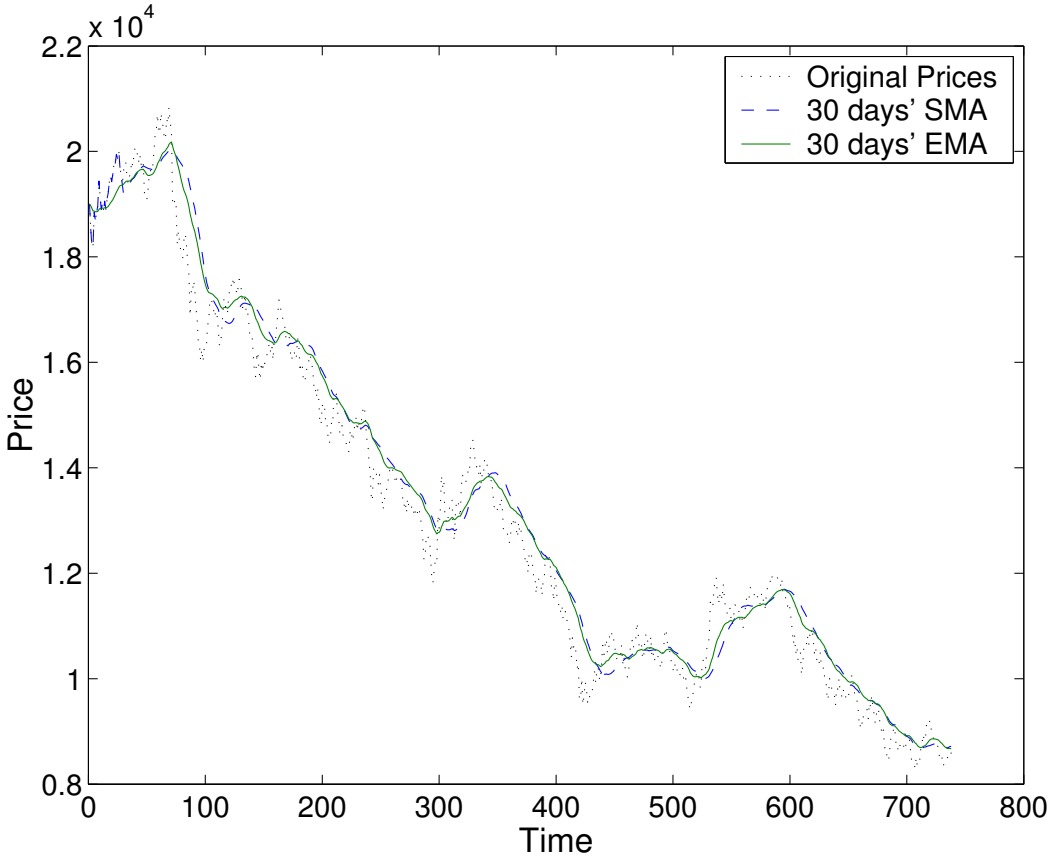
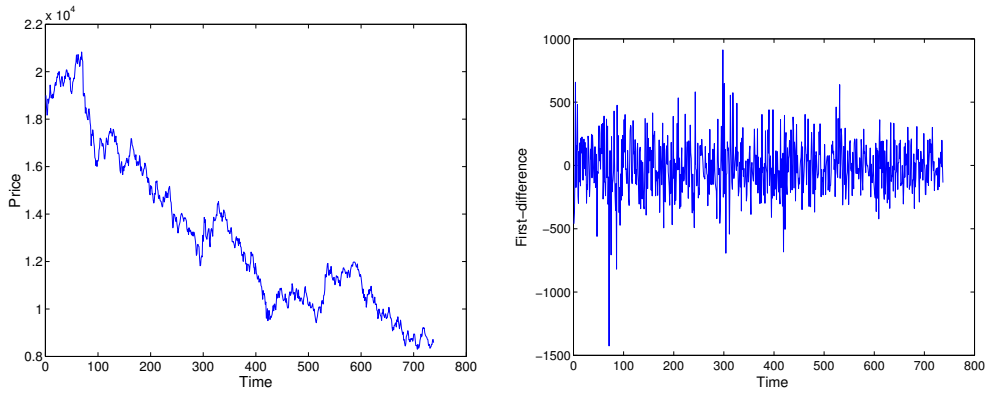


Figure 2.2: Experimental Data used in this thesis: daily closing prices of Japanese Nikkei225 from Jan. 04, 2000 to Dec. 30, 2002 with a 30 days' SMA and with a 30 days' EMA.

Differencing

Differencing is another method for preprocessing when there is a substantial trend in the data. Concretely, a data series $x_1, x_2, x_3, \ldots$ becomes $(x_2 - x_1), (x_3 - x_2), \ldots$ after taking a first-difference. Generally, the original time series becomes $\nabla x_t = x_{t+1} - x_t$ after first-difference. While this procedure usually makes a data series stationary in the mean. If not, a second-difference of the series can be taken. This procedure may be done until the series becomes stationary (the definition of stationary please see [22]. Other notes are that taking a first-difference will reduce the number of data points in the series by one, but the noise of data is cumulative. Figure 2.3 presents an original price series and the result of a financial index after first-differencing.



(a) Original prices (b) First-difference results

Figure 2.3: Experimental Data used in this thesis: daily closing prices of Japanese Nikkei225 from Jan. 04, 2000 to Dec. 30, 2002 with first-differencing

2.1.2 Model Building

After processing the original data, we should turn to the main problem: what does the model inhere in the given data; how to learn the model from the given data, or how to build the model based on the given data. The general techniques for time series analysis and prediction are classified into two

categories:

1. **Known Model Structure**–If the structure of the underlying model of a time series is revealed by a lot of information clearly, for example, the structure is linear, quadratic or periodic, etc., the main task left is then to estimate a few parameters of the model to fit the observation data. Sufficient observations will help to make this kind of model quite accurate and powerful. Unfortunately, for many practical problems, the underlying models are often unknown or ill-specified.

2. **Unknown Model Structure**–When the data does not reveal much information, the only thing available is a set of observations. For such problems, people often assume that the underlying model has some *state variables*, which determine what the values of the time series should be.

In [22], a general formulation for state space models is given to approximate nonlinear models as follows:

Let $\dots, x_{t-1}, x_t, x_{t+1}, \dots$ be a time series, it is assumed

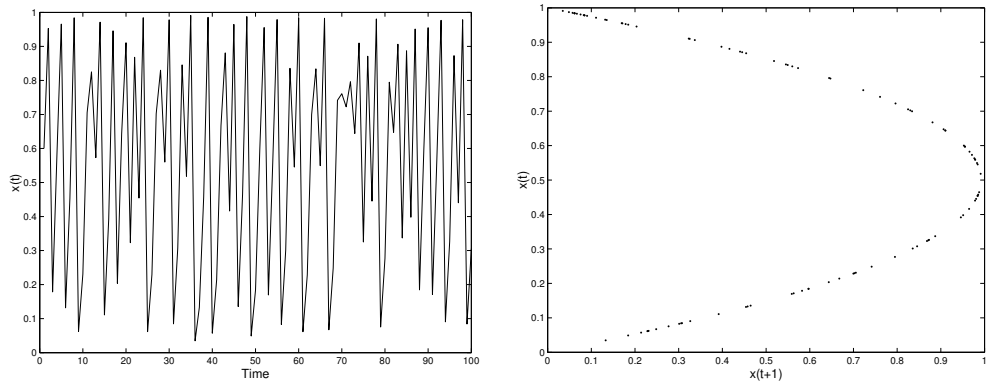
$$x_{t+1} = f(s^1_{t+1}, \dots, s^i_{t+1}, \dots, s^d_{t+1}) + \epsilon_{t+1},$$

where ϵ_{t+1} represents random noise at time $t+1$ and $s^1_{t+1}, \dots, s^d_{t+1}$ are *state variables*, and

$$s^i_{t+1} = g_i(s^1_t, \dots, s^k_t, \dots, s^d_t, x_t, x_{t-1}, \dots), \quad k = 1, 2, \dots, d,$$

where f and g_is are some functions. Note that x and s can be scale quantities for univariate model, but they could also be vector values for the more general multivariate setting. The motivation of using state variables is that they often correspond to certain features or properties of the time series and can help to understand and characterize the series. They can also help to simplify the

computations for analysis and prediction. Figure 2.4 gives a simple illustration: complicated time series may be represented simply by other state variables. In general, f is the objective function estimated by some models, and gs are some functions used to process the original data, such as those methods introduced in Subsection 2.1.1.



(a) (b)

Figure 2.4: A time plot (a) of the logistic function $x_{t+1} = 3.97 * x_t * (1 - x_t)$ is difficult to figure out, but a state space plot (b) clearly shows the underlying model. Using a state variable $s^1_{t+1} = x_t$, this example can be rewritten as $x_t = 3.97 * s^1_t * (1 - s^1_t)$.

2.1.3 Forecasting Procedure

Forecasting the future values of an observed time series is an important problem in many areas, e. g. economics, production planning, sales forecasting and stock control [22]. In [22], Chatfield classified the types of forecasting procedure into three categories:

Subjective Forecasts can be made subjectively based on judgement, intuition, commercial knowledge and any other information.

Univariate Forecasts can be made on an entire basis of past observations in a given time series, by fitting a model to the data and extrapolating. For

instance, forecasts of future sales of a product would be based entirely on past sales.

Multivariate Forecasts can be made by taking other observations or other variables into account. For example, sales may depend on stocks. Regression models are of these types of models, e. g. econometric models. The using of a leading indicator also comes into this category.

[22] also stated that a forecasting procedure may involve a combination of the above approaches practically. For instance, univariate forecasts are often computed, and then adjusted subjectively.

Sometimes the model may not be stable or need to be more accurate, the building model has to adjust to fit new observations. Therefore, other methods are proposed to adjust the model based on new observations automatically. For example, in [93], Wah and Qian presented new constraints on cross-validation to adjust their previous constructed model. Another interesting note in [93] is their assumption of stock price and their data preprocessing. They assumed stock prices consist of low-frequency and high-frequency components, where low-frequency components are predictive. Therefore, they applied low-pass filtering to the price time series at first. However, usually lag problem occurs when low-pass filtering is applied. Then they proposed methods to overcome this problem.

2.2 Model Descriptions

There are many models for time-series analysis. Here we classify them into *linear* and *nonlinear* models as in [23], see Figure 2.5.

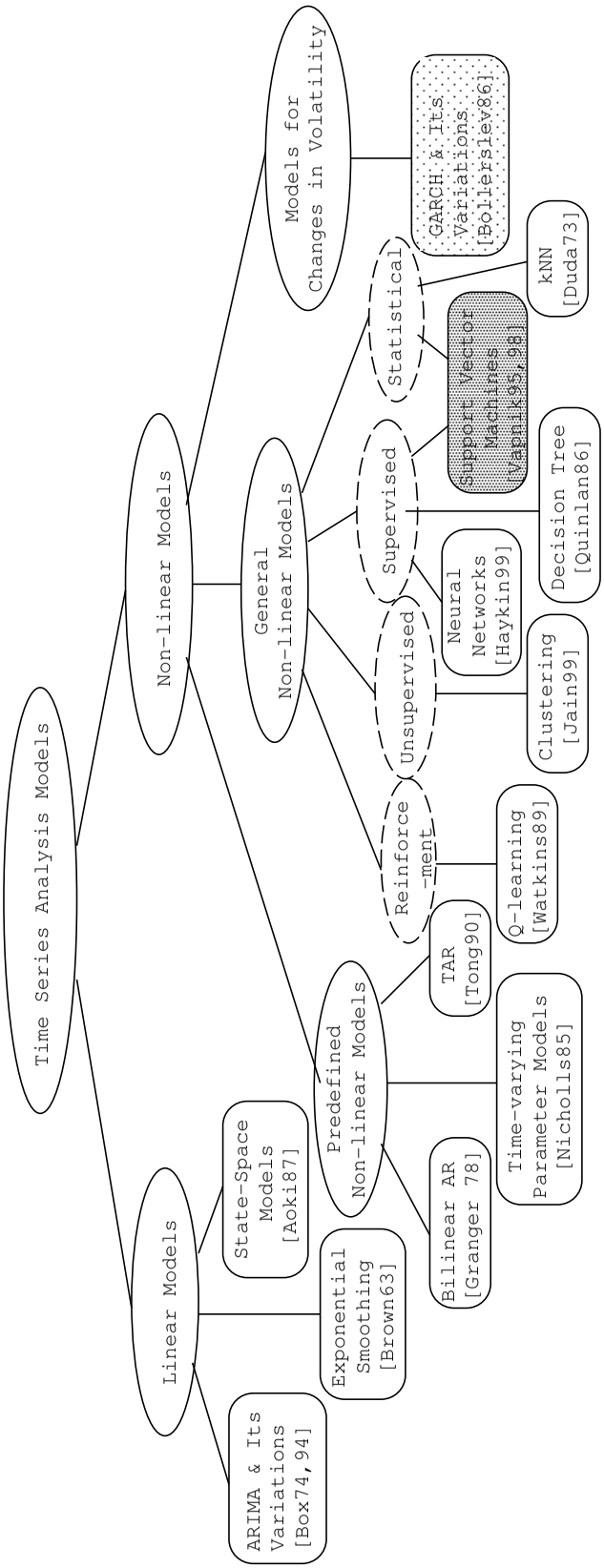


Figure 2.5: Time series analysis models.

2.2.1 Linear Models

Linear models have the following characteristics: simplicity, usefulness and easy application; and they work well for linear time series, but may fail otherwise. Here we present three types of linear models:

1. **ARIMA and Its Variations**

   Autoregressive integrated moving average (ARIMA) models have been developed by Box and Jenkins [10]. There are many variations have been produced over the last 30 years. The procedure of ARIMA model is to differnece a non-stationary time series, as differencing in Subsection 2.1.1, until stationary before applying (mixed) AutoRegressive Moving Average (ARMA) models. This approach is widely used in econometrics [10]. We will give a detailed description in Subsection 2.2.3

   The setting of an ARIMA models consist of five stages [10]:

   (a) Defferencing If the data is non-stationary, the data will be differencing until it becomes stationary.

   (b) Model Identification In this stage, the work is to examine the data to identify the model, i. e., to determine which order p and q will be most appropriate for the model. In general, there is no optimal way to do it. Some useful tools are the sample autocorrelation (ACF) and partial autocorrelation (PACF) functions. ACF measures the correlation between different lags of a time series, while PACF measures the residual correlation after the correlation implied from earlier lags is subtracted out.

   (c) Estimation In this stage, it is to estimate the parameters of the chosen model. Least squares method is usually used to find the parameters. Please see [10] for more details.

(d) Diagnostic Checking To check whether the building model is adequate, one method is to examine the residuals from the fitted model.

(e) Alternative Models Considering If the fitted model appears to be inadequate for some reasons, then other ARIMA models may be tried until a satisfactory model is found.

A simple variant of ARIMA is that if the time series is seasonal, then a seasonal ARIMA (SARIMA) model may be used to fit the series. However, although seasonal variation exists through the year, there is no particular reason to set the model coefficientconstant throughout the year. Therefore, periodic autoregressive (PAR) models provide a variant to SARIMA models wherein the value of the autoregressive parameters are allowed to vary through the seasonal cycle. More generally periodic correlation arises when the size of autocorrelation coefficients depends, not only on the lag, but also on the position in the seasonal cycle [23].

An interesting variant of ARIMA models is Fractional integrated ARMA (ARFIMA) [23]. The difference of ARFIMA is that this models allow the difference order to be non-integer instead of integer in ARIMA. But these models contain several drawbacks: (a) they are difficult to give an intuitive interpretation to a non-integer difference; (b) they are difficult to compute the fractional difference since the difference is a binomial expansion. The stationary ARFIMA models, with the difference order within 0 and 0.5, are a class of models called long-memory models [40].

2. Exponential Smoothing

Exponential smoothing models are another type of linear models [11]; they work well for linear time series but fail to model complicated non-linearity and trends in financial time series. One application of them

may be applied in data processing, e. g. Subsection 2.1.1.

3. State Space Models

State space models [1] are a class of linear models that represent inputs as a linear combination of a set of state vectors that evolve over time according to some linear equations. Different state space formulations cover a very range of models and include the so-called structural models in [42] as well as the dynamic linear models in [95], where the latter uses a bayesian formulation. Models called unobserved component models by econometricians are also of state-space form. However, in practice, state vectors and their dimensions of these models are hard to choose [23].

2.2.2 Non-linear Models

Although linear models have both mathematical and practical convenience, there is no reason why real life time series should all be linear, and so the use of non-linear models seems potential promising [23]. Here we consider the non-linear models in the following three types:

1. Predefined Non-linear Models

In the 1980's, non-linear models were investigated and were proposed from the existing linear models, e.g. ARIMA models, [40, 65]. For example, Bilinear autoregressive or Bilinear AR models [39], time-varying parameter models [67, 60] and threshold autoregressive (TAR) model [83]. These models are agreeable due to the scrutiny given in their development for the standard statistical considerations of model specification, estimation, and diagnosis, but their general parametric nature tends to require significant a prior knowledge of the form of relationship being modeled. Therefore, they are not effective for modeling financial time series because the nonlinear functions are hard to choose.

2. General Non-linear Models

Another class of nonlinear models are general non-linear models, also called machine learning. These models can learn a model from a given time series without non-linear assumptions. They include reinforcement learning, e. g., Q-learning [94], unsupervised learning, e. g., clustering methods [45], supervised learning, e. g., decision tree [66] and neural network (NN) models [68, 24, 3, 43], and statistical learning, e. g., k-nearest-neighbors(kNN) [30]. Support Vector Machines (SVMs) are new learning machines that also can model non-linear relationship of the data. SVMs are grounded on the statistical learning theory, or Vapnik-Chervonenkis (VC) theory. They also are modeled by a training sample with targets, and used to predict (classification/value) in a new testing sample. Therefore, SVMs fall both in the statistical learning and supervised learning. Detailed description will be given in Subsection 2.2.4.

3. Models for Change in Volatility

Models for changes in volatility are a completely different class of models that modeling changes in variance. The objective of these models is not to give better point forecasts of the observations in the given series but rather to give better estimates of the (local) variance which in turn allows more reliable prediction intervals to be computed, this can lead to a better assessment of risk [23].

The estimation of local variance is especially important in financial applications, where observed time series often show clear evidence of changing volatility, e. g. large absolute values tend to be followed by more large (absolute) values, while small absolute values are often followed by more small values, indicating high or low volatility, respectively.

To estimate the local variance, Engle in 1982 firstly provided a systematic framework for volatility modeling, the AutoRegressive Conditionally

Heteroscedastic (ARCH) model [32, 31]. The basic ideas of ARCH models are [85]:

1. the mean-corrected asset return is serially uncorrelated, but dependent, and
2. the dependence of asset return at time t can be described by a simple quadratic function of its lagged values.

An ARCH model with order p, in short ARCH(p), assumes that the variance at time t, σ_t^2 is linearly dependent on the last p squared values of the time series, i. e.,

$$r_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \alpha_1 r_{t-1}^2 + \ldots + \alpha_m r_{t-m}^2,$$

where r_t is a serial asset return, ϵ_t is a sequence of independent and identically distributed (i.i.d.) random variables with mean zero and variance 1, and α_i are coefficients must satisfy some regular conditions to ensure that the unconditional variance of r_t is finite with $\alpha_0 > 0$, $\alpha_i \geq 0$ for $i > 0$. In practice, ϵ_t is often assumed to follow the standard normal or a standardized student-t distribution.

ARCH models were extended by Bollerslev in 1986, a generalized ARCH (GARCH) model [8]. Similar to ARMA, a GARCH model can be used to estimate a high order ARCH model with fewer parameters. A GARCH model with order p and q, in short GARCH(p,q), assumes that the conditional variance depends on the squares of the last p values of the series and on the last q values of conditional variance, i. e.,

$$r_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2,$$

where again ϵ_t is a sequence of i.i.d. random variables with mean zero and variance 1, and $\alpha_0 > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$, and $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$.

The GARCH(1,1) model has become the 'standard' model for describing changing variance for no reason other than relative simplicity. There are also some extensions of the basic GARCH model, such as Quadratic GARCH (QGARCH) and Exponential GARCH (EGARCH). The QGARCH models allow for negative 'shocks' to have more effect on the conditional variance than positive 'shocks'. The EGARCH models allow an asymmetric response by modeling $\log \sigma_t^2$, rather than σ_t^2. Summaries of this family of models can be seen in [31].

Although GARCH models are applied in a wide range of problems usefully, they do have limitations:

1. GARCH models are only part of a solution. Although GARCH models are usually applied to return series, financial decisions are rarely based solely on expected returns and volatilities.
2. GARCH models are parametric specifications that operate best under relatively stable market conditions [38]. Although GARCH is explicitly designed to model time-varying conditional variances, GARCH models often fail to capture highly irregular phenomena, including wild market fluctuations (e. g. crashes and subsequent rebounds), and other highly unanticipated events that can lead to significant structural change.
3. GARCH models often fail to fully capture the fat tails observed in asset return series. Heteroskedasticity explains some of the fat tail behavior, but typically not all of it. Fat tail distributions, e. g. student-t, have been applied in GARCH modeling, but often the choice of distribution is a matter of trial and error.

2.2.3 ARMA Models

ARMA models are linear models to capture the linear correlation between any specified lags of a univariate time series and the error term of the model from previous time points. In general, an ARMA(p,q) model can be written as

$$x_t = \mu + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_p x_{t-p} + \epsilon_t + b_1 \epsilon_{t-1} + \ldots + b_q \epsilon_{t-q},$$

where μ is the mean of the time series and a's and b's are constant coefficients.

Moving average (MA) models are special cases of ARMA models. In these models, the observation in time t depends on the error term of the model from previous time points, usually these errors are considered as random events [22]. Generally, an MA(q) model is

$$x_t = \mu + \epsilon_t + b_1 \epsilon_{t-1} + \ldots + b_q \epsilon_{t-q}.$$

From the above formula, we can see that this MA is total different to the smoothing methods in Subsection 2.1.1, although they are with same names in different situations.

Autoregressive (AR) models are another special cases of ARMA models. In this model, the observation in time t is regressed not on other independent variables but on one or more of the lagged values of the time series [22, 10]. A general form of an AR(p) model is

$$x_t = \mu + a_1 x_{t-1} + a_2 x_{t-2} + \ldots + a_p x_{t-p} + \epsilon_t,$$

where ϵ_t is a purely random process (also called white noise) with mean zero and variance σ_ϵ^2.

The simplest example of AR models is the first-order case, i. e., AR(1), takes the following form

$$x_t = \mu + a x_{t-1} + \epsilon_t.$$

If $\mu = 0$ and $a = 1$, it is the best-known case of AR models, random walk model,

$$x_{t+1} = x_t + \epsilon_{t+1}.$$

This model is correlated to the Efficient Market Hypothesis (EMH). The EMH was developed by Fama [33, 34] and found broad acceptance in the financial community [54, 86].

The EMH, in its weak form, states that the past market prices and data are fully reflected in the price of asset, i. e., the movement of the price is unpredictable. The best prediction for a price is the current price and the actual prices follow what is called a random walk. The EMH is based on the assumption that all news is promptly incorporated in prices; since news is unpredictable (by definition), prices are unpredictable. Much effort has been expended trying to prove or disprove the EMH. Current opinion is that the theory has been disproved [75, 49], and much evidence suggests that the capital markets are not efficient [53].

If the EMH was true, then the best estimation of a financial time series is:

$$\hat{x}_{t+1} = x_t.$$

In other words, if the series is truly a random walk, then the best estimate for the next time period is equal to the current estimate. However, in this thesis, we assume that there is a predictable component of the series.

In summary, ARMA models are a combination of AR and MA models. An advantage of ARMA models lies in the fact that a stationary time series may often be described by an ARMA model involving fewer parameters than a MA or an AR model by itself [22].

2.2.4 Support Vector Machines

Support Vector Machines (SVMs) firstly appeared at COLT 1992 [9]. SVMs are grounded on the Statistical Learning Theory, or VC theory, which was first developed by Vapnik and his co-workers [87, 88, 89].

SVMs have the following advantages:

Theoretical Background

SVMs are based on the VC theory, which has been developed for the past thirty years. This theory claims to guarantee generalization, i. e., the generalization error was bounded by the sum of the training error (empirical risk) plus a term, the term depending on the VC dimension of the learning machine [87, 89].

Geometric Interpretation

SVMs were firstly proposed to solve classification problems. When constructing a SVM, the objective is not only to minimize the empirical risk, but also to maximize the margin [88, 6].

Global and Unique Solution

Training SVM leads to solve the Quadratic Programming (QP) problem. For any convex programming problem, every local solution will be also global. Therefore, SVM training always finds a global solution, and usually unique [13]. This is superior to NN, where NN usually falls in local minima [12].

Mathematical Tractability

Using a kernel function, SVMs can be thought as an alternative training technique for Polynomial, Radial Basis Function and Multi-Layer Perceptron classifiers, in which the weights of the network are found by solving a Quadratic Programming (QP) problem with linear inequality

and equality constraints, rather than by solving a non-convex, unconstrained minimization problem, as in standard neural network training techniques [62].

Due to the above advantages, SVMs have attracted researches' interest and have been applied in a wide range of applications with excellent performances [25, 71]. These applications include pattern recognition, such as handwritten digit recognition [9, 26], face detection in images [62], and text classification [47].

The margin concept in SVM is also extended to the regression problem and an analogue of the margin is constructed in the space of the target values by using the ε-insensitive loss function in Support Vector Regression (SVR) to solve the regression task [88, 77, 71]. Better results are obtained in time series prediction [91, 58, 57], especially SVMs improved the best known result on the benchmark by 29%. SVMs have also been successfully applied in the financial related applications [81, 16, 17].

There are also some extensions of standard SVMs. For example, a weighted SVM was proposed in [62]. This extension is used to handle two frequent cases in classification and pattern recognition: (a) an unequal proportion of data samples between the classes; (b) a need to tilt the balance or weight one class versus the other, which is very frequent when a classification error of one type is more expensive or undesirable than other [62]. This extension is separated the cost of error C into C^+ and C^-, which will penalize with higher penalty the most undesirable type of error. One advantage of this extension has no real impact on the complexity of the problem of finding the optimal vector of Lagrange multipliers. This extension could be changed even further to allow, e. g. higher values of C for highly reliable or valuable data points and lower values for data points of less confidence or value [62]. A similar extension also appeared in the financial time series forecasting, e. g. [16]. In [16], the

authors assumed that in the non-stationary financial time series, the recent past data could provide more important information than the distant past data. Concretely, the authors used a sigmoid-like function to decrease the weight of C when the data is far from recent. Their experiments also confirmed their assumption of financial time series.

An important issue of making SVMs practically useful is automatic model selection. Most existing approaches use the leave-one-out (LOO) methods [20, 90, 50, 21]. This procedure of LOO consists of removing one element from the training data, constructing the decision rule on the basis of the remaining training data and then testing the removed data. This procedure can be done until all of the training data are tested. The procedure of LOO is usually used to estimate the probability of test error of a learning algorithm. Luntz and Brailovsky have proved a lemma [90]: The leave-one-out procedure gives an almost unbiased estimate of the probability of test error

$$Ep_{error}^{N-1} = E(\frac{\mathcal{L}(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N)}{N}),$$

where p_{error}^{N-1} is the probability of test error for the machine trained on a sample of size $N-1$, $\mathcal{L}(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_N, y_N)$ is the number of errors in the leave-one-out procedure.

The theoretical bounds of LOO in [90] are also applied in [20] to select the parameter (width) of RBF kernel. The relation between the LOO rate and the stopping criteria of the decomposition method for SVM is also studied in [50] and the authors found that using a very loose stopping criteria for the decomposition method, the best model can still be obtained. Such an observation leaded the authors to design a simple and practical automatic model selection software. Other methods to estimate the parameters of SVMs, e. g. C in [48].

Since the number of the dual variables in the QP problem is equal to the

number of data points, when the data set is large, the optimization problem becomes very challenging, because the quadratic form is complete dense and the memory requirements grow with the square of the number of data points [62]. To handle large data sets using SVM with non-linear kernels, the Reduced Support Vector Machines (RSVM) have been proposed as an alternate of the standard SVM by preselecting a subset of data as support vectors and solving a smaller optimization problem [51, 52]. In [51], the number of support vectors was restricted by solving the Reduced SVM (RSVM). Especially the kernel matrix is reduced from $N \times N$ to $N \times M$, where N is the number of data points and M is the size of a randomly selected subset of training data considered as candidates of support vectors. The performance (testing accuracy) is also as good as the regular SVM [51]. In [52], the authors showed that the RSVM formulation is already in a form of linear SVM and discussed four RSVM implementations. Their experiments indicated that generally the test accuracy of RSVM is a little lower than that of the standard SVM. In addition, for problems with up to tens of thousands of data, if the percentage of support vectors is not high, existing implementations for SVM is quite competitive on the training time. Therefore, RSVM will be mainly useful for either larger problems or those with many support vectors.

After describing the above famous models, we will turn to introduce the regression model, in particular the Support Vector Regression, in next chapter.

Chapter 3

Support Vector Regression

Now SVMs have been applied in the regression tasks [91, 29, 77, 69, 41, 70, 92, 37]. In this part, we will describe SVR beginning from the regression problem, then to the procedure of how to solve this problem.

3.1 Regression Problem

A regression problem is to estimate (learn) a function

$$f(\mathbf{x}, \boldsymbol{\lambda}) : \quad X(\mathbb{R}^d) \to \mathbb{R},$$

where X denotes the space of the input patterns, e.g., $\mathbb{R}^d$. It might be, for instance, stock prices for a company at subsequent days together with corresponding econometric indicators; $\boldsymbol{\lambda} \in \Lambda$, Λ is a set of abstract parameters, from a set of independent identically distributed (i.i.d.) samples with size N,

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N), \quad \mathbf{x}_i \in X(\mathbb{R}^d), \quad y_i \in \mathbb{R}, \tag{3.1}$$

where the above samples were drawn from an unknown distribution $P(\mathbf{x}, y)$.

Now the aim is to find a function $f(\mathbf{x}, \boldsymbol{\lambda}^*)$ with the smallest possible value for the *expected risk* (or test error) as

$$R[\boldsymbol{\lambda}] = \int l(y, f(\mathbf{x}, \boldsymbol{\lambda}))P(\mathbf{x}, y)d\mathbf{x}dy, \tag{3.2}$$

where l is a loss function which can be defined as one needs.

Usually the probability of distribution $P(\mathbf{x}, y)$ is unknown. Hence we are unable to compute, and to minimize, the expected risk $R[\boldsymbol{\lambda}]$ in Eq. (3.2). But we may know some information of $P(\mathbf{x}, y)$ from the samples of (3.1). So we compute a stochastic approximation of $R[\boldsymbol{\lambda}]$ by the so called *empirical risk*:

$$R_{emp}[\boldsymbol{\lambda}] = \frac{1}{N}\sum_{i=1}^{N} l(y_i, f(\mathbf{x}_i, \boldsymbol{\lambda})). \tag{3.3}$$

It is because that the law of large numbers guarantees that the empirical risk converges in probability to the expected risk. However, for practical problem, the size of samples is small. Only minimizing the empirical risk may cause problems, such as bad estimation or overfitting, and we cannot obtain good result when new data come in.

To solve the small sample problem, the statistical theory, or VC theory, has provided bounds on the deviation of the empirical risk from the expected risk [87, 89]. A typical uniform Vapnik and Chervonenkis bound, which holds with probability $1 - \eta$, has the following form:

$$R[\boldsymbol{\lambda}] \leq R_{emp}[\boldsymbol{\lambda}] + \sqrt{\frac{h(\ln\frac{2N}{h} + 1) - \ln\frac{\eta}{4}}{N}}, \quad \forall \boldsymbol{\lambda} \in \Lambda, \tag{3.4}$$

where h is the VC-dimension of $f(\mathbf{x}, \boldsymbol{\lambda})$.

From this bound, it is clear that in order to achieve small expected risk, i. e. good generalization performance, both the empirical risk and the ratio between the VC-dimension and the number of data points has to be small. Since the empirical risk is usually a decreasing function of h, it turns out that for a given number of samples, there is an optimal value of the VC-dimension. The choice of an appropriate value of h (which in most techniques is controlled by the number of free parameters of the model) is very important in order to get good performances, especially when the number of data points is small.

Therefore, a technique, *Structural Risk Minimization* (SRM), was developed by Vapnik [87, 88, 89] in the attempt to overcome the problem of choosing an appropriate VC-dimension. A different induction principle, *Structural Risk Minimization Principle*, was proposed in [87].

Support Vector Machines (SVMs) were developed to implement the SRM principle [88]. The SVMs were used in the classification at first; they were also applied in solving regression problem. When SVMs were used to solve the regression problem, they were usually called Support Vector Regression (SVR) and the aim of SVR is to find a function f with parameters $\mathbf{w}$ and b by minimizing the following regression risk:

$$R_{reg}(f) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} l(f(\mathbf{x}_i), y_i), \tag{3.5}$$

where C is a tradeoff term, called the cost of error; $\langle , \rangle$ denotes the inner product, the first term can be seen as the margin in SVMs and therefore, can measure the VC-dimension [88]. A common interpretation is that the Euclidean norm, $\langle \mathbf{w}, \mathbf{w} \rangle$, measures the flatness of the function f, minimizing $\langle \mathbf{w}, \mathbf{w} \rangle$ will make the objective function as flat as possible [77].

The function f is defined as,

$$f(\mathbf{x}, \mathbf{w}, b) = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle + b, \tag{3.6}$$

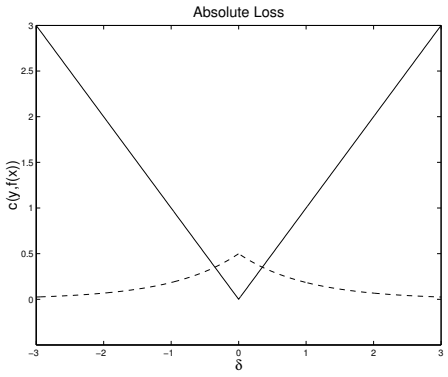
where $\boldsymbol{\phi}(\mathbf{x}) : \mathbf{x} \to \Omega$, maps $\mathbf{x} \in X(\mathbb{R}^d)$ into a high (possible infinite) dimensional space Ω, and $b \in \mathbb{R}$.

3.2 Loss Function

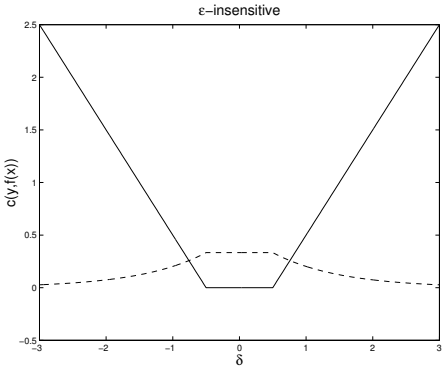
In order to measure the empirical risk, we should specify a loss function. There are many loss functions, such as, squared loss function, Huber's loss function, ε-insensitive loss function. Table 3.1 lists some common loss functions and their

Table 3.1: Loss functions and their corresponding density functions

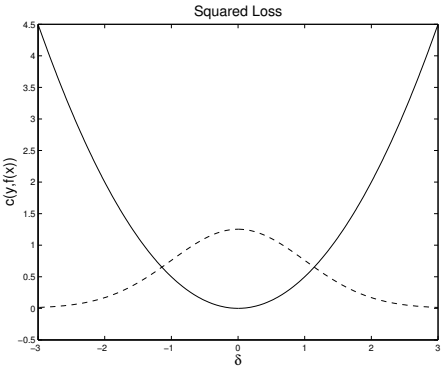
	loss function $l(\delta)$	density function $p(\delta)$
Linear ε-insensitive	$\lvert\delta\rvert_\varepsilon$	$\frac{1}{2(1+\varepsilon)}\exp(-\lvert\delta\rvert_\varepsilon)$
Laplacian	$\lvert\delta\rvert$	$\frac{1}{2}\exp(-\lvert\delta\rvert)$
Gaussian	$\frac{1}{2}\delta^2$	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{\delta^2}{2})$
Quadratic ε-insensitive	$\frac{1}{2}\delta_\varepsilon^2$	$\frac{1}{s^*}\exp(-\frac{1}{2}\delta_\varepsilon^2)$ $s^*=e^\varepsilon\sqrt{2\pi}-\sqrt{2\pi(e^{2\varepsilon}-1)}+2\sqrt{2\varepsilon}$
Huber's robust	$\begin{cases}\frac{1}{2\sigma}\delta^2, & \text{if } \lvert\delta\rvert\le\sigma\\ \lvert\delta\rvert-\frac{\sigma}{2}, & \text{otherwise}\end{cases}$	$\propto\begin{cases}\exp(-\frac{1}{2\sigma}\delta^2), & \text{if } \lvert\delta\rvert\le\sigma\\ \exp(\frac{\sigma}{2}-\lvert\delta\rvert), & \text{otherwise}\end{cases}$
Polynomial	$\frac{1}{d}\lvert\delta\rvert^d$	$\frac{d}{2\Gamma(1/d)}\exp(-\lvert\delta\rvert^d)$
Piecewise polynomial	$\begin{cases}\frac{1}{d\sigma^{d-1}}\delta^d, & \text{if } \lvert\delta\rvert\le\sigma\\ \lvert\delta\rvert-\sigma\frac{d-1}{d}, & \text{otherwise}\end{cases}$	$\propto\begin{cases}\exp(-\frac{\lvert\delta\rvert^d}{d\sigma^{d-1}}), & \text{if } \lvert\delta\rvert\le\sigma\\ \exp(\sigma\frac{d-1}{d}-\lvert\delta\rvert), & \text{otherwise}\end{cases}$



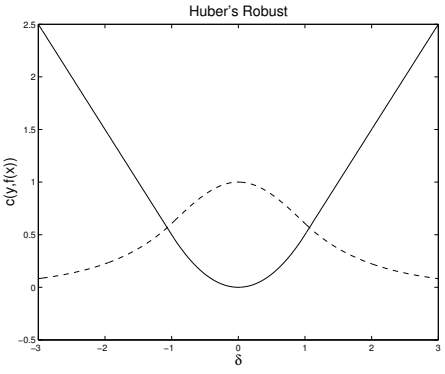
(a) Laplacian



(b) ε-insensitive



(c) Gaussian



(d) Huber's Robust

Figure 3.1: Typical loss functions with their density functions.

density functions, Quadratic ε-insensitive loss function is added comparing to the functions listed in [73].

A statistical perspective is given in [73]. It assumes that the target values y are generated by an underlying functional dependency f plus additive noise δ with density p_δ, i. e. $f(\mathbf{x}_i) = f_{true} + \delta_i$. Therefore, minimizing R_{emp} coincides to choose loss function equal minus log-density function, i. e. $l(f(\mathbf{x}), y) = -\log p(y|\mathbf{x}, f)$.

Therefore, the most frequently used loss function, square loss function (see Fig. 3.1(c)), corresponds to that the observation y is corrupted by normal noise.

$$l_2(y, f(\mathbf{x})) = \frac{1}{2}(y - f(\mathbf{x}))^2 \quad \text{or} \quad l_2(\delta) = \frac{1}{2}\delta^2, \tag{3.7}$$

the squared loss corresponds to the assumption of Gaussian noise. However, the squared loss is not always the best choice. There are still many loss functions can be chosen for different problems.

In [88], the ε-insensitive loss function (Fig. 3.1(b)) is proposed,

$$l_\varepsilon(y, f(\mathbf{x})) = \begin{cases} 0, & \text{if} \quad |y - f(\mathbf{x})| < \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}. \tag{3.8}$$

The difference between ε-insensitive loss function and Laplacian function (see Fig. 3.1(a)) is that in ε-insensitive loss function, when the data points are in the range of $\pm\varepsilon$, they do not contribute to the output error. Therefore, increasing the ε value, one reduces the number of support vectors, extremely one may obtain a constant regression function. This indirectly affects the complexity and generalization of models.

Due to the advantage of ε-insensitive loss function, we just consider it as loss function here, the corresponding SVR is also called ε-SVR. The minimization of Eq. (3.5) is equivalent to the following constrained minimization problem:

$$\min \quad \Upsilon(\mathbf{w}, b, \boldsymbol{\xi}^{(*)}) = \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} (\xi_i + \xi_i^*), \tag{3.9}$$

$$\begin{aligned} \text{subject to} \quad y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x_i}) \rangle + b) &\leq \varepsilon + \xi_i, \\ \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x_i}) \rangle + b) - y_i &\leq \varepsilon + \xi_i^*, \\ \xi_i^{(*)} &\geq 0. \end{aligned} \tag{3.10}$$

Here and below, it is understood that $i = 1, \ldots, N$ and $(*)$ is a shorthand implying both the variables with and without asterisks. ξ_i and ξ_i^* measures the up error and down error for sample $(\mathbf{x_i}, y_i)$ respectively, see Fig. 3.2(a).

A standard method to find the optimal solution of above minimization problem Eq. (3.9), therefore find the function f as Eq. (3.6), is to construct the dual problem of this optimization problem (primal problem) by Lagrange Method and to translate the (primal) minimization problem to maximize its dual function, basic results in Appendix. A. Therefore, a Quadratic Programming (QP) problem is obtained [88]:

$$\begin{aligned} \min \quad Q(\boldsymbol{\alpha}^{(*)}) &= \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j) \rangle \\ &+ \sum_{i=1}^{N} (\varepsilon - y_i)\alpha_i + \sum_{i=1}^{N} (\varepsilon + y_i)\alpha_i^*, \end{aligned} \tag{3.11}$$

subject to

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0, \quad \alpha_i^{(*)} \in [0, C]. \tag{3.12}$$

After solving this QP problem, we obtain the objective function

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle + b,$$

where α, α^* are the Lagrange multipliers used to pull and push f towards to the observation y.

Although the QP problem is solved, b is not calculated. The computation of b is exploited by the Karush-Kuhn-Tucker (KKT) conditions. Here, they are

$$\begin{aligned} \alpha_i(\varepsilon + \xi_i - y_i + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b) &= 0, \\ \alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - b) &= 0, \end{aligned}$$

and

$$\begin{aligned} (C - \alpha_i)\xi_i &= 0, \\ (C - \alpha_i^*)\xi_i^* &= 0. \end{aligned}$$

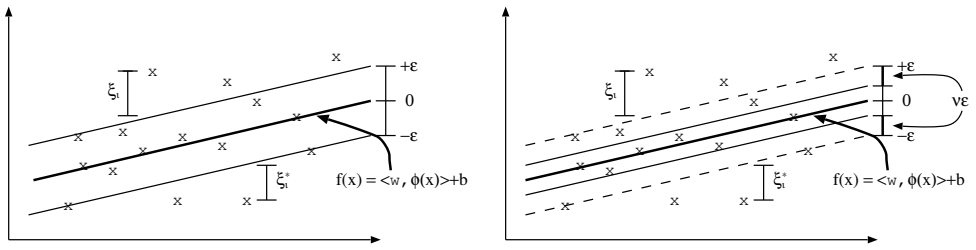
Therefore, several useful conclusions are obtained. Firstly, $\alpha_i^{(*)} = C$ means that samples $(\mathbf{x}_i, y_i)$ lie outside of the ε margin. Secondly, $\alpha_i\alpha_i^* = 0$, this means that any pair of dual variables α_i, α_i^* are not nonzero simultaneously, otherwise, it will require nonzero slack in both directions. Finally $\alpha_i^{(*)} \in (0, C)$ corresponds to samples $(\mathbf{x}_i, y_i)$ lying on the ε margin and b can be computed as follows:

$$b = \begin{cases} y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - \varepsilon, & \text{for } \alpha_i \in (0, C) \\ y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + \varepsilon, & \text{for } \alpha_i^* \in (0, C) \end{cases}.$$

When no $\alpha_i^{(*)} \in (0, C)$, such method as in [19] are used.

Usually, those sample points $(\mathbf{x}_i, y_i)$ with nonzero α_i or α_i^* are called *support vectors*.

The parameter ε is usually difficult to control [58], as one does not know beforehand how accurately one is able to fit the curve. A partial solution is using the ν-SVR, which is a modification of the ε-SVR. In ν-SVR, it introduces a new parameter ν (see Fig. 3.2(b)) to replace ε and this ν controls the fraction of examples outside of the ε-tube and indirectly controls the size of the ε-tube [69, 74, 70, 27, 63].



(a) ε-SVR (b) ν-SVR

Figure 3.2: Linear regression on the feature space by ε-SVR and ν-SVR.

3.3 Kernel Function

To solve the non-linear samples, SVR exploits the mapping function $\boldsymbol{\phi}$, this function maps the input space X into a new space $\Omega = \{\boldsymbol{\phi}(\mathbf{x}) \mid \mathbf{x} \in X\}$, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ becomes to $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))$ and a linear regression function is obtained in the feature space (Ω), see Fig. 3.2(a).

In Eq. (3.11), the maximizing objective function contains an inner product of mapping function. Here we can see another advantage of SVR. By using the trick of kernel function, one lets the kernel function be the inner product of mapping function, $K(\mathbf{x}, \mathbf{z}) = \langle \boldsymbol{\phi}(\mathbf{x}), \boldsymbol{\phi}(\mathbf{z}) \rangle$. Therefore, one only needs to specify a kernel function without considering the mapping function or the feature space explicitly.

The name *kernel* is derived from integral operator theory, which supports much of the theory of the relation between kernels and their corresponding feature spaces. An important consequence of the dual representation is that the dimension of the feature space need not affect the computation. As one does not represent the feature vectors explicitly, the number of operations required to compute the inner product by evaluating the kernel function is not necessarily proportional to the number of features. The use of kernel makes it possible to map the data implicitly into a feature space and to train a linear machine in such a space, potentially side-stepping the computational

problems inherent in evaluating the feature map. The only information used about the training examples is that Gram matrix, or kernel matrix, in the feature space [28].

Kernel function should satisfy the Mercer's Theorem. From this theorem, a mapping function $\boldsymbol{\phi}(\mathbf{x})$ for a kernel matrix $\mathbf{K}$ can be constructed as follows [70, 28],

$$\boldsymbol{\phi} : \mathbf{x}_i \longmapsto (\sqrt{\lambda_t}\upsilon_{ti})_{t=1}^{N} \in \mathbb{R}^N.$$

where λ_t is the eigenvalues of $\mathbf{K}$, $\boldsymbol{\upsilon}_t = (\upsilon_{ti})_{i=1}^{N}$ is the corresponding eigenvector.

$$\langle \boldsymbol{\phi}(\mathbf{x}_k), \boldsymbol{\phi}(\mathbf{x}_l) \rangle = \sum_{t=1}^{N} \lambda_t \upsilon_{tk} \upsilon_{tl} = (\mathbf{V}^T \Lambda \mathbf{V})_{kl} = \mathbf{K}_{kl} = K(\mathbf{x}_k, \mathbf{x}_l),$$

However, for the same kernel function, the mapping function is not unique. For example, we may choose a kernel function as $K(\mathbf{x}_k, \mathbf{x}_l) = \langle \mathbf{x}_k, \mathbf{x}_l \rangle^2$, where $\mathbf{x} \in \mathbb{R}^2$. It is easy to see that $\boldsymbol{\phi}_1(\mathbf{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)'$, $\boldsymbol{\phi}_2(\mathbf{x}) = \frac{1}{\sqrt{2}}(x_1^2 - x_2^2, 2x_1x_2, x_1^2 + x_2^2)'$, $\boldsymbol{\phi}_3(\mathbf{x}) = (x_1^2, x_1x_2, x_1x_2, x_2^2)'$ all are can be the mapping function of this kernel function K [12].

Four common kernel functions include:

Linear function: $K(\mathbf{x}_k, \mathbf{x}_l) = \langle \mathbf{x}_k, \mathbf{x}_l \rangle$;

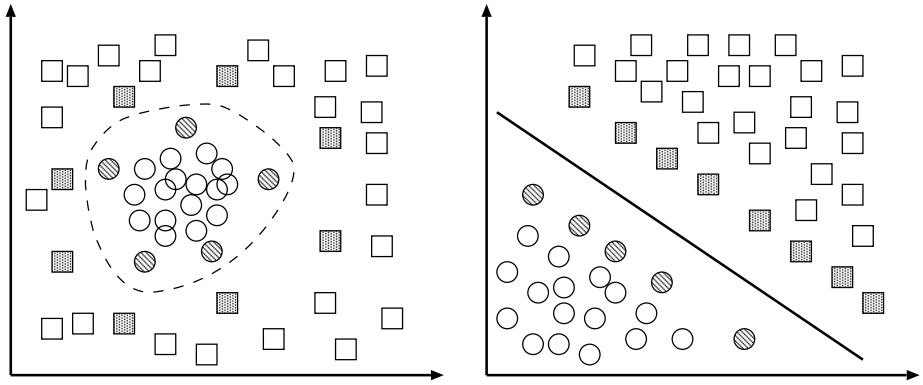
Polynomial function with parameter d, $K(\mathbf{x}_k, \mathbf{x}_l) = (\langle \mathbf{x}_k, \mathbf{x}_l \rangle + 1)^d$;

Radial Basis Function (RBF) with parameter β:

$$K(\mathbf{x}_k, \mathbf{x}_l) = \exp(-\beta \|\mathbf{x}_k - \mathbf{x}_l\|^2), \tag{3.13}$$

a demonstration for separable classes by RBF kernel function is illustrated in Fig. 3.3(a), a mapping to feature space is depicted in Fig. 3.3(b).

Hyperbolic tangent: $K(\mathbf{x}_k, \mathbf{x}_l) = \tanh(2\langle \mathbf{x}_k, \mathbf{x}_l \rangle + 1)$.



(a) Radial Basis Function (b) RBF mapping

Figure 3.3: Separable classification with Radial Basis kernel functions in different space. Left: original space. Right: feature space.

3.4 Relation to Other Models

3.4.1 Relation to Support Vector Classification

A SVM for (separable) classification (SVC) is to construct a hyperplane in the same form of Eq. (3.6) from data

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N), \quad \mathbf{x}_i \in \mathbb{R}^d, \quad y_i \in \{\pm 1\}, \tag{3.14}$$

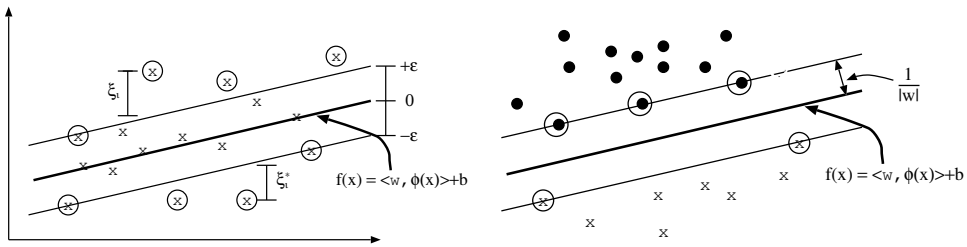
by solving the following minimization problem [9]:

$$\min \quad \langle \mathbf{w}, \mathbf{w} \rangle \quad \textit{subject to} \quad y_i \cdot f(\mathbf{x}_i) \geq 1, \tag{3.15}$$

where $\frac{1}{\langle \mathbf{w}, \mathbf{w} \rangle}$ means the width of margin, minimizing $\langle \mathbf{w}, \mathbf{w} \rangle$ is equivalent to maximizing the margin width between two classes (here the margin width is defined by the distance of the hyperplane to the nearest of either class). After that, the decision function takes the form

$$f(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}) \rangle + b).$$

At first, we can see that the regression problem, Eq. (3.9) with constraints Eq. (3.10), is so different to the classification problem, Eq. (3.15). The first



(a) SVR (b) SVC

Figure 3.4: Demonstration for regression and classification on the feature space. Sample points with circles are support vectors. (a) support vectors lie on or out of the margin bound, sample points inside margin bound have no contribution to the decision function; (b) support vectors lie on the margin bound, sample points outside margin bound have no contribution to the decision function.

difference is their constraints: SVR is additive, SVC is multiplicative. The second difference is the support vectors: in SVR, support vectors lie on or out of the margin bound, see Fig. 3.4(a); in SVC, support vectors lie on the margin bound (Fig. 3.4(b)). The third difference is that other points in SVR are required to lie within a margin bound of radius ε, where in SVC, they are required to lie outside of margin bound and on the correct side (Fig. 3.4). These points (both for regression and classification) do not contribute to the decision function.

Although the margin concept is different, a connection of margins in regression and classification is given in [74] by the following ε-margin definition.

Definition 1 (ε-margin) Let $(E, \|\cdot\|_E)$,$(F, \|\cdot\|_F)$ be normed space, and $X \subset E$. The ε-margin of a function $f : X \longmapsto F$ is defined as

$$m_\varepsilon(f) := \inf\{\|\mathbf{x} - \mathbf{y}\|_E \mid \mathbf{x}, \mathbf{y} \in X, \|f(\mathbf{x}) - f(\mathbf{y})\|_F \geq 2\varepsilon\}.$$

Therefore, for a linear function $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$, the ε-margin takes the form $m_\varepsilon(f) = \frac{2\varepsilon}{\|\mathbf{w}\|}$, detailed description see Example 9 in [74]. Hence, for fixed ε, maximizing the margin amounts to minimizing $\|\mathbf{w}\|$ as done in the SV

regression: in the simplest form, *cf.* Eq. (3.9) without slack variable ξ_i, the training on data Eq. (3.1) consists of minimizing $\|\mathbf{w}\|^2$ subject to

$$|f(\mathbf{x}_i) - y_i| \leq \varepsilon. \tag{3.16}$$

Therefore, minimizing $\|\mathbf{w}\|^2$ means to find a function f as flat as possible [77].

For classification, the margin can be set to $m_1(f) = \frac{2}{\|\mathbf{w}\|}$, which is equal to the margin defined for Vapnik's canonical hyperplane [88]. Given the data set as Eq. (3.14), an oriented hyperplane in E can be uniquely expressed by a linear function as Eq. (3.6) with

$$\min \ \{|f(\mathbf{x})| \mid \mathbf{x} \in X\} = 1. \tag{3.17}$$

From Eq. (3.15), the parameter ε is superfluous. However, the decision function, $f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b)$, will not change if minimizing $\langle \mathbf{w}, \mathbf{w} \rangle = \|\mathbf{w}\|^2$, *subject to* $y_i \cdot f(\mathbf{x}_i) \geq \varepsilon$. For the points on the margin bound, Fig. 3.5, there are $1 = y_i \cdot f(\mathbf{x}_i) = 1 - |f(\mathbf{x}_i) - y_i|$.

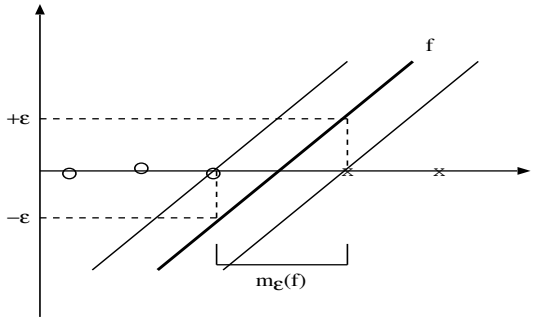


Figure 3.5: 1-D toy example: separate 'o' from 'x'. The SV classification algorithm constructs a linear function $f(x) = \langle \mathbf{w}, \mathbf{x} \rangle + b$ satisfying Eq. (3.17) with $\varepsilon = 1$. To maximize the margin $m_\varepsilon(f)$, one has to minimize $\|\mathbf{w}\|$.

3.4.2 Relation to Ridge Regression

Ridge regression is originated as a linear regression [44], it chooses a function that minimizes a combination of square loss and norm of the $\mathbf{w}$ vector, which is

analogous to the maximal margin hyperplane in the SVMs. The original motivation of Ridge regression is based on statistical and numerical consideration. A ridge regression algorithm minimizes the penalized loss function

$$S(\mathbf{w}, b) = \lambda\langle \mathbf{w}, \mathbf{w}\rangle + \sum_{i=1}^{N}(\langle \mathbf{w}, \mathbf{x}_i\rangle + b - y_i)^2,$$

where the parameter λ controls the trade-off between low sum square loss and low norm of the solution (analogous to C in SVMs).

Using matrix notation and adding b^2 in the first term of above equation, S can be represented as,

$$S(\tilde{\mathbf{w}}) = \lambda(\mathbf{I}_{N+1}\tilde{\mathbf{w}})^T(\mathbf{I}_{N+1}\tilde{\mathbf{w}}) + (\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}})^T(\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}),$$

where $\mathbf{I}_n$ is an $n \times n$ identity matrix, $\tilde{\mathbf{w}} = (\mathbf{w}^T, b)^T$, $\tilde{\mathbf{X}} = (\mathbf{X}', \mathbf{1}_N)$, superscript T denotes the transpose.

In order to get the optimal solution, we let $\frac{\partial S}{\partial \tilde{\mathbf{w}}} = 0$, i. e.,

$$\frac{\partial S}{\partial \tilde{\mathbf{w}}} = 2\lambda \mathbf{I}_N\tilde{\mathbf{w}} + 2(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \tilde{\mathbf{X}}^T\mathbf{y}) = 0,$$

hence

$$(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}_{N+1})\tilde{\mathbf{w}} = \tilde{\mathbf{X}}^T\mathbf{y},$$

$$\tilde{\mathbf{w}} = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}} + \lambda\mathbf{I}_{N+1})^{-1}\tilde{\mathbf{X}}^T\mathbf{y},$$

where $(\cdot)^{-1}$ is the inverse of matrix.

Similarly, using the trick of kernel function, $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is just a kernel matrix with linear kernel function. Other kernel function with corresponding kernel matrix can be constructed to replace the $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ by $\mathbf{K}$, and we can obtain the corresponding optimal solution for nonlinear regression by this kernel $\mathbf{K}$,

$$\tilde{\mathbf{w}} = (\mathbf{K} + \lambda\mathbf{I}_{N+1})^{-1}\tilde{\mathbf{X}}^T\mathbf{y}.$$

We can see that this is exactly the Least-Squares Support Vector Machines (LS-SVMs). LS-SVMs are another class of learning machines using the name

of SVM [79, 80]. However, these models are different to SVM formulations. In these models, the quadratic loss function are considered. The inequality constraints are replaced to be equality. The dual problem becomes to solve a set of linear system.

3.4.3 Relation to Radial Basis Function

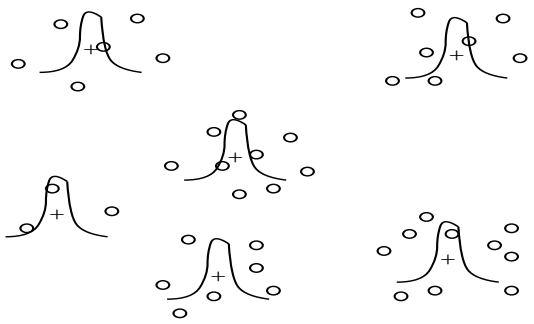


Figure 3.6: A demonstration of standard RBF network for regression, marked '+' denote the center of RBF nodes.

An SVR with RBF kernels (Eq. (3.13)) results an architecture of an RBF network. However, there are some differences between SVR and RBF network. In a standard RBF network, the number of nodes and their centers is determined by k-means clustering (Fig. 3.6). In contrast, an SVR with RBF kernels uses RBF nodes centered on the support vectors. The number of nodes equals to the number of support vectors and the centers of the RBF nodes identify with the support vectors themselves (Fig. 3.4(a)). The RBF function in both models provides same action, adjusting the distance of a point to the centers of RBF nodes [14].

3.5 Implemented Algorithms

In practice, SVMs need to solve a QP problem. The SVM algorithm for solving this QP problem is complex, subtle, and difficult for an average engineer

to implement. Hence, in the beginning, researchers have to use QP optimal packages, such as MINOS, LOQO. These packages are some quadratic program subroutines provided in the Matlab optimization toolbox, but they are usually commercial. In the sequel, there are a large number of SV algorithms have been proposed over the years: the Newton method, gradient descent method [15], primal dual interior-point method [77], subset selection algorithms such as chunking (introduced by Vapnik, 1982 [87]), Sequential Minimal Optimization (SMO), proposed by Platt [64], etc [61, 36, 78]. Now these are also some packages implemented by the above algorithms available on the internet. For example, the package SVM^{light} of Joachims [46], the libSVM, which is prepared by Chih-Jen Lin [19], the Matlab SVM Toolbox, by Steve Gunn [41]. Here, we will briefly review some of the most common algorithms.

Gradient Descent

Gradient descent is the simplest method for solving optimization problem, it is also known as the steepest descent algorithm [5, 15, 7]. The algorithm begins with an initial estimate for the solution, and then iteratively updates the vector following the steepest descent path. At each iteration the direction of the update is determined by the steepest descent strategy while the length of the step may be fixed, which is also known as learning rate [28].

Therefore, there is another way to see the QP algorithm, i. e., the quantity $Q(\boldsymbol{\alpha})$ in Eq. (3.11) is iteratively decreased by fixing all variables but one. Hence a multi-dimensional problem is reduced to a sequence of one dimensional ones. For QP problems there is a global maximum solution, and this global optimal solution can be found only by choosing a suitable learning rate. From the point of view of speed, such a strategy is not usually obtained an optimal solution, but it is good for those data set with thousand of points and it is very easy to implementation.

Using this algorithm, the i-th component of the gradient of $Q(\boldsymbol{\alpha})$ in Eq. (3.11)

is

$$\frac{\partial Q(\boldsymbol{\alpha})}{\partial \alpha_i} = \mathbf{Q}\boldsymbol{\alpha} + \mathbf{p},$$

where $\mathbf{Q} = \begin{bmatrix} \mathbf{K} & -\mathbf{K} \\ -\mathbf{K} & \mathbf{K} \end{bmatrix}$, $\mathbf{K}$ is the kernel matrix. Thus, $Q(\boldsymbol{\alpha})$ can be minimized simply by iterating the update rule,

$$\alpha_i \leftarrow \alpha_i - r_i \frac{\partial Q(\boldsymbol{\alpha})}{\partial \alpha_i}$$

with a suitable different learning rate r_i for different α_i. After each iteration, α_i should still keep the constraints $0 \leq \alpha_i \leq C$. This can be completed by resetting α_i to zero if α_i becomes negative and forcing α_i to C when $\alpha_i > C$.

The uniqueness of the global maximum guarantees that for suitable choices of r_i of algorithm will always find the solution. Such a strategy is usually not optimal from the point of view of speed, but is surprisingly good for datasets of up to a couple of thousand points and has the advantage that its implementation is very straightforward [28]. However, gradient method has lack to solve high dimensional data, especially the dimension is up to 300.

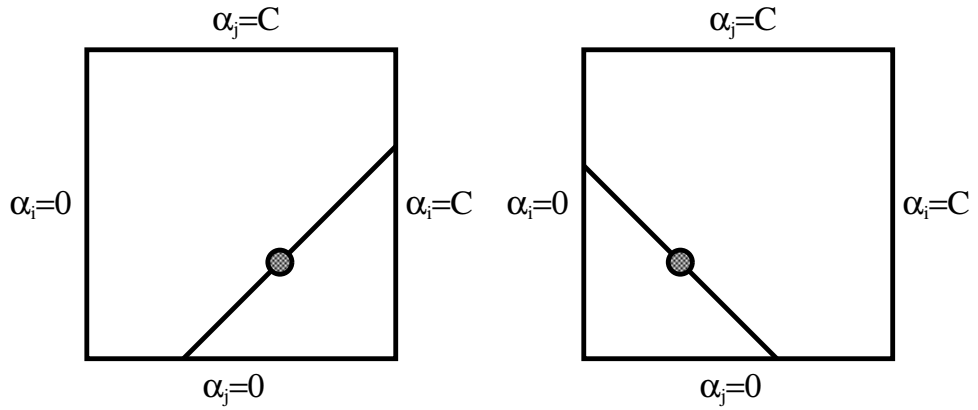
Sequential Minimal Optimization (SMO)

The Sequential Minimal Optimization(SMO) algorithm is devised by Platt [64], which is the simplest decomposition method and optimizes a minimal subset of just two points at each iteration. The advantage of this technique lies in the fact that the optimization problem for two data points admits an analytical solution, eliminating the need to use an iterative quadratic programme optimizer as part of the algorithm.

The idea of SMO is to solve the smallest possible optimization problem at each step for the standard SVM QP problem. To keep the constraint $\mathbf{y}^T\boldsymbol{\alpha} = 0$ always true, a subset with just two points at each iteration is needed, which is the smallest possible optimization problem (Fig. 3.7). This also implies that

when a Lagrange multiplier is updated, at least one other multiplier needs to be adjusted in order to keep the constraint true.

The procedure of SMO is that at each iteration, SMO chooses two elements α_i and α_j with others are fixed, then updates α_i, α_j with analytical expressions accordingly. The choice of these two points is determined by an heuristic algorithm, while the optimization of the two multipliers is performed analytically.



(a) Case I: $y_i \neq y_j$ induces $\alpha_i - \alpha_j = k$ (b) Case II: $y_i = y_j$ induces $\alpha_i + \alpha_j = k$

Figure 3.7: The selected Lagrange multipliers must satisfy all of the constraints of the QP problem. To meet the inequality constraints, the Lagrange multipliers must lie in the box; To satisfy the linear equality, the Lagrange multipliers must lie on a diagonal line. Hence, one step of SMO must find an optimum of the objective function on a diagonal line segment.

In libSVM [19], the authors have implemented the SVM using this algorithm. They select the index i and j by the following criteria.

$$i \equiv \text{ argmax } (-\nabla Q(\boldsymbol{\alpha})_l \mid y_l = 1, \alpha_l < C, \nabla Q(\boldsymbol{\alpha})_l \mid y_l = -1, \alpha_l > 0), \quad (3.18)$$

$$j \equiv \text{ argmin } (\nabla Q(\boldsymbol{\alpha})_l \mid y_l = -1, \alpha_l < C, -\nabla Q(\boldsymbol{\alpha})_l \mid y_l = 1, \alpha_l > 0),$$

where $Q(\boldsymbol{\alpha})$ is the same in Eq. (3.11). α_i and α_j are two elements that violate the following KKT conditions the most,

$$\mathbf{W}\boldsymbol{\alpha} + \mathbf{p} + b\mathbf{y} = \boldsymbol{\mu} - \boldsymbol{\nu},$$

$$\mu_i\alpha_i = 0, \nu_i(C-\alpha)_i = 0, \mu_i \geq 0, \nu_i \geq 0.$$

After selecting these two multipliers, they begin to update the multipliers. Since the constraints in Eq. (3.12), $\mathbf{y}^T\boldsymbol{\alpha} = 0$, can not be violated, the new values of the multipliers must lie on a diagonal line Fig. 3.7(a) and Fig. 3.7(b),

$$\alpha_i y_i + \alpha_j y_j = \text{constant} = \alpha_i^{old} y_i + \alpha_j^{old} y_j.$$

Without considering the constraints, α_j can be computed as,

$$\alpha_j^{new} = \begin{cases} \alpha_j + \frac{-G_i - G_j}{Q_{ij}+Q_{jj}+2Q_{ij}}, & \text{if} \quad y_i \neq y_j \\ \alpha_j + \frac{G_i - G_j}{Q_{ij}+Q_{jj}-2Q_{ij}}, & \text{if} \quad y_i = y_j \end{cases}, \tag{3.19}$$

where $G_i \equiv \nabla W(\alpha)_i$ and $G_j \equiv \nabla W(\alpha)_j$.

However, there is still a box constraint $0 \leq \alpha_i, \alpha_j \leq C$ need to satisfy. In SMO, it is to force α_j to satisfy a new constraint, $L \leq \alpha_j^{new} \leq H$, by a clipping procedure,

$$\alpha_j^{new,clipped} = \begin{cases} H, & \text{if} \quad \alpha_j^{new} \geq H \\ \alpha_j^{new}, & \text{if} \quad L < \alpha_j^{new} < H \\ L, & \text{if} \quad \alpha_j^{new} \leq L \end{cases}, \tag{3.20}$$

where if $y_i \neq y_j$,

$$L = \max(0, \alpha_j^{old} - \alpha_i^{old}), \qquad H = \min(C, C + \alpha_j^{old} - \alpha_i^{old}),$$

and if $y_i = y_j$

$$L = \max(0, \alpha_j^{old} + \alpha_i^{old} - C), \qquad H = \min(C, \alpha_j^{old} + \alpha_i^{old}),$$

and the value of α_i is obtained from $\alpha_j^{new,clipped}$ as follows,

$$\alpha_i^{new} = \alpha_i^{old} + y_i y_j (\alpha_j^{old} - \alpha_j^{new,clipped}). \tag{3.21}$$

In summary, the procedure of SMO algorithm is to find the index i, j by Eq. (3.18), then to update α_j by Eq. (3.19) and to clip α_j by Eq. (3.20) to satisfy the box constraint, next is to update α_i by Eq. (3.21), these procedure are done until stop criteria is met.

Comparing SMO with other algorithms, SMO has following advantages:

1. SMO breaks the QP problem into a series of smallest possible QP problems, which only includes two variables;

2. SMO can solve the small QP problem analytically, which avoids using a time-consuming numerical QP optimization as an inner loop.

3. SMO reduces the needed memory largely; the amount of memory required for SMO is linear in the training set size.

Chapter 4

Margins in Support Vector Regression

It is well known that a model can be constructed to fit a fixed data set (i. e. the "in sample" or "training set" data) arbitrarily well, but that does not necessarily imply that the model will describe new data (i. e. the "out of sample" or "testing set" data) from that domain equally well. From Chapter 3, we know the Statistical Learning (VC) theory provides a upper bound, Eq. (3.4), on the test error. This upper bound depends on both the empirical risk and the capacity of the function class. Minimization of this upper bound leads to the principle of *Structural Risk Minimization* (SRM). SVMs are a kind of models implementing the SRM principle.

Due to its theoretical ground, SVM has been applied successfully in many applications, such as pattern recognition [26], text categorization [47], classification task as OCR [88], and time series prediction [58, 57]. Especially, it succeeded in financial applications, e. g. bankruptcy prediction [35, 99], time series forecasting [82, 16, 17].

When SVR is applied in time series forecasting, the ε-insensitive loss function is usually used to measure the empirical risk. In the following, we will indicate why ε-insensitive loss function is used and what is the problem about it. In addition, we will describe how to measure the experimental accuracy in

the whole thesis.

4.1 Problem

In this thesis, we use ε-insensitive loss function as the loss function. The value 2ε is called as the width of margin here. This loss function not only measures the training error (empirical risk), but also controls the sparsity of the solution (the number of support vectors). When the ε-margin width is increased, it may tend to reduce the number of support vectors [88]. Extremely, a too wide margin may result in a constant objective function. The setting of ε-margin width affects the complexity and the generalization of the objective function indirectly.

Therefore, the setting of ε value is very important. Usually, there are four methods to deal with it. Firstly, most practitioners set the value of ε as a non-negative constant value just for convenience. For example, in [84], they simply set the margin width to 0. This amounts to the least modulus loss function. In other instances the margin width has been set to a very small value [91, 57, 19]. The second method is the cross-validation technique, e. g. [58, 16]. It is usually too expensive in terms of computation. A more efficient approach is to use another variant called ν-SVR [69, 74, 70, 63], which determines ε by using another parameter ν. It is stated that ν may be easier to specify than ε. Another approach by Smola *et al.* [76] is to find the "optimal" choice of ε based on maximizing the statistical efficiency of a location parameter estimator. They showed that the asymptotically optimal ε should scale linearly with the input noise of the training the data, and this was verified experimentally. However, their predicted value of the optimal ε does not have a close match with their experimental results. Due to the special characteristics of financial time series, we will use different methods to set the ε value.

4.2 General ε-insensitive Loss Function

At first, we note that the margin in ε-insensitive loss function contains two characteristics: fixed and symmetrical. Based on these two characteristics, we have proposed a general ε-insensitive loss function and classified the margin into four case in [96]: Fixed and Symmetrical Margin (FASM), Fixed and Asymmetrical Margin (FAAM), Non-fixed and Symmetrical Margin (NASM) and Non-fixed and Asymmetrical Margin (NAAM). Table 4.1 gives a simple description of the classification. FASM is equivalent to the margin in ε-insensitive loss function, see Fig. 4.1(a). FAAM is divided into up margin and down margin, each margin is fixed but they are not equal (Fig. 4.1(b)). While NASM is with equal up margin and down margin, but they are varied with data (Fig. 4.1(c)). NAAM combines two characteristics of the margin (Fig. 4.1(d)).

Table 4.1: Margin categories

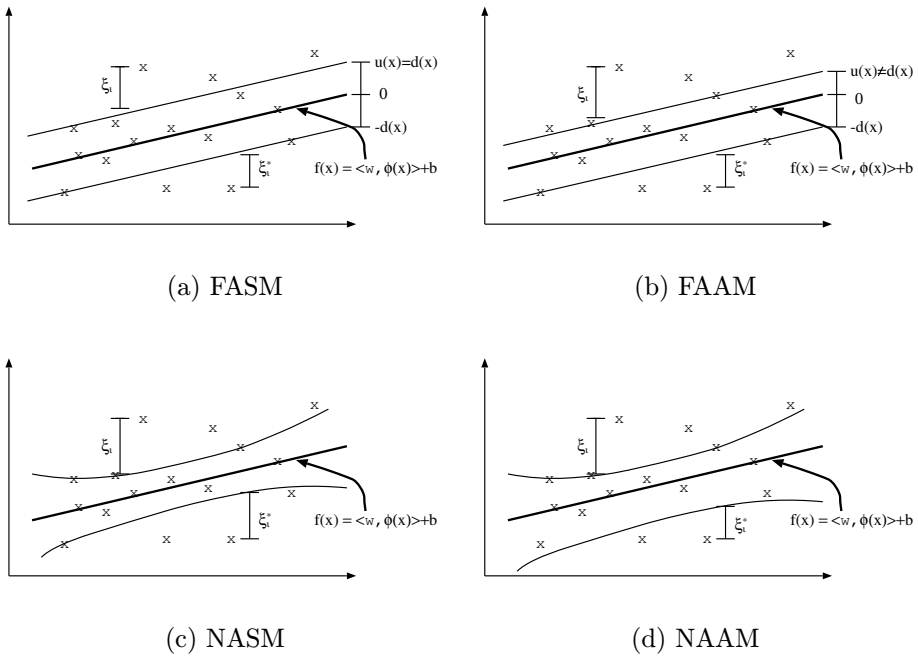
	Symmetrical	Asymmetrical
Fixed	FASM	FAAM
Non-fixed	NASM	NAAM

In the following, we will derive the SV formula based on the general ε-insensitive loss function.

The general ε-insensitive loss function splits the margin in the original ε-insensitive loss function into two parts: up margin and down margin,

$$l'_\varepsilon(f(\mathbf{x}_i) - y_i) = \begin{cases} 0, & \text{if } -d(\mathbf{x}_i) < y_i - f(\mathbf{x}_i) < u(\mathbf{x}_i) \\ y_i - f(\mathbf{x}_i) - u(\mathbf{x}_i), & \text{if } y_i - f(\mathbf{x}_i) \geq u(\mathbf{x}_i) \\ f(\mathbf{x}_i) - y_i - d(\mathbf{x}_i), & \text{if } f(\mathbf{x}_i) - y_i \geq d(\mathbf{x}_i) \end{cases}, \tag{4.1}$$

where $d(\mathbf{x}_i), u(\mathbf{x}_i) \geq 0$,, are two functions to determine the down margin and up margin at point $\mathbf{x}_i$ respectively. Again, here and below $i = 1, \ldots, N$. When



(a) FASM (b) FAAM

(c) NASM (d) NAAM

Figure 4.1: Four categories in general ε-insensitive loss function of SVR.

$d(\mathbf{x})$ and $u(\mathbf{x})$ are both constant functions and $d(\mathbf{x}) = u(\mathbf{x})$, Eq. (4.1) amounts to the ε-insensitive loss function in Eq. (3.8) and we labeled it as FASM (Fixed and Symmetrical Margin). When $d(\mathbf{x})$ and $u(\mathbf{x})$ are both constant functions but $d(\mathbf{x}) \neq u(\mathbf{x})$, this case is labeled as FAAM (Fixed and Asymmetrical Margin). In the case of NASM (Non-fixed and Symmetrical Margin), $d(\mathbf{x}) = u(\mathbf{x})$ but are varied with the data. The last case is with a non-fixed and asymmetrical margin(NAAM) where $d(\mathbf{x})$ and $u(\mathbf{x})$ are varied with the data and $d(\mathbf{x}) \neq u(\mathbf{x})$.

In the same way, we use the standard method to find the solution of Eq. (3.5) with the cost function of Eq. (4.1) as [88] and obtain

$$\min_{\mathbf{w},b,\boldsymbol{\xi}^{(*)}} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle + C \sum_{i=1}^{N} (\xi_i + \xi_i^*), \tag{4.2}$$

subject to

$$\begin{aligned} y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - b &\leq u(x_i) + \xi_i, \\ \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b - y_i &\leq d(x_i) + \xi_i^*, \\ \xi_i^{(*)} &\geq 0. \end{aligned}$$

Here $(*)$ has the same meaning as before, i. e, they are two kinds of variables, one with asterisks, another without asterisks.

Similar to the standard method in Appendix A.2, we construct the Lagrange function as

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi^*}) &= \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\mu_i\xi_i + \mu_i^*\xi_i^*) \\ &- \sum_{i=1}^{N}\alpha_i(u(x_i) + \xi_i - y_i + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b) \\ &- \sum_{i=1}^{N}\alpha_i^*(d(x_i) + \xi_i^* + y_i - \langle \mathbf{w}\cdot\boldsymbol{\phi}(\mathbf{x}_i)\rangle - b). \end{aligned} \tag{4.3}$$

At the saddle point of this Lagrange function, we have

$$\begin{aligned} \tfrac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i) &&= 0, \\ \tfrac{\partial L}{\partial b} &= \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i^* &&= 0, \\ \tfrac{\partial L}{\partial \xi_i} &= C - \alpha_i - \mu_i &&= 0, \\ \tfrac{\partial L}{\partial \xi_i^*} &= C - \alpha_i^* - \mu_i^* &&= 0. \end{aligned}$$

i. e.,

$$\mathbf{w} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i), \qquad \sum_{i=1}^{N}\alpha_i = \sum_{i=1}^{N}\alpha_i^*, \tag{4.4}$$
$$\alpha_i^{(*)} \in [0, C].$$

Substituting Eq. (4.4) into Eq. (4.4) and applying the Dual Theory (Appendix A.1), we also obtain a QP problem,

$$\begin{aligned}\min \Phi(\boldsymbol{\alpha}^{(*)}) &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\boldsymbol{\phi}\langle \mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle \\ &\quad + \sum_{i=1}^{N}(u(\mathbf{x}_i) - y_i)\alpha_i + \sum_{i=1}^{N}(d(\mathbf{x}_i) + y_i)\alpha_i^*,\end{aligned} \tag{4.5}$$

subject to

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0, C].$$

The above QP problem is very similar to the original QP problem in [88], therefore, it is easy to modify the previous algorithm to implement this QP problem. Practically, we implement our QP problem by modifying the libSVM from [19] with adding a new data structure to store both margins: up margin, $u(\mathbf{x})$, and down margin, $d(\mathbf{x})$. Obviously, this will not impact the time complexity of the SVR algorithm; we just need more space, linear to the size of data points, to store the corresponding margins.

After solving this QP problem, we also obtain the objective function

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x})\rangle + b, \tag{4.6}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$ are corresponding Lagrange multipliers also used to pull and push f towards to the observation y

The computation of b is similar as in Section 3.2. The computation of b is also exploited by the Karush-Kuhn-Tucker (KKT) conditions. Here, they are

$$\begin{aligned}\alpha_i(u(\mathbf{x}_i) + \xi_i - y_i + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b) &= 0, \\ \alpha_i^*(d(\mathbf{x}_i) + \xi_i^* + y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - b) &= 0,\end{aligned}$$

and

$$\begin{aligned}(C - \alpha_i)\xi_i &= 0, \\ (C - \alpha_i^*)\xi_i^* &= 0.\end{aligned}$$

Therefore, when there exists i, such that $\alpha_i \in (0, C)$ or $\alpha_i^* \in (0, C)$, b can be computed as follows:

$$b = \begin{cases} y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle - u(\mathbf{x}_i), & \text{for } \alpha_i \in (0, C) \\ y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i) \rangle + d(\mathbf{x}_i), & \text{for } \alpha_i^* \in (0, C) \end{cases}.$$

When no $\alpha_i^{(*)} \in (0, C)$, methods e. g. [19] are used.

4.3 Accuracy Metrics and Risk Measures

In order to measure the prediction performance of our model, we define the Mean Absolute Error (MAE) first. Usually, Mean Squared Error (MSE) is used to measure the predictive performance. Here considering the particularity of the loss function, we use L_1 norm to measure the predictive errors.

Let a_t and p_t be the actual values and predicted values at day t, let m be the number of testing data.

Definition 2 Mean Absolute Error (MAE) measures the discrepancy between the actual and predicted values; the smaller the value of MAE, the closer are the predicted values to the actual values. MAE is calculated by

$$\text{MAE} = \frac{1}{m} \sum_{t=1}^{m} |a_t - p_t|. \tag{4.7}$$

We also consider the risk of using this model in the prediction. Actually, risk is a term frequently encountered in strategic management and financial literature. However, risk has a variety of different meanings and rarely is the meaning used in a particular project clarified [4]. In financial literature, Markowitz first formulated the portfolio selection into a mathematical model [56]. In his model, the "return" of a portfolio is measured by the expected value of the random portfolio return and the associated "risk" is quantified by the variance of the portfolio return. However, the use of variance to measure risk makes no distinction between gains and losses. Markowitz also

proposed to use semi-variance to measure the risk of loss. That is the sum of the squares of negative deviations from the mean, divided by the total number of observations:

$$\frac{1}{m}\sum_{t=1}^{m}[\min(r_t-\mu,0)]^2.$$

The great advantage of the use of semi-variance over variance is that it does not include positive gains, so what is considered as risk takes into account only negative deviations. However, minimizing downside does not mean minimizing only negative deviations. For example, if the distribution, like the normal curve, is symmetric, minimizing variance and semi-variance will lead to the same problem. The only case that justifies the use of semi-variance is when the presence of skewness is observed [2]. A generalization of semi-variance is given in [2],

$$\text{downside risk} \Rightarrow \frac{1}{m}\sum_{t=1}^{m}[\min(r_t-\mu,0)]^k, \tag{4.8}$$

where k is any power that one chooses; when k=1, it should be considered the absolute value of the term in the brackets and μ is a chosen benchmark (not necessarily the mean).

Based on Eq. (4.8), we choose k=1 and define the following risk measurements.

Definition 3 Up side Mean Absolute Error (UMAE) measures up side risk; the smaller the value of UMAE, the smaller the up side risk. UMAE is defined as

$$\text{UMAE} = \frac{1}{m}\sum_{\substack{t=1\\ a_t\geq p_t}}^{m}|a_t-p_t|. \tag{4.9}$$

Definition 4 Down side Mean Absolute Error (DMAE) measures the down side risk; the smaller the value of DMAE, the smaller the down side risk.

DMAE is defined as

$$\text{DMAE} = \frac{1}{m} \sum_{\substack{t=1 \\ a_t < p_t}}^{m} |a_t - p_t|. \tag{4.10}$$

Chapter 5

Margin Variation

We now have all the necessary tools to try our hand at predicting financial time series. Chapter 2 introduced the basic concept of time series analysis with different models descriptions; Chapter 3 and Chapter 4 detailed the SVR algorithm and ideas that we need to modify the models. The performance measures also are given in Chapter 4. In this chapter we will present the results of SVR by varying the width of margins with controlling the symmetry of margins; we will provide an answer to the question of whether or not these modifications offer any improvement over the original (fixed and symmetrical) margin setting; what is the performance of SVR model for the financial time series prediction. Ultimately, our answer to the first question will be 'yes'; however, we cannot provide a satisfiable answer to the second question. We cannot give out how confidence of our model, how better/worse the SVR model comparing with other models, and we cannot answer the most interesting problem, how can we profit by using this model.

5.1 Non-fixed Margin Cases

5.1.1 Momentum

In [98], we have focused on the case of NAAM. More specifically, we have added a momentum term in the margin setting. The margin is a linear combination

of the standard deviation and the momentum. The motivation here is that we would want to use the standard deviation to set the width of the margin, which could reflect the volatility of stock market; while we use the momentum term to detect the up(down) trend of stock market, such that we could reduce the downside risk. Hence, the up margin and down margin are set in the following forms:

$$\begin{aligned} u(\mathbf{x}_i) &= \lambda_1 \cdot \sigma(\mathbf{x}_i) + \mu \cdot \Delta(\mathbf{x}_i), \quad i = 1, \ldots, N, \\ d(\mathbf{x}_i) &= \lambda_2 \cdot \sigma(\mathbf{x}_i) - \mu \cdot \Delta(\mathbf{x}_i), \quad i = 1, \ldots, N, \end{aligned} \tag{5.1}$$

where $\sigma(\mathbf{x}_i)$ is the standard deviation of input $\mathbf{x}_i$, $\Delta(\mathbf{x}_i)$ is the momentum at point $\mathbf{x}_i$, λ_1, λ_2 are both positive constants and μ is a non-negative constant. Therefore, the width of margin at point $\mathbf{x}_i$ is

$$W(\mathbf{x}_i) = (\lambda_1 + \lambda_2) \cdot \sigma(\mathbf{x}_i).$$

It is determined by $\sigma(\mathbf{x}_i)$ and the sum of λ_1 and λ_2. Here, we call λ_1, λ_2 as the coefficients of the margin width. We also call μ as the coefficient of momentum and we know that the margin setting of Eq. (5.1) includes the case of NASM (when $\mu = 0$).

From our previous experience [96], when $\mu \neq 0$ and $\Delta(x) > 0$, the up margin is larger than the down margin and we can under-predict the stock price. While $\mu \neq 0$ and $\Delta(x) < 0$, the up margin is smaller than the down margin and we can over-predict the stock price. A simple illustration is shown in 5.1. Based on these observations, in our prediction, we assume that we are risk aversion, or downside risk aversion. When we find the stock price reveals an up trend, we know that it will not be always up, so we tend to under-predict the stock prices in this case. On the contrary, when the stock price goes down, we tend to over-predict it. We add this information in the margin setting by controlling the momentum term.

Actually, there are many ways to calculate the momentum. For example, the simplest way may be set it as a constant. In this paper, we will concentrate

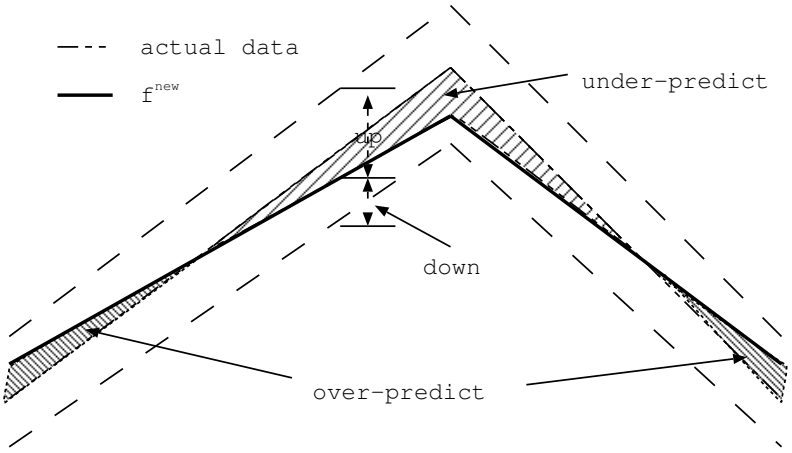


Figure 5.1: Margin settings: dashed lines are the bounds of margins; dashed-dotted lines are actual data series; solid-bold lines are the new objective function, f^{new}, by new margin settings. The upper shadow area is the case of new objective function under-predicted to the actual function; the lower shadow parts are the case of over-predicted.

on using the EMA, which has been introduced in subsection 2.1.1. The reason of using EMA is that it is time-varying and can reflect the up trend and down tendency of the financial data, although it exists the lag problem. An n-day's EMA sequence begins from the first day, i. e. $EMA_1 = y_1$ and the following is calculated by

$$EMA_i = EMA_{i-1} \times (1 - r) + y_i \times r,$$

where $r = 2/(1+n)$. Here, the current day's momentum is set as the difference between the current day's EMA and the EMA in the previous k day, i. e.

$$\Delta(\mathbf{x}_i) = EMA_i - EMA_{i-k}.$$

5.1.2 GARCH

In the above methods, the data sets we used in the experiments are the price of the share [96, 98]. We use the standard deviation of input $\mathbf{x}_t$, which can reflect the volatility of the financial time series over time, to determine the width of margin at time t in our prediction. Actually, GARCH model is a

more common used model to reflect the volatility of the financial time series, see Chapter 2.

We apply the Matlab toolbox to calculate the GARCH model. In the Matlab toolbox, the standard GARCH(p, q) model with Gaussian shocks takes the following form,

$$y_t = c_0 + \mathbf{x}_t^T \mathbf{b} + \epsilon_t, \quad \epsilon_t | \Psi_{t-1} = N(0, \sigma_t^2),$$

where

$$\sigma_t^2 = \kappa_0 + \sum_{i=1}^{p} \lambda_i \sigma_{t-i}^2 + \sum_{j=1}^{q} \mu_j \epsilon_{t-j}^2.$$

This GARCH toolbox is applied on the return series. So we use the continuous compounded return as the data series and use the σ_t calculated by GARCH(1,1) as the width of margin at time t.

5.2 Experiments

In this section, we will perform the experiments by using the momentum and GARCH models to set the margins.

The original experiments [96, 98] are conducted on a Pentium 4, with 1.4 GHZ, 512M RAM and Windows2000. Now they are conducted on Sun Blade 1000, RAM 2GB, 100Mbps network speed and Solaris 8.

5.2.1 Momentum

Two data sets are used in this experiment:

HSI: daily closing prices of Hong Kong's Hang Seng Index (HSI) from January 2nd, 1998 to December 29, 2000.

DJIA: daily closing prices of Dow Jones Industrial Average (DJIA) from January 2nd, 1998 to December 29, 2000.

The ratio of the number of training data and the number of testing data is set to 5:1. Therefore, the corresponding training time periods are obtained and listed as in Table 5.1.

SVR Algorithm

We model the system as $p_t = f(\mathbf{x}_t)$, where f is learned by the SVR algorithm from the training data, $\mathbf{x}_t = (a_{t-4}, a_{t-3}, a_{t-2}, a_{t-1})$, a_t is the daily closing index in day t.

Before generating the model, we do a cross-validation on the training data to determine the parameters that are needed in SVR. They are C, the cost of error and β, the parameter of kernel function. The corresponding parameters are also listed in Table 5.1. With these parameters, we begin to build the model by SVR from the initial training data. After obtaining the predictive value, we shift the input window to the next time-step and train the model again to obtain the next day's price. This one-step ahead prediction is done as the window shifted for the remaining data.

Table 5.1: Indices, time periods and parameters for momentum experiments

Indices	Training time periods	C	β
HSI	02/01/1998 - 04/07/00	16000	2^{-27}
DJIA	02/01/1998 - 29/06/00	8000	2^{-22}

Non-fixed Cases: The margins setting is followed as Eq. (5.1). In the case of NASM, we set $\lambda_1 = \lambda_2 = \frac{1}{2}$ and $\mu = 0$, thus the overall margin width at day t is equal to the standard deviation of input $\mathbf{x}_t$, $\sigma(\mathbf{x}_t)$.

In the case of NAAM, we also fix $\lambda_1 = \lambda_2 = \frac{1}{2}$, hence we have a fair comparison of NASM case. In addition, we have to determine three parameters, i.e., n, the length of EMA; k, the lag of EMA; μ, the coefficient of momentum. We have performed the following experiments to test their effects:

(a) At first, we set $k = 1$, $\mu = 1$ and use 10, 30, 50, 100 as the length of EMA respectively. From the result of Table 5.2 we can see that the DMAE values in all cases of NAAM are smaller than that in NASM case, thus we have a smaller downside risk in NAAM case; this exactly meets our assumption. We also see that the MAE gradually decreases with the length of EMA increases and when the length equals to 100, the MAE and the DMAE are the smallest in all case of NAAM for data set HSI. For data set DJIA, when the length equals to 30, the MAE and the DMAE are also the smallest in all cases of NAAM.

Table 5.2: Effect of the length of EMA on HSI with parameters (k, μ)=(1,1)

		HSI			DJIA		
type	n	MAE	UMAE	DMAE	MAE	UMAE	DMAE
NASM		216.78	104.58	112.20	85.33	40.29	45.04
NAAM	10	222.43	115.64	106.79	85.68	43.13	42.55
	30	218.18	114.04	104.14	**84.12**	41.82	**42.30**
	50	217.93	113.38	104.55	84.57	42.12	42.45
	100	**216.50**	113.04	**103.46**	84.80	42.41	42.39

In the following, we will use the best length of EMA from the above experiments for the corresponding data sets, i. e., $n = 100$ for data set HSI and $n = 30$ for data set DJIA.

(b) When testing the effect of lag, k, we let $\mu = 1$ and set k to to 1, 2, 4, 8 respectively for both data sets. The results are listed in Table 5.3. They show that the MAE increases with the lag of EMA increases. These indicate that the results when the lag of EMA equals to 1 are superior to the other cases.

(c) Here, we set $k = 1$ and $\mu = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ respectively for both data set to see the effect of the μ. From the Table 5.4, we see that the DMAE increases gradually with the coefficient of EMA decreases

and the MAE is smaller than the value in the NASM case. The change of the MAE for data set HSI in (2–4 columns of) Table 5.4 is fluctuating and the MAE in (5–7 columns of) Table 5.4 increases gradually with the decrease of the coefficient of EMA.

Table 5.3: Effect of the distance of EMA on HSI and DJIA

	HSI with $(n, \mu) = (100, 1)$			DJIA with $(n, \mu) = (30, 1)$		
k	MAE	UMAE	DMAE	MAE	UMAE	DMAE
1	216.50	113.04	103.46	84.12	41.82	42.30
2	219.02	125.30	93.72	85.42	43.91	41.51
4	228.25	149.36	78.88	90.99	49.16	41.83
8	260.73	200.74	59.99	103.77	58.03	45.74

Table 5.4: Effect of the coefficient of Momentum on HSI and DJIA

	HSI with $(n, k) = (100, 1)$			DJIA with $(n, k) = (30, 1)$		
μ	MAE	UMAE	DMAE	MAE	UMAE	DMAE
1	216.50	113.04	103.46	84.12	41.82	42.30
$\frac{1}{2}$	216.55	108.97	107.58	84.88	41.32	43.56
$\frac{1}{4}$	216.19	106.36	109.83	85.02	41.14	43.88
$\frac{1}{8}$	216.41	105.32	111.08	85.22	40.86	44.36

We also plot the daily closing prices of HSI with 100-days' EMA and the prices of DJIA with 30-days' EMA in Figure 5.2 and Figure 5.3 respectively and list the Average Standard Deviations (ASD) of input $\mathbf{x}$ of the training data sets, HSI and DJIA, respectively in Table 5.5, the Average of Absolute Momentums (AAM) of input $\mathbf{x}$ for the best length of both training data sets respectively in Table 5.5. We can observe that the ASD of HSI is higher than that of DJIA and the ratio of AAM to ASD is smaller for HSI than that for DJIA.

Now, we will make a summary for the above experiments. At first, we can know the effects of n, k and μ from the above experiments results. Following these results, we can say that a suitable setting for $k = 1$ and $\mu = 1$ will both be 1, they can be applied when a new data set comes.

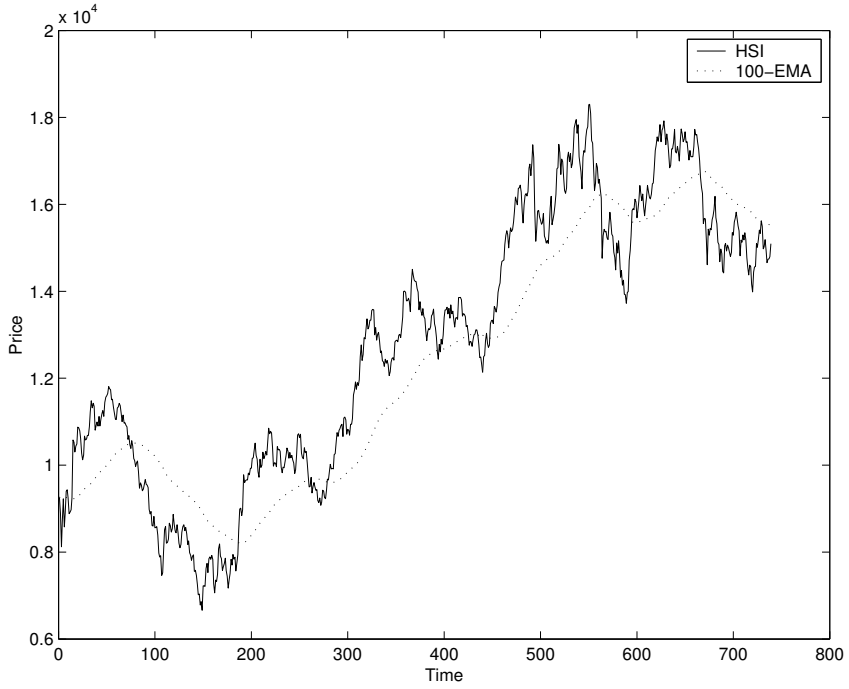


Figure 5.2: HSI with 100 days' EMA.

Table 5.5: ASD and AAM

data set	ASD	AAM		ratio
		n	Δ	
HSI	182.28	100	20.80	0.114
DJIA	79.95	30	15.64	0.196

The only parameter needs to determine is the length of EMA, n, this may refer to the ASD of the training data set. When the ASD is larger, we may use a longer length of EMA. On the contrary, when the ASD is smaller, we may use a shorter length of EMA.

Fixed Cases: After considering the non-fixed margin cases, we also test the predictive results of fixed margins. Actually, for data set HSI, we let the width of margin equal to 200 (approximate to the ASD of HSI), i. e., $u(\mathbf{x}) + d(\mathbf{x}) = 200$. The up margin $u(\mathbf{x})$ ranges from 0 to 200, each increment is one-tenth of 200, i. e., 20. The results are listed in the (1–5 columns of) Table 5.6. Similarly, for data set DJIA, we let the width of

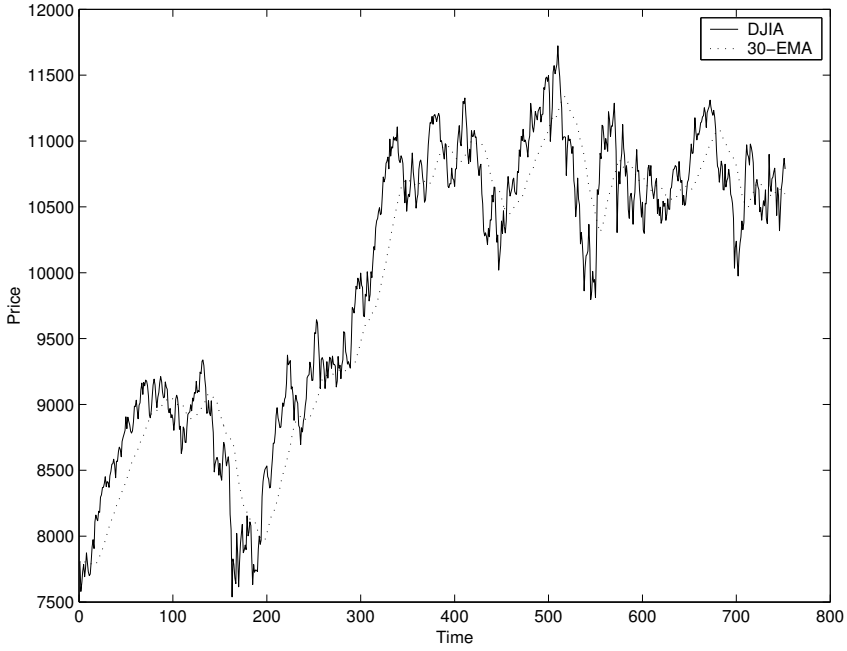


Figure 5.3: DJIA with 30 days' EMA.

margin equal to 90 (approximate to ASD of DJIA), i. e., $u(\mathbf{x})+d(\mathbf{x}) = 90$. The up margin $u(\mathbf{x})$ ranges from 0 to 90, each increment is also one-tenth of 90, i. e., 9. The results are listed in the (6–10 columns of) Table 5.6. We can see that for both data sets, as the up margin increases, the DMAE tends to decrease.

Comparing the results in Table 5.2 with the results in Table 5.6 (the results comparison graphs are plotted in Fig. 5.4(b) and Fig. 5.5(b) respectively), we can see that NASM and NAAM are both superior to FASM and FAAM in both data sets.

In the following, we will perform other models, such as AR models and RBF network, on the above two data sets. The best results comparison graphs of all the models are illustrated in Fig. 5.4(a) for HSI and Fig. 5.5(a) respectively.

Table 5.6: Results of FASM and FAAM for HSI and DJIA

HSI ($u(\mathbf{x})$+$d(\mathbf{x})$ = 200)					DJIA ($u(\mathbf{x})$+$d(\mathbf{x})$ = 90)				
$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE	$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
0	200	236.04	62.24	173.80	0	90	91.63	20.45	71.18
20	180	230.85	69.65	161.20	9	81	89.14	23.70	65.44
40	160	226.29	77.37	148.92	18	72	87.35	27.31	60.04
60	140	222.24	85.34	136.90	27	63	86.09	31.18	54.91
80	120	219.35	93.90	125.45	36	54	85.30	35.28	50.02
100	100	217.83	103.14	114.69	45	45	85.45	39.86	45.59
120	80	217.35	112.90	104.45	54	36	86.33	44.80	41.53
140	60	217.88	123.16	94.72	63	27	87.40	49.83	37.57
160	40	219.49	133.97	85.52	72	18	88.64	54.95	33.69
180	20	221.66	145.05	76.61	81	9	90.80	60.53	30.27
200	0	224.83	156.64	68.19	90	0	93.75	66.51	27.24

Table 5.7: Results on AR(4)

data set	MAE	UMAE	DMAE
HSI	217.75	105.96	111.79
DJIA	88.74	46.36	42.38

AR Models

For AR models, we use the AR model with order 4 to predict the prices of HSI and DJIA, hence we can compare the AR model with NASM, NAAM in SVR with the same order in the input patterns, X. The results are listed in the Table 5.7. From these results, we can see that NASM and NAAM are superior to AR model with same order.

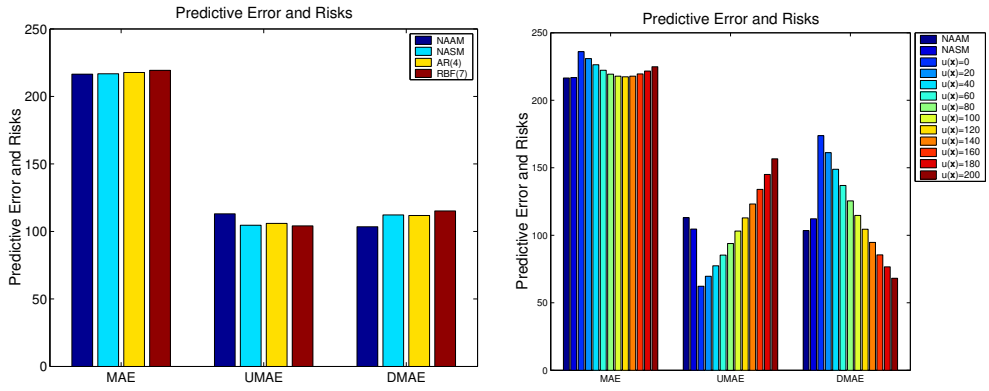
RBF Network

For the RBF network, we use the RBF network which was implemented in NETLAB [59] and perform the one-step ahead prediction to predict the prices of HSI and DJIA. Concretely, we let other parameters as default and set the number of hidden units to 3, 5, 7, 9 to learn f by training the RBF network on the training samples and we get the results in Table 5.8 for both data sets. Comparing the results in Table 5.2 with the results in Table 5.8, we can see

that NASM and NAAM are also better than RBF network.

Table 5.8: Effect of number of hidden units on HSI and DJIA

	HSI			DJIA		
# hidden	MAE	UMAE	DMAE	MAE	UMAE	DMAE
3	386.65	165.08	221.57	88.31	44.60	43.71
5	277.83	128.92	148.91	98.44	48.46	49.98
7	219.32	104.15	115.17	90.53	46.22	44.31
9	221.81	109.46	112.35	87.23	44.09	43.14



(a) NAAM,NASM vs. AR(4),RBF(7) (b) NAAM,NASM vs. FAAM,FASM

Figure 5.4: Experimental results comparison graphs of HSI.

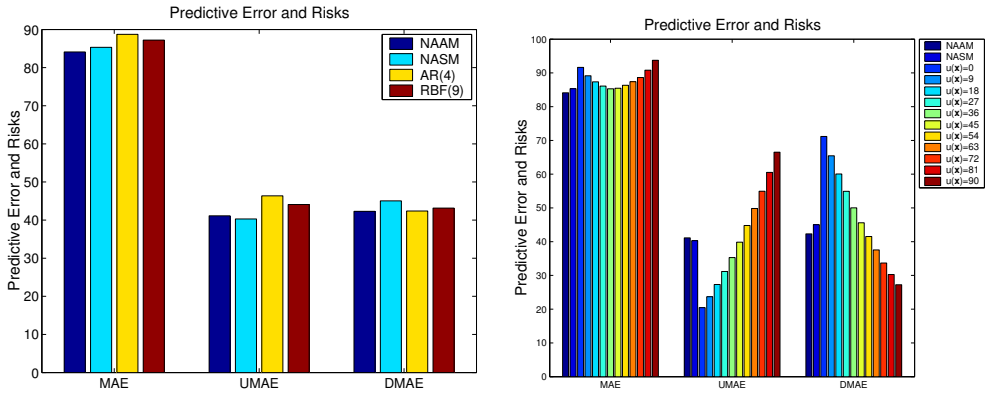
5.2.2 GARCH

In this experiment, the experimental data are 3 years' daily closing indices (2000-2002) from stock markets in different countries:

Nikkei225: Nikkei225 Stock Average from Japan, the daily closing prices are plotted in Fig. 5.9(a);

DJIA00-02: Dow Jones Industrial Average (DJIA) from U.S.A., the daily closing prices are plotted in Fig. 5.11(a)

FTSE100: FTSE100 index from U.K., the daily closing prices are plotted Fig. 5.13(a).



(a) NAAM,NASM vs. AR(4),RBF(7) (b) NAAM,NASM vs. FAAM,FASM

Figure 5.5: Experimental results comparison graphs of DJIA.

In the data processing step, the daily closing prices of these indices are converted to continuously compounded returns and the ratio of the number of training data to the number of testing data is set to 5:1. Therefore, we obtain and list the corresponding training and testing period in Table 5.9.

Table 5.9: GARCH experimental data description

Indices	Training Period	Testing Period
Nikkei225	4-Jan.-2000 $\sim$ 2-Jul.-2002	4-Jul.-2002 $\sim$ 30-Dec.-2002
DJIA00-02	3-Jan.-2000 $\sim$ 3-Jul.-2002	5-Jul.-2002 $\sim$ 31-Dec.-2002
FTSE100	4-Jan.-2000 $\sim$ 3-Jul.-2002	4-Jul.-2002 $\sim$ 31-Dec.-2002

GARCH(1,1)

Before running the SVR algorithm, we run the GARCH(1,1) model to determine the width of margin in SVR. For Nikkei225, we obtain the parameter estimates and their standard errors in Table 5.10, i. e., the best fits for Nikkei225 by GARCH(1,1) is

$$\begin{aligned} y_t &= 0.49468 + \varepsilon_t, \\ \sigma_t^2 &= 0.00073917 + 0.8682\sigma_{t-1}^2 + 0.077218\varepsilon_{t-1}^2. \end{aligned}$$

We also show the log-likelihood contours of GARCH(1,1) model fit to the returns of data set, Nikkei225. The log-likelihood contours are plotted in a GARCH Coefficient-ARCH Coefficient ($G_1 - A_1$) plane, holding the parameters c_0 and κ_0 fixed at their maximum likelihood estimates 0.49468 and 0.00073917, respectively. The contours confirm the results in Table 5.10. The maximum log-likelihood value occurs at the coordinates $G_1 = \text{GARCH}(1) = 0.8682$ and $A_1 = \text{ARCH}(1) = 0.077218$. This figure also reveals a highly negative correlation between the estimates of the G_1 and A_1 parameters of the GARCH(1,1) model. It implies that a small change in the estimate of the G_1 parameter is nearly compensated for a corresponding change of opposite sign in the A_1 parameter. The innovations, standard deviations (σ_t) and returns of Nikkei225 are shown in Fig. 5.6(b). For data set DJIA00-02, GARCH(1,1) parameter estimates are listed in Table 5.11, i.e., the best fits for DJIA00-02 by GARCH(1,1) is

$$\begin{aligned} y_t &= 0.60363 + \varepsilon_t, \\ \sigma_t^2 &= 0.00056832 + 0.85971\sigma_{t-1}^2 + 0.092295\varepsilon_{t-1}^2. \end{aligned}$$

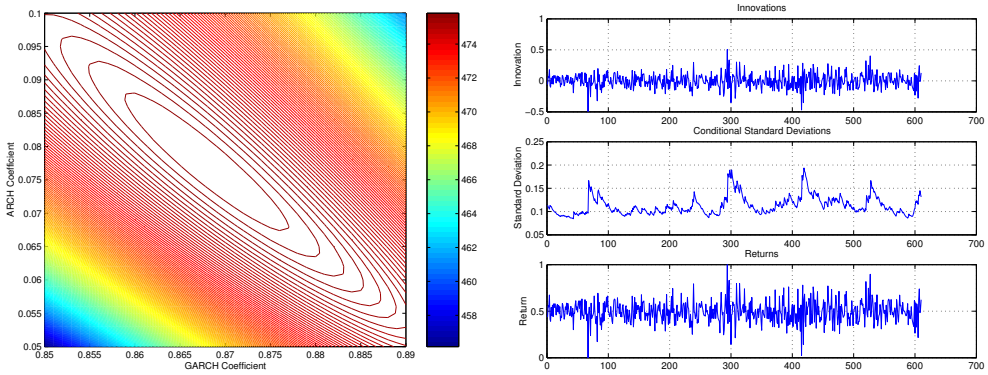
The corresponding log-likelihood contours of DJIA00-02 are plotted in Fig. 5.7(a), the maximum log-likelihood value occurs at the coordinates $G_1 = \text{GARCH}(1) = 0.85971$ and $A_1 = \text{ARCH}(1) = 0.09229$. The corresponding innovations, standard deviation and returns of DJIA00-02 are shown in Fig. 5.7(b). For data set FTSE100, GARCH(1,1) parameter estimates are listed in Table 5.12, i.e., the best fits for FTSE100 by GARCH(1,1) is

$$\begin{aligned} y_t &= 0.50444 + \varepsilon_t, \\ \sigma_t^2 &= 0.0011599 + 0.82253\sigma_{t-1}^2 + 0.12693\varepsilon_{t-1}^2. \end{aligned}$$

The corresponding log-likelihood contours of FTSE100 are plotted in Fig. 5.8(a), the maximum log-likelihood value occurs at the coordinates $G_1 = \text{GARCH}(1) = 0.82253$ and $A_1 = \text{ARCH}(1) = 0.12693$. The corresponding innovations, standard deviation and returns of FTSE100 are shown in Fig. 5.8(b).

Table 5.10: GARCH parameter for Nikkei225

Parameter	Value	Standard Error	T Statistic
c_0	0.49468	0.0045008	109.9083
κ_0	0.00073917	0.00034866	2.1200
GARCH(1)	0.8682	0.048144	18.0334
ARCH(1)	0.077218	0.027279	2.8306



(a) GARCH(1,1) log-likelihood contours of Nikkei225

(b) Innovations, conditional standard deviations and returns of Nikkei225

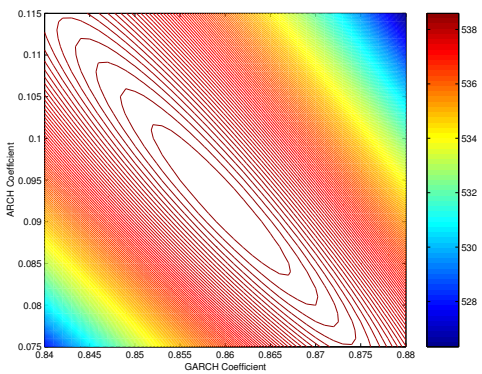
Figure 5.6: GARCH(1,1) of Nikkei225. The color-coded bar at the right of (a) indicates the height of the log-likelihood surface of the GARCH(1,1) plane.

Table 5.11: GARCH parameter for DJIA00-02

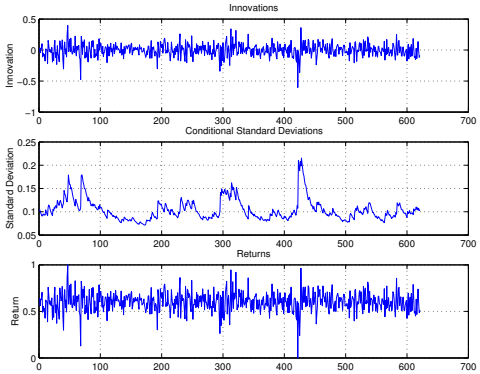
Parameter	Value	Standard Error	T Statistic
c_0	0.60363	0.0041185	146.5631
κ_0	0.00056832	0.00023491	2.4193
GARCH(1)	0.85971	0.031773	27.0580
ARCH(1)	0.092295	0.020352	4.5350

Table 5.12: GARCH parameter for FTSE100

Parameter	Value	Standard Error	T Statistic
c_0	0.50444	0.0053313	94.6180
κ_0	0.0011599	0.00049206	2.3573
GARCH(1)	0.82253	0.04906	16.7658
ARCH(1)	0.12693	0.034698	3.6582

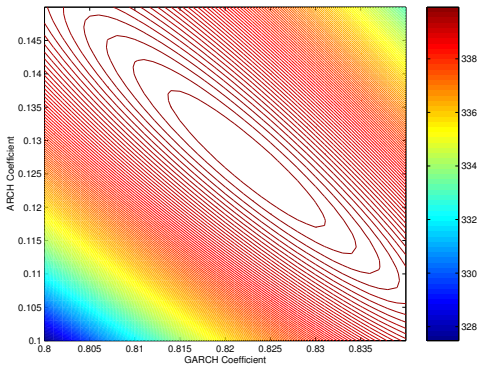


(a) GARCH(1,1) log-likelihood contours of DJIA00-02

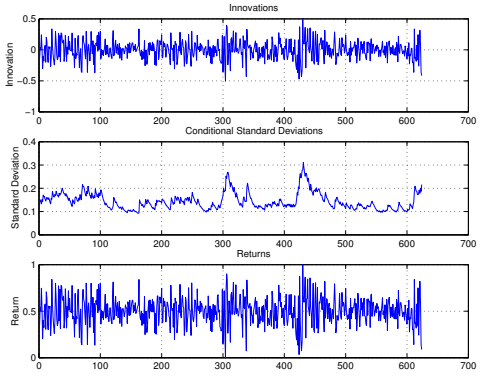


(b) Innovations, conditional standard deviations and returns of DJIA00-02

Figure 5.7: GARCH(1,1) of DJIA00-02. The color-coded bar at the right of (a) indicates the height of the log-likelihood surface of the GARCH(1,1) plane.



(a) GARCH(1,1) log-likelihood contours of FTSE100



(b) Innovations, conditional standard deviations and returns of FTSE100

Figure 5.8: GARCH(1,1) of FTSE100. The color-coded bar at the right of (a) indicates the height of the log-likelihood surface of the GARCH(1,1) plane.

SVR Algorithm

For SVR algorithm, the experimental procedure consists of three steps: at first, we normalize the return value by $t_i = \frac{r_i - r_{low}}{r_{high} - r_{low}}$, where r_i is the actual return of the stock at day i, r_{low} and r_{high} are the correspondingly minimum and maximum return in the training data respectively. Then, we train the normalized training data once and then obtain the normalized predicted return value $p_{n_i} = f(\mathbf{x}_i)$, where $\mathbf{x}_i = (t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1})$. Finally, we unnormalize p_{n_i}, convert the result to price and obtain the corresponding predicted price p_i.

Before running the SVR algorithm, we have to choose two parameters: C, the cost of error; β, the parameter of kernel function. Here the parameters we choose are the same respectively for different indices. They are listed in Table 5.13.

Here, we just consider the case of NASM; the margin setting is as Eq. (5.1). Concretely, we set the margin width to σ calculated by GARCH(1,1) from return series y, therefore $\lambda_1 = \lambda_2 = \frac{1}{2}$ and $\mu = 0$. For fixed margin cases, we set the margin width as 0.1, i. e., $u(\mathbf{x}) + d(\mathbf{x}) = 0.1$, and each increment is 0.02. The corresponding predictive results are shown in the Table 5.15, Table 5.16 and Table 5.17, respectively. The corresponding training error results are shown in Table 5.14. We also plot the training and testing data results of NAAM in Fig. 5.10(a) and Fig. 5.10(b) for index Nikkei225, in Fig. 5.12(a) and Fig. 5.12(b) for index DJIA00-02, in Fig. 5.14(a) and Fig. 5.14(b) for index FTSE100, respectively. From these results, we can see that for FTSE100 index, NASM outperforms in the prediction than fixed margin cases. For Nikkei225, when $u(\mathbf{x}) = 0.06, d(\mathbf{x}) = 0.04$ and $u(\mathbf{x}) = 0.08, d(\mathbf{x}) = 0.02$, the predicted results are better than NASM. For DJIA00-02, when $u(\mathbf{x}) = 0.06, d(\mathbf{x}) = 0.04$, the predicted result is slightly better than NASM.

AR Models

We also use AR model with different orders (1-6) to predict the prices of the above three indices. The experimental procedure is to apply the AR model on training return series and to obtain the predicted return value from testing data. Then we convert the predicted return values to price values. We obtain the experimental results and show them in Table 5.18. After comparing the results in Table 5.15, Table 5.17 with the results in 2–4 and 8–10 columns of Table 5.18, we can see that for Nikkei225 and FTSE100 index, the NASM method is better than AR model. For DJIA, we can see that NASM method is slight worse than AR(1), but better than other order of AR model.

For index Nikkei225, the graphs of the predictive error and risks comparison results are shown in Fig. 5.9(b), the corresponding bar values are from Table 5.15 and (2–4 columns of) Table 5.18. The predictive error and risks of DJIA00-02 are shown in Fig. 5.11(b), where the corresponding bar values are from Table 5.16 and (5–7 columns of) Table 5.18. The predictive error and risks of FTSE100 are shown in Fig. 5.13(b), where the corresponding bar values are from Table 5.17 and (8–10 columns of) Table 5.18.

RBF Network

For the RBF network, we use the RBF network implemented in NETLAB [59] and perform the one-step ahead prediction to predict the returns of the above three dataset, then we convert the predictive return to price and compare their values with actual price values. Concretely, we let other parameters as default and set the number of hidden units to 3, 5, 7, 9 to learn f by training the RBF network on the training samples and we get the results in Table 5.19. We can see that NASM is better than RBF network for dataset Nikkei225 and FTSE100, other than DJIA00-02.

5.3 Discussions

Having described the experiments and their results, we know that NASM is superior to FASM and FAAM generally. One reason is that NASM catches the stock market information and adds the information into the setting of the margin. This provides helpful information for the prediction. Another reason is that by using NASM, the margin width is determined by a meaningful value. This value changes with the stock market. Obviously, this method is more flexible than fixed margin cases and avoids risk of getting bad predictive results partially when the margin values are determined by random selection in the fixed margin cases.

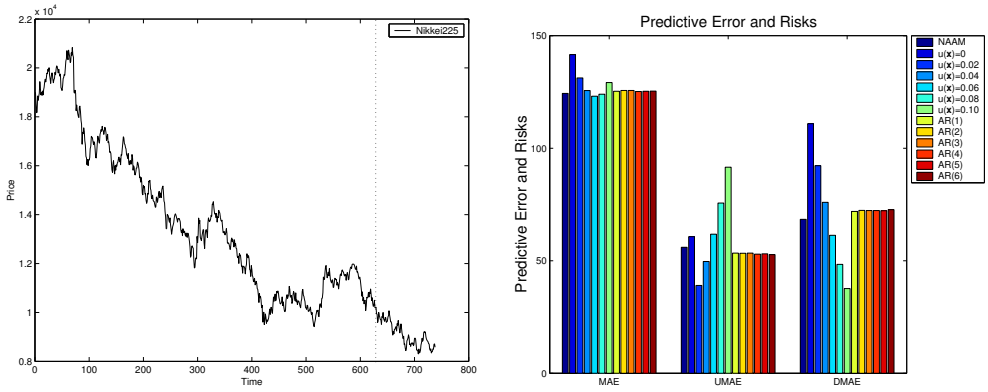
Furthermore, we know that NAAM may be better than NASM. For example, by adding a momentum, we may not only improve the accuracy of prediction, but also reduce the predictive downside risk.

Another notice is that by cautiously selecting parameters, SVR algorithm has similar predictive performance to other models, from Fig. 5.4(a) and Fig. 5.5(a). However, for a novice, the SVR libraries are easy to run. Since every local optimum is the global optimum, it guarantees the user to find an optimal solution easily and stably. This advantage is very useful for a novice to learn a new model, or library, and strengthen their confidence of learning new things comparing with learning other non-linear model, e. g. RBF networks.

In general, our methods can be considered as a model selection, determining the parameter, ε,. We do not consider the setting of other parameters, such as C and β. We just use the cross-validation technique to find suitable values for them. However, this procedure is time-consuming. We may add some market information to set these parameters, e. g. [16]. In addition, the margin width set by GARCH model is too wide; we may need to add more useful term to shrink it. This can be one of our future works. A valuable experience is that the normalized procedure will be helpful for selecting suitable parameters

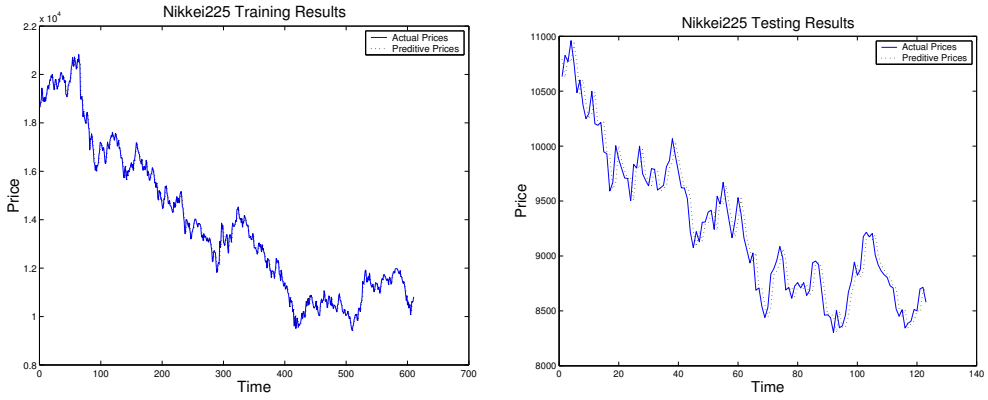
easily and stably.

Finally, we turn to a key weakness of our model; the predictive model does not lead to direct profit making in real life and we do not provide the confidence of these predictive models. However, we may find some useful information through using our model to predict the stock market prices; the predictive results may provide some helpful suggestions.



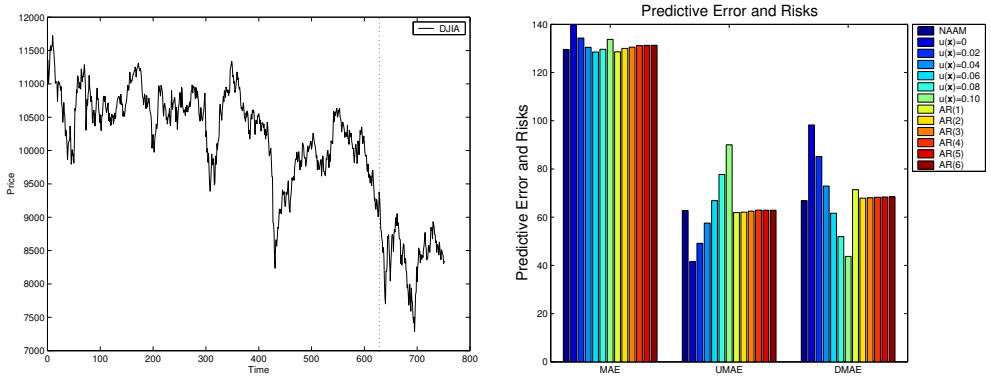
(a) Data plot (b) Results comparison

Figure 5.9: Nikkei225 data plot and experimental results graphs.



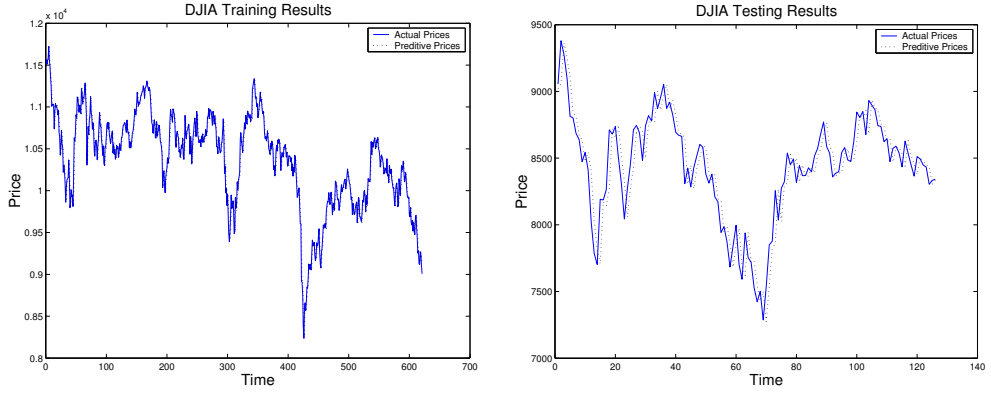
(a) Training results (b) Testing results

Figure 5.10: Experimental results graphs using GARCH method for Nikkei225.



(a) Data plot (b) Results comparison

Figure 5.11: DJIA00-02 data plot and experimental results graphs.



(a) Training results (b) Testing results

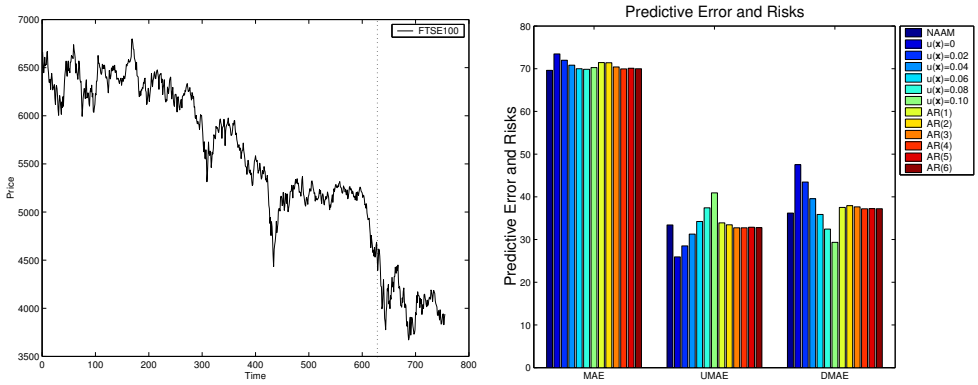
Figure 5.12: Experimental results graphs using GARCH method for DJIA00-02.

Table 5.13: Parameters in GARCH experiments for NASM

Indices	C	β	Indices	C	β	Indices	C	β
Nikkei225	2	2^{-4}	DJIA	2	2^{-4}	FTSE100	2	2^{-4}

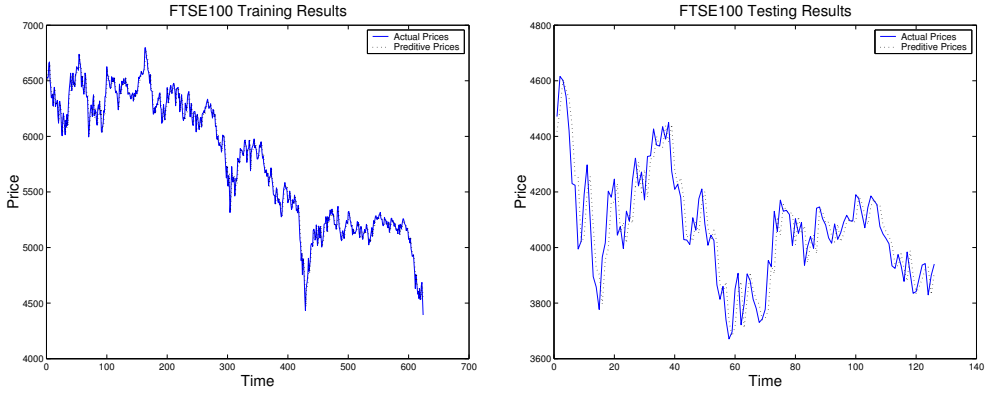
Table 5.14: SVR training results

Nikkei225			DJIA00-02			FTSE100		
MAE	UMAE	DMAE	MAE	UMAE	DMAE	MAE	UMAE	DMAE
166.51	81.68	84.83	99.30	48.78	50.52	52.74	26.08	26.66



(a) Data plot (b) Results comparison

Figure 5.13: FTSE100 data plot and experimental results graphs.



(a) Training results (b) Testing results

Figure 5.14: Experimental results graphs using GARCH method for FTSE100.

Table 5.15: SVR results for Nikkei225

Type	$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
NASM	σ	σ	124.37	55.97	68.40
FAAM	0	0.1	141.6	30.7	110.9
	0.02	0.08	131.25	39.02	92.23
	0.04	0.06	125.63	49.66	75.97
	0.06	0.04	123.11	61.81	61.3
	0.08	0.02	124	75.63	48.37
	0.10	0	129.19	91.56	37.63

Table 5.16: SVR results for DJIA00-02

Type	$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
NASM	σ	σ	129.56	62.74	66.83
FAAM	0	0.1	139.82	41.56	98.26
	0.02	0.08	134.33	49.16	85.17
	0.04	0.06	130.49	57.56	72.93
	0.06	0.04	128.51	66.87	61.64
	0.08	0.02	129.65	77.72	51.94
	0.10	0	133.76	90.02	43.74

Table 5.17: SVR results for FTSE100

Type	$u(\mathbf{x})$	$d(\mathbf{x})$	MAE	UMAE	DMAE
NASM	σ	σ	69.61	33.42	36.19
FAAM	0	0.1	73.46	25.93	47.53
	0.02	0.08	71.98	28.52	43.46
	0.04	0.06	70.83	31.27	39.56
	0.06	0.04	70.1	34.22	35.88
	0.08	0.02	69.86	37.42	32.45
	0.10	0	70.26	40.92	29.34

Table 5.18: AR results

	Nikkei225			DJIA00-02			FTSE100		
Order	MAE	UMAE	DMAE	MAE	UMAE	DMAE	MAE	UMAE	DMAE
1	125.31	53.40	71.91	128.58	61.67	66.91	71.44	33.9	37.53
2	125.68	53.31	72.36	130.00	62.08	67.92	71.40	33.46	37.94
3	125.67	53.37	72.30	130.56	62.50	68.06	70.41	32.76	37.65
4	125.22	52.91	72.31	131.20	62.93	68.27	69.96	32.76	37.20
5	125.32	53.08	72.24	131.27	62.90	68.38	70.12	32.89	37.23
6	125.40	52.72	72.68	131.32	62.89	68.43	69.99	32.78	37.21

Table 5.19: RBF results

	Nikkei225			DJIA00-02			FTSE100		
#hidden	MAE	UMAE	DMAE	MAE	UMAE	DMAE	MAE	UMAE	DMAE
3	125.33	57.04	68.28	128.89	60.43	68.50	71.24	35.10	36.14
5	124.76	56.85	67.91	128.71	60.28	68.44	71.20	33.32	37.90
7	124.55	56.94	67.61	127.50	58.92	68.57	70.92	31.25	39.67
9	125.14	57.46	67.69	127.94	69.82	68.13	71.85	32.20	39.65

Chapter 6

Relation between Downside Risk and Asymmetrical Margin Settings

From our previous work [96], it is interesting to note that when the up margin is increased, the predicted value will become smaller. In this chapter, we formalize this phenomenon and find out the condition to keep this result valid. The result also leads to control the predictive downside risk. Practically, we also propose an algorithm to check the validity of the condition, such that we may know the changing trend of predictive downside risk only by running this algorithm on the training data set without doing practical SVR training procedure.

6.1 Mathematical Derivation

Let the symmetrical kernel matrix be $\mathbf{K} = (k_{ij})$ and the kernel function be RBF, Eq. (3.13), i. e., $k_{ij} = \exp(-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2)$, and suppose

$$\mathbf{K}^{-1}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_N \end{pmatrix}. \tag{6.1}$$

We will note that when the above d_i are greater than 0, if we increase the up margin, we will obtain a smaller predictive value. So we state the result as follows:

Theorem 5 Let a condition be defined as,

$$d_i > 0, \quad \forall i = 1, 2, \ldots, N, \tag{6.2}$$

If the condition in Eq. (6.2) is valid and the margin setting is FASM or FAAM, the decision function of SVR using RBF kernel function will be a monotone decreasing function to the up margin, $u(\mathbf{x})$.

Proof: Here we just consider the kernel function, RBF. From Eq. (4.6), the decision function of SVR for a testing data $\mathbf{x}_t$ is

$$\begin{aligned} f(\mathbf{x}_t) &= \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \kappa(\mathbf{x}_t, \mathbf{x}_i) + b \\ &= \boldsymbol{\gamma}(\mathbf{x}_t)^T \tilde{\boldsymbol{\alpha}} + b, \end{aligned}$$

$$\begin{aligned} \text{where} \quad \kappa(\mathbf{x}_t, \mathbf{x}_i) &= \exp(-\beta \|\mathbf{x}_t - \mathbf{x}_i\|^2) > 0, \\ \boldsymbol{\gamma}(\mathbf{x}_t) &= [\kappa(\mathbf{x}_t, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_t, \mathbf{x}_N)]^T, \\ \tilde{\boldsymbol{\alpha}} &= [\alpha_1 - \alpha_1^*, \ldots, \alpha_N - \alpha_N^*]^T, \end{aligned} \tag{6.3}$$

and $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$ are generated from QP problem in Eq. (4.5), which is rewritten in a matrix notation as,

$$\min \; Q(\bar{\boldsymbol{\alpha}}) = \frac{1}{2} \bar{\boldsymbol{\alpha}}^T \mathbf{Q} \bar{\boldsymbol{\alpha}} + \mathbf{p}^T \bar{\boldsymbol{\alpha}}, \tag{6.4}$$

subject to

$$\mathbf{1}_N^T \hat{\boldsymbol{I}}_N \bar{\boldsymbol{\alpha}} = 0, \quad \mathbf{0} \leq \boldsymbol{\alpha}^{(*)} \leq C \mathbf{1}_{2N},$$

where

$$\begin{aligned}
\bar{\boldsymbol{\alpha}} &= [\alpha_1, \ldots, \alpha_N, \alpha_1^*, \ldots, \alpha_N^*]^T, \\
\mathbf{1}_N &= [1, \ldots, 1]^T, \quad \text{an } N \times 1 \text{ vector}, \\
\mathbf{Q} &= \begin{bmatrix} \mathbf{K} & -\mathbf{K} \\ -\mathbf{K} & \mathbf{K} \end{bmatrix}, \mathbf{K} = [\boldsymbol{\gamma}(\mathbf{x}_1), \ldots, \boldsymbol{\gamma}(\mathbf{x}_N)], \ \text{an } N \times N \text{ matrix}, \\
\hat{\boldsymbol{I}}_N &= [\mathbf{I}_N, -\mathbf{I}_N], \quad \mathbf{I}_N \text{ is an } N \times N \text{ identity matrix}, \\
\mathbf{p} &= [u(\mathbf{x}_1) - y_1, \ldots, u(\mathbf{x}_N) - y_N, d(\mathbf{x}_1) + y_1, \ldots, d(\mathbf{x}_N) + y_N]^T.
\end{aligned}$$

The optimal solution of Eq. (6.4) is equivalent to finding an $\bar{\boldsymbol{\alpha}}$, such that minimize the following Lagrange function,

$$L(\bar{\boldsymbol{\alpha}}) = \frac{1}{2}\bar{\boldsymbol{\alpha}}^T\mathbf{Q}\bar{\boldsymbol{\alpha}} + \mathbf{p}^T\bar{\boldsymbol{\alpha}} - \lambda\mathbf{1}_N^T\hat{\boldsymbol{I}}_N\bar{\boldsymbol{\alpha}} - \boldsymbol{\mu}^T(C\mathbf{1}_{2N} - \bar{\boldsymbol{\alpha}}) - \boldsymbol{\nu}^T\bar{\boldsymbol{\alpha}}, \tag{6.5}$$

where $\lambda, \mu_i, \nu_i, i = 1, \ldots, 2N$ are corresponding Lagrange multipliers.

Therefore, $\bar{\boldsymbol{\alpha}}$ satisfies the following equation,

$$\mathbf{Q}\bar{\boldsymbol{\alpha}} + \mathbf{p} - \lambda(\mathbf{1}_N^T\hat{\boldsymbol{I}}_N)^T + \boldsymbol{\mu} - \boldsymbol{\nu} = 0,$$

i. e.,

$$\begin{aligned}
\mathbf{K}\tilde{\boldsymbol{\alpha}} + \mathbf{p}_{1..N} - \lambda\mathbf{I}_N + \boldsymbol{\mu}_{1..N} - \boldsymbol{\nu}_{1..N} &= \mathbf{0}, \\
-\mathbf{K}\tilde{\boldsymbol{\alpha}} + \mathbf{p}_{N+1..2N} + \lambda\mathbf{I}_N + \boldsymbol{\mu}_{N+1..2N} - \boldsymbol{\nu}_{N+1..2N} &= \mathbf{0},
\end{aligned}$$

where $\mathbf{l}_{1..N} = [l_1, \ldots, l_N], \mathbf{l}_{N+1..2N} = [l_{N+1}, \ldots, l_{2N}], l = \mathbf{p}, \boldsymbol{\mu}$ or $\boldsymbol{\nu}$.

Subtracting the above two equations and multiplying both side of the equation by $\frac{1}{2}\mathbf{K}^{-1}$, we obtain

$$\tilde{\boldsymbol{\alpha}} = -\frac{1}{2}\mathbf{K}^{-1}[\mathbf{p}_{1..N} - \mathbf{p}_{N+1..2N} - 2\lambda\mathbf{I}_N + (\boldsymbol{\mu}_{1..N} - \boldsymbol{\mu}_{N+1..2N}) - (\boldsymbol{\nu}_{1..N} - \boldsymbol{\nu}_{N+1..2N})],$$

Hence,

$$\begin{aligned}
f(\mathbf{x}_t) = \boldsymbol{\gamma}(\mathbf{x}_t)^T\tilde{\boldsymbol{\alpha}} + b &= -\frac{1}{2}\boldsymbol{\gamma}(\mathbf{x}_t)^T\mathbf{K}^{-1}[\mathbf{p}_{1..N} - \mathbf{p}_{N+1..2N} - 2\lambda\mathbf{I}_N \\
&+ (\boldsymbol{\mu}_{1..N} - \boldsymbol{\mu}_{N+1..2N}) - (\boldsymbol{\nu}_{1..N} - \boldsymbol{\nu}_{N+1..2N})] + b.
\end{aligned}$$

In the fixed margin cases, all up margin are the same, while all down margin are the same, and the sum of up margin and down margin equals a constant value, say c. That is we have $u(\mathbf{x}_i) = u(\mathbf{x}_t)$ and $d(\mathbf{x}_i) = c - u(\mathbf{x}_t)$, $i = 1, \ldots, N$. Therefore,

$$f(\mathbf{x}_t) = -\frac{1}{2}\boldsymbol{\gamma}(\mathbf{x}_t)^T\mathbf{K}^{-1}[\tilde{\boldsymbol{p}} - 2\lambda\mathbf{I}_N + (\boldsymbol{\mu}_{1..N} - \boldsymbol{\mu}_{N+1..2N}) - (\boldsymbol{\nu}_{1..N} - \boldsymbol{\nu}_{N+1..2N})] + b,$$

where $\tilde{p}_i = 2u(\mathbf{x}_t) - 2y_i - c$. So

$$\begin{aligned} \frac{\partial f(\mathbf{x}_t)}{\partial u(\mathbf{x}_t)} &= -\frac{2}{2}\boldsymbol{\gamma}(\mathbf{x}_t)^T\mathbf{K}^{-1}\mathbf{1}_N, \qquad (6.6) \\ &< 0. \quad \text{(by Eq. (6.3) and Eq. (6.2))} \end{aligned}$$

Therefore, in the fixed margin setting cases, given condition of Eq. (6.2), the decision function is a monotone decreasing function to the up margin. ■

From above theorem, increasing the up margin, $u(\mathbf{x})$, will produce a smaller predicted value. By using the DMAE as in Eq. (4.10) to measure the downside risk, we obtain the following corollary:

Corollary 6 If the condition in Eq. (6.2) is valid and the margin setting is FASM or FAAM, increasing the up margin, $u(\mathbf{x})$, will reduce the predictive downside risk or keep it to vanish.

Proof: Suppose that there are two margin settings in the fixed margin cases, FASM or FAAM. $u^1(\mathbf{x})$ and $d^1(\mathbf{x})$ are the up margin and down margin to the first fixed margin setting respectively. $u^2(\mathbf{x})$ and $d^2(\mathbf{x})$ are the up margin and down margin to the other fixed margin setting respectively. Here we assume $u^1(\mathbf{x}) + d^1(\mathbf{x}) = u^2(\mathbf{x}) + d^2(\mathbf{x})$ and $u^1(\mathbf{x}) < u^2(\mathbf{x})$.

Using these margin settings, we can obtain the corresponding predicted values p_j^1, $p_j^2, j = 1, \ldots, m$, where m is the number of testing data. We can calculate the corresponding DMAE as DMAE^1 and DMAE^2 respectively.

Since $u^1(\mathbf{x}) < u^2(\mathbf{x})$, from Theorem 5, we know that $p_j^1 > p_j^2, j = 1, \ldots, m$. The respective relations of p_j^1, p_j^2 and a_j consist of three cases.

Case 1: If $\forall j,\ j \in \{1, \ldots, m\}$, such that $p_j^2 < p_j^1 < a_j$. From Definition 4, $\mathrm{DMAE}^1 = \mathrm{DMAE}^2 = 0$.

Case 2: If there is $j_0,\ j_0 \in \{1, \ldots, m\}$, such that $p_{j_0}^2 < a_{j_0} < p_{j_0}^1$. From Definition 4, we know that DMAE^1 has at least one more term than DMAE^2, so we have $\mathrm{DMAE}^1 > \mathrm{DMAE}^2$.

Case 3: If $\forall j,\ j \in \{1, \ldots, m\}$, such that $a_j < p_j^2 < p_j^1$. From Definition 4, $\mathrm{DMAE}^1 > \mathrm{DMAE}^2$.

Therefore, all the cases make our conclusions. ■

6.2 Algorithm

In the following, we propose an algorithm to test the validity of the condition of Eq. (6.2),

Algorithm 1: Detective Algorithm

Construct the kernel function matrix $\mathbf{K}$;
Calculate the determinant of $\mathbf{K}$, $|\mathbf{K}|$;
valid = true;
for *i = 1 to N* **do**
  Substitute the values of i^{th} row of $\mathbf{K}$ to 1, form a new matrix $\mathbf{K'}_{\mathrm{i}}$;
  Calculate the determinant of $\mathbf{K'}_{\mathrm{i}}$, $|\mathbf{K'}_{\mathrm{i}}|$;
  if $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| < 0$, valid = false;
return valid;

In order to state that the above algorithm can check the validity of the condition of Eq. (6.2), we propose the following theorem.

Theorem 7

$$\forall i = 1, \ldots, N, \quad \mathrm{sgn}(d_i) = \mathrm{sgn}(|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}|),$$

where $\mathrm{sgn}(\cdot)$ denotes the sign of the value.

Proof: Let $\mathbf{K} = (k_{ij})$, then the adjoint of $\mathbf{K}$ is $adj\,\mathbf{K} = (K_{ij})^T$, where K_{ij} is the cofactor of k_{ij}.

If the determinant of $\mathbf{K}$ is not equal to 0, i. e. $|\mathbf{K}| \neq 0$, then $\mathbf{K}$ is non-singular and

$$\mathbf{K}^{-1} = \frac{1}{|\mathbf{K}|}\, adj\,\mathbf{K}.$$

From Eq. (6.1),

$$\begin{aligned}
(d_1, \ldots, d_N)^T &= \mathbf{K}^{-1}\,(1, \ldots, 1)^T \\
&= \frac{1}{|\mathbf{K}|}\, adj\,\mathbf{K}\,(1, \ldots, 1)^T \\
&= \frac{1}{|\mathbf{K}|} \begin{pmatrix} K_{11} & K_{21} & \ldots & K_{N1} \\ K_{12} & K_{22} & \ldots & K_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ K_{1N} & K_{2N} & \ldots & K_{NN} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.
\end{aligned}$$

While the determinant of $\mathbf{K}'_{\mathrm{i}}$ is

$$\begin{aligned}
|\mathbf{K}'_{\mathrm{i}}| &= \left| \begin{pmatrix} k_{11} & k_{12} & \ldots & k_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \ldots & k_{NN} \end{pmatrix} \right| \leftarrow i^{th} \text{ row} \\
&= \sum_{j=1}^{N} K_{ij} \cdot 1.
\end{aligned}$$

So,

$$\begin{aligned}
(d_1, \ldots, d_N)^T &= \frac{1}{|\mathbf{K}|}\,(|\mathbf{K}'_1|, |\mathbf{K}'_2|, \ldots, |\mathbf{K}'_{\mathrm{N}}|)^T \\
|\mathbf{K}|^2 \times (d_1, \ldots, d_N)^T &= |\mathbf{K}| \times (|\mathbf{K}'_1|, |\mathbf{K}'_2|, \ldots, |\mathbf{K}'_{\mathrm{N}}|)^T \\
|\mathbf{K}|^2 \times d_i &= |\mathbf{K}| \times |\mathbf{K}'_{\mathrm{i}}|, \quad i = 1, \ldots, N.
\end{aligned}$$

Therefore, we have

$$\mathrm{sgn}(d_i) = \mathrm{sgn}(|\mathbf{K}|^2 \times d_i) = \mathrm{sgn}(|\mathbf{K}| \times |\mathbf{K}'_{\mathrm{i}}|).$$

■

6.3 Experiments

We also do the following experiments to test our theoretical results. The experimental data are all the data sets used in [96, 98]:

HSI01: the daily closing prices of Hong Kong's HSI from January 15, 2001 to June 19, 2001, a total of 104 days' data;

HSI98-00: the daily closing prices of HSI from January 2, 1998 to December, 29, 2000, a total of three years' data;

DJIA98-00: the closing daily prices of DJIA from January 2, 1998 to December, 29, 2000, totally three years' data.

The corresponding training and testing periods are the same in [96, 98] and we list it again in Table 6.1.

Table 6.1: Experimental data description

Indices	Training Period	Testing Period
HSI01	15-Jan.-2001 $\sim$ 22-May-2001	23-May-2001 $\sim$ 19-Jun.-2001
HSI98-00	02-Jan.-1998 $\sim$ 04-Jul.-2000	05-Jul.-2000 $\sim$ 29-Dec.-2000
DJIA98-00	02-Jan.-1998 $\sim$ 29-Jun.-2000	30-Jun.-2000 $\sim$ 29-Dec.-2000

This experimental procedure consists of three main parts: at first, we normalize the prices by $t_i = \frac{a_i - a_{low}}{a_{high} - a_{low}}$, where t_i is the normalized price at day i, a_i is the actual price of the stock at day i, a_{low} and a_{high} are the correspondingly minimum and maximum prices in the training data respectively. Then, we use the training data to run the detective algorithm, Algorithm 1. Finally, we apply the SVR algorithm, which is modeled as $p_{n\_i} = f(\mathbf{x}_i)$, where $\mathbf{x}_i = (t_{i-4}, t_{i-3}, t_{i-2}, t_{i-1})$, to train the training data and then, we use the testing data to obtain the normalized predicted values. After unnormalizing these values p_{n_i}, we obtain the corresponding predicted prices p_i.

Detective Algorithm

When running Algorithm 1, we use different $\beta = 2^{\tau}$, τ=-15, ... , 15, to construct the kernel matrix $\mathbf{K}$ and test the validity of the condition in Eq. (6.2). We show the results in Table 6.2, 6.3, 6.4 for data sets HSI01, HSI98-00 and DJIA98-00 respectively. From Table 6.2, we find that there are three kinds of results for data set HSI01: when τ ranges from -15 to 3, $\forall i = 1, \ldots, N$, $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| = 0$; when τ is in 4 and 9, non of $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}|$ equals 0, but there are some i, such that $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| < 0$, i. e. some d_i is less than 0; when τ is from 10 to 15, $\forall i = 1, \ldots, N, |\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| > 0$, i. e. all d_i are greater than 0 and satisfy the condition in Eq. (6.2); N here is equal to the size of training set in HSI01. For data HSI98-00 and DJIA98-00, they also have similar results: when τ is from -15 to 9, $\forall i = 1, \ldots, N$, all $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| = 0$; when τ is within 10 and 13, no $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| = 0$, but there are some i, such that $|\mathbf{K}| \times |\mathbf{K'}_{\mathrm{i}}| < 0$, i. e. some $d_i < 0$; when τ is 14 or 15, all $d_i > 0$, here N refers to the size of training set in HSI98-00, DJIA98-00 respectively.

Table 6.2: Validated results for HSI01

case	I	II	III
τ	-15 $\sim$ 3	4 $\sim$ 9	10 $\sim$ 15
β	$2^{-15} \sim 2^{3}$	$2^{4} \sim 2^{9}$	$2^{10} \sim 2^{15}$
valid	true	false	true

Table 6.3: Validated results for HSI98-00

case	I	II	III
τ	-15 $\sim$ 9	10 $\sim$ 13	14 $\sim$ 15
β	$2^{-15} \sim 2^{9}$	$2^{10} \sim 2^{13}$	$2^{14} \sim 2^{15}$
valid	true	false	true

Table 6.4: Validated results for DJIA98-00

case	I	II	III
τ	-15 $\sim$ 9	10 $\sim$ 13	14 $\sim$ 15
β	$2^{-15} \sim 2^{9}$	$2^{10} \sim 2^{13}$	$2^{14} \sim 2^{15}$
valid	true	false	true

SVR Algorithm

We perform experiments using SVR algorithm with $\beta = 2^{\tau}$, τ=-15, ... , 15, $C = 1$ and the width of margin being 0.01, i. e., $u(\mathbf{x}) + d(\mathbf{x}) = 0.01$, each increment is 0.0025. From the experimental results, we find that all $DMAE$ decrease with the increase of up margin for these three datasets. We just list partial results in the following Table 6.5, 6.6, 6.7, 6.8, 6.9, 6.10 and Table 6.11, 6.12, 6.13.

For data set HSI01, we only list the results of $\beta = 2^{-2}, 1, 2^{4}, 2^{9}$ and $\beta = 2^{10}, 2^{12}$ in Table 6.5, Table 6.6 and Table 6.7 respectively. These tables consist of two tables with the results on the left and two figures corresponding to the relation of up margin and DMAE on the right. The β_{S} selected in Table 6.5, Table 6.6 and Table 6.7 to represent the results in Table 6.2 of case I, case II and case III respectively. From these results we can see that as the up margin increases, DMAE decreases gradually (all figures clearly show this phenomenon). We also note that condition in Eq. (6.2) is only a sufficient condition. When there are some $d_i < 0$, from Eq. (6.6), we know that it is possible that $\boldsymbol{\gamma}(\mathbf{x}_t)^T\mathbf{K}^{-1} > 0$ and $\frac{\partial f(\mathbf{x}_t)}{\partial u(\mathbf{x}_t)} < 0$, this also can derive the result of Theorem 5. When $|\mathbf{K}| \times |\mathbf{K}'_{\mathrm{i}}| = 0$, it may $|\mathbf{K}|$ equal to 0. In this case, $\mathbf{K}$ is singular and we do not prove this case in Theorem 5.

For data set HSI98-00 and data set DJIA98-00, we select $\beta = 2^{-2}$, 1, 2^{10}, 2^{13}, 2^{14}, 2^{15} to represent the corresponding results in Table 6.3 and Table 6.4 respectively. The corresponding results are shown in Table 6.8, Table 6.9,

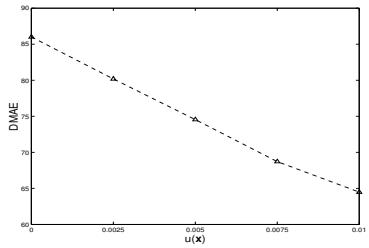
Table 6.10 for HSI98-00 and in Table 6.11, Table 6.12, Table 6.13 for DJIA98-00 respectively. All of these results meet our results in Theorem 5. We also note that when $\beta = 2^{13}$, 2^{14}, or 2^{15}, since all of the predicted values are under-predicted, i. e. less than the actual values, we have all DMAE=0.

6.4 Discussions

From the experimental results, we can see that as the up margin increases, the predictive values tend to smaller. This actually meets our theoretical results. The proposed algorithm also can be applied to the training data set to see the changing trend of predictive value before prediction. That is to say, if we use FASM or FAAM as the method to predict the stock price, we can use the above algorithm to check the validity of the condition. And then, we know the changing trend of predictive values when we adjust the up (down) margin. Therefore, we can reduce/increase the predictive downside risk.

However, there are some lacks in our theoretical results. First, we do not consider the case when $\mathbf{K}$ is singular. The experimental results also indicated in this case, the phenomenon, the increase of up margin leads to smaller predictive value, is also true. Second, in the proof of Theorem. 5, we just prove the cases of FASM and FAAM. We do not consider the cases of NASM and NAAM. But in our opinions, from the results of setting momentum term in the margin, for non-fixed margin settings, this phenomenon may be also true. Third, the experimental results indicated that although the condition 6.2 is not met, e. g. case I and case II, the phenomenon is still true. This means that we still can loose the condition to get the same theorem.

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{-2}	0	0.01	86.01
	0.0025	0.0075	80.18
	0.005	0.005	74.54
	0.0075	0.0025	68.71
	0.01	0	64.48



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
1	0	0.01	89.76
	0.0025	0.0075	83.20
	0.005	0.005	77.56
	0.0075	0.0025	71.93
	0.01	0	66.41

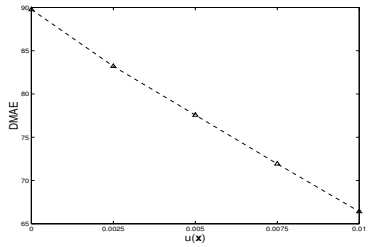
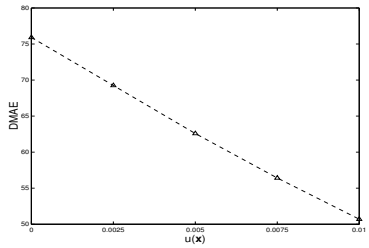


Table 6.5: DMAE for HSI01 of case I

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^4	0	0.01	75.91
	0.0025	0.0075	69.24
	0.005	0.005	62.58
	0.0075	0.0025	56.41
	0.01	0	50.70



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^9	0	0.01	359.64
	0.0025	0.0075	350.41
	0.005	0.005	341.19
	0.0075	0.0025	331.96
	0.01	0	322.73

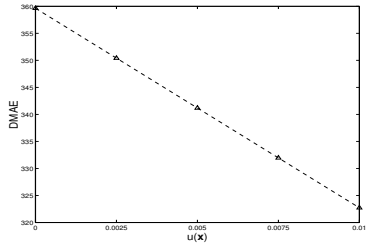
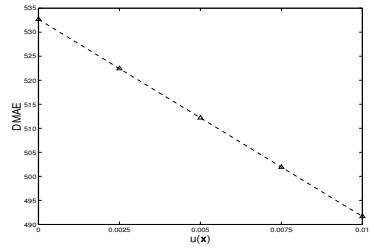


Table 6.6: DMAE for HSI01 of case II

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{-2}	0	0.01	532.67
	0.0025	0.0075	522.42
	0.005	0.005	512.17
	0.0075	0.0025	501.92
	0.01	0	491.67



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
1	0	0.01	627.41
	0.0025	0.0075	617.16
	0.005	0.005	606.91
	0.0075	0.0025	596.66
	0.01	0	586.41

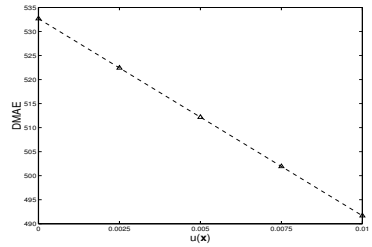
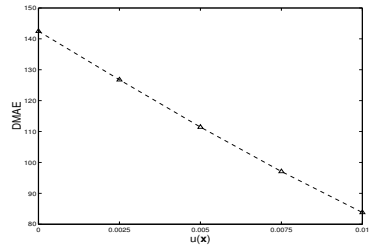


Table 6.7: DMAE for HSI01 of case III

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{-2}	0	0.01	142.48
	0.0025	0.0075	126.76
	0.005	0.005	111.44
	0.0075	0.0025	97.06
	0.01	0	83.30



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
1	0	0.01	139.82
	0.0025	0.0075	120.74
	0.005	0.005	108.20
	0.0075	0.0025	92.66
	0.01	0	82.60

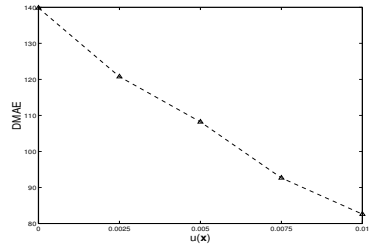
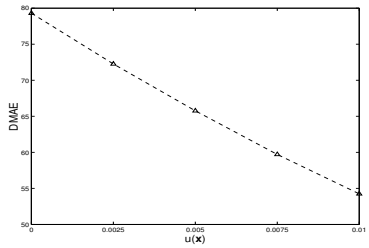


Table 6.8: DMAE for HSI98-00 of case I

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{10}	0	0.01	79.30
	0.0025	0.0075	76.26
	0.005	0.005	65.76
	0.0075	0.0025	59.70
	0.01	0	54.24



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{13}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0

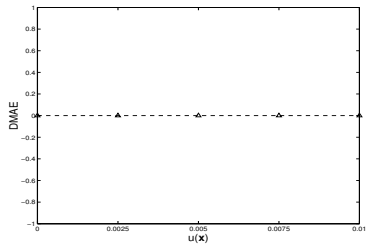
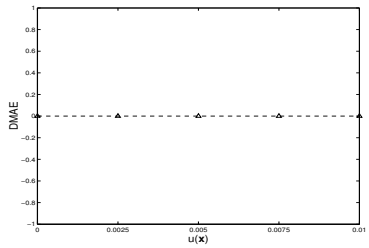


Table 6.9: DMAE for HSI98-00 of case II

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{14}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{15}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0

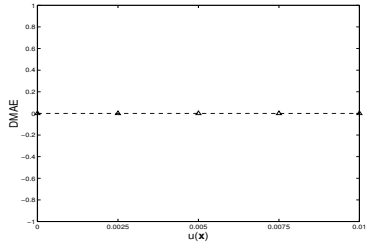
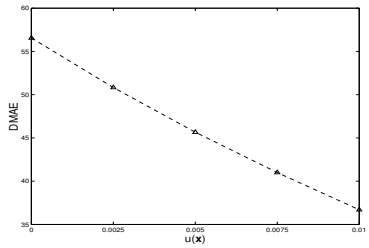


Table 6.10: DMAE for HSI98-00 of case III

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{-2}	0	0.01	56.56
	0.0025	0.0075	50.82
	0.005	0.005	45.66
	0.0075	0.0025	40.99
	0.01	0	36.69



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
1	0	0.01	56.09
	0.0025	0.0075	50.42
	0.005	0.005	45.08
	0.0075	0.0025	40.34
	0.01	0	36.00

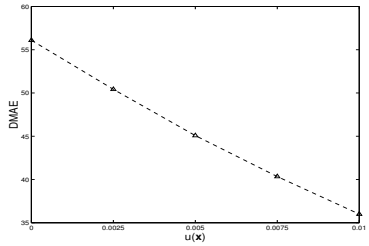
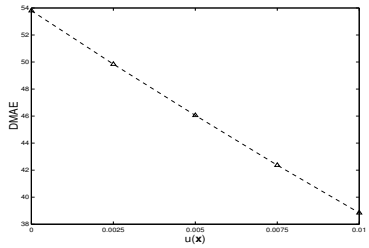


Table 6.11: DMAE for DJIA98-00 of case I

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{10}	0	0.01	53.80
	0.0025	0.0075	49.84
	0.005	0.005	46.05
	0.0075	0.0025	42.37
	0.01	0	38.84



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{13}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0

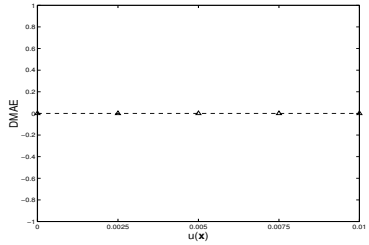
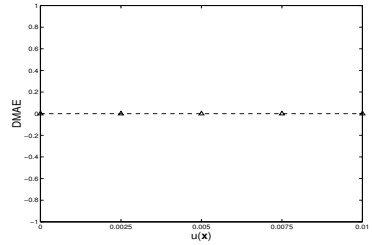


Table 6.12: DMAE for DJIA98-00 of case II

β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{14}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0



β	$u(\mathbf{x})$	$d(\mathbf{x})$	$DMAE$
2^{15}	0	0.01	0
	0.0025	0.0075	0
	0.005	0.005	0
	0.0075	0.0025	0
	0.01	0	0

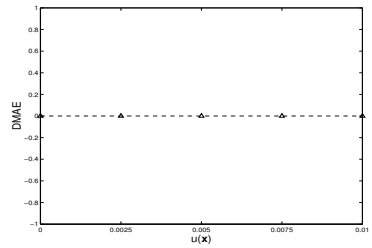


Table 6.13: DMAE for DJIA98-00 of case III

Chapter 7

Conclusion

In this thesis, we propose to vary the margin in the ε insensitive loss function, and then we extend it to a general ε loss function. After applying it to financial prediction tasks, we obtain the following results: (a) By varying the width of margin and with a momentum, we can reflect the volatility of the stock market and capture the up/down trend of the stock market. Adding these information into the margin setting is helpful for the prediction of stock market prices. (b) GARCH method can also be applied in the setting of margin. (c) In the fixed margin cases, if the sufficient condition, i.e. Eq. (6.2), is true, the predictive value is monotone to the up margin and hence, we may reduce the predictive downside risk or keep it zero by increasing the up margin.

A reliable prediction algorithm to predict the stock market implies a better system which can help us make more money than other investment strategies. However, it is hard to fulfil this objective. Although the general mechanisms underlying the evolution of stock market prices elude us, we believe there is still room for cautious optimism about the use of exploratory statistical modeling the financial markets. At the very least, these statistical models may provide a systematic way to monitor the stock market from huge amount of information on the stock market.

Apart from the financial applications, our idea of asymmetrical margin settings as in Chapter 6 may also be applied in other applications, e. g. biased

classification. It may be similar to the weighted SVM in [62], which used larger value of C (cost of the error) to penalize highly reliable class data points and smaller value of C for less confidence class. Our idea of asymmetrical margins in SVC is to find out the classification hyperplane by standard SVC first, and to push the hyperplane away from the highly confidence class simply.

Further improvements can be made in our model. First, we should find ways to optimize the values of some parameters, such as the cost of error, C; the RBF kernel parameter, β. Second, our model assumes that the information of the whole stock market is captured in the price values and there is a non-linear relation between current stock price and the previous four days' stock prices. However, the actual stock market is very complicated; there are still many useful information for the prediction of stock markets other than prices. Other information, e. g. volumes, may also be added in the input vector $\mathbf{x}$. It may improve the predictive performance.

Appendix A

Basic Results for Solving SVR

A.1 Dual Theory

Let X^0 be a nonempty open set in $\mathbb{R}^n$, and let f be a numerical function defined on X^0, i.e., $f : X^0 \longmapsto \mathbb{R}$, $\mathbf{g}$ be respectively a p-dimensional vector function defined on X^0, i.e., $\mathbf{g} : X^0 \longmapsto \mathbb{R}^p$.

The primal (minimization) problem is defined as follows,

Definition 8 The (primal) Minimization Problem (MP) [55, 5],

Find an $\bar{\mathbf{x}}$, if it exists, such that

$$f(\bar{\mathbf{x}}) = \min_{\mathbf{x} \in X} f(\mathbf{x}), \quad \bar{\mathbf{x}} \in X = \{x \,|\, x \in X^0, \mathbf{g}(\mathbf{x}) \leq \mathbf{0}\} \tag{A.1}$$

The corresponding dual (maximization) problem is,

Definition 9 The Dual (maximization) Problem (DP) [55, 5],

Let f and $\mathbf{g}$ be differentiable on X^0. Find an $\hat{\mathbf{x}}$ and a $\hat{\mathbf{u}} \in \mathbb{R}^p$, if they exist, such that

$$\begin{aligned} L(\hat{\mathbf{x}}, \hat{\mathbf{u}}) &= \max_{(\mathbf{x},\mathbf{u}) \in Y} L(\mathbf{x}, \mathbf{u}), \\ (\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in Y &= \{(\mathbf{x}, \mathbf{u}) \,|\, \mathbf{x} \in X^0, \mathbf{u} \in \mathbb{R}^p, \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \mathbf{u} \geq \mathbf{0}\}, \\ L(\mathbf{x}, \mathbf{u}) &= f(\mathbf{x}) + \mathbf{u}^T \mathbf{g}(\mathbf{x}). \end{aligned} \tag{A.2}$$

The function L is usually called Lagrange function or Dual function.

Theorem 10 Wolfe's duality theorem [55]

Let X^0 be an open set in $\mathbb{R}$, let f and $\mathbf{g}$ be differentiable and convex on X^0, let $\bar{\mathbf{x}}$ be the solution of Eq. (A.1), and let $\mathbf{g}$ satisfy the Kuhn-Tucker conditions. Then there exists a $\bar{\mathbf{u}} \in \mathbb{R}^p$ such that $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ be the solution of Eq. (A.2) and $f(\bar{\mathbf{x}}) = L(\bar{\mathbf{x}}, \bar{\mathbf{u}})$.

Proof: Since $\mathbf{g}$ satisfy the Kuhn-Tucker conditions, there exist a $\bar{\mathbf{u}} \in \mathbb{R}^p$, such that $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ satisfies the Kuhn-Tucker conditions

$$\begin{aligned} \boldsymbol{\nabla} f(\bar{\mathbf{x}}) + \bar{\mathbf{u}}^T \boldsymbol{\nabla} \mathbf{g}(\bar{\mathbf{x}}) &= \mathbf{0}, \\ \bar{\mathbf{u}}^T \mathbf{g}(\bar{\mathbf{x}}) &= 0, \\ \mathbf{g}(\bar{\mathbf{x}}) &\leq \mathbf{0}, \\ \bar{\mathbf{u}} &\geq \mathbf{0}. \end{aligned}$$

Hence

$$(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \in Y = \{(\mathbf{x}, \mathbf{u}) | \mathbf{x} \in X^0, \mathbf{u} \in \mathbb{R}^p, \boldsymbol{\nabla} f(\mathbf{x}) + \mathbf{u}^T \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}) = \mathbf{0}, \mathbf{u} \geq \mathbf{0}\}.$$

Now let $(\mathbf{x}, \mathbf{u})$ be an arbitrary element of the set Y. Then

$$\begin{aligned} L(\bar{\mathbf{x}}, \bar{\mathbf{u}}) - L(\mathbf{x}, \mathbf{u}) &= f(\bar{\mathbf{x}}) - f(\mathbf{x}) + \bar{\mathbf{u}}^T \mathbf{g}(\bar{\mathbf{x}}) - \mathbf{u}^T \mathbf{g}(\mathbf{x}) \\ &\geq \boldsymbol{\nabla} f(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x}) - \mathbf{u}^T \mathbf{g}(\mathbf{x}) \\ &\qquad (\because f \text{ is convex and } \bar{\mathbf{u}}^T \mathbf{g}(\bar{\mathbf{x}}) = \mathbf{0}) \\ &\geq \boldsymbol{\nabla} f(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x}) + \mathbf{u}^T [-\mathbf{g}(\bar{\mathbf{x}}) + \boldsymbol{\nabla} \mathbf{g}(\mathbf{x})(\bar{\mathbf{x}} - \mathbf{x})] \\ &\qquad (\because \mathbf{g} \text{ are convex and } \mathbf{u} \geq \mathbf{0}) \\ &= [\boldsymbol{\nabla} f(\mathbf{x}) + \mathbf{u}^T \boldsymbol{\nabla} \mathbf{g}(\mathbf{x})](\bar{\mathbf{x}} - \mathbf{x}) - \mathbf{u}^T \mathbf{g}(\bar{\mathbf{x}}) \\ &= -\mathbf{u}^T \mathbf{g}(\bar{\mathbf{x}}) \qquad (\because \boldsymbol{\nabla} f(\mathbf{x}) + \mathbf{u}^T \boldsymbol{\nabla} \mathbf{g}(\mathbf{x}) = \mathbf{0}) \\ &\geq 0 \qquad (\because \mathbf{u} \geq \mathbf{0} \text{ and } \mathbf{g}(\bar{\mathbf{x}}) \leq \mathbf{0}) \end{aligned}$$

Hence

$$L(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \max_{(\mathbf{x}, \mathbf{u}) \in Y} L(\mathbf{x}, \mathbf{u}) \quad (\bar{\mathbf{x}}, \bar{\mathbf{u}}) \in Y.$$

Since $\bar{\mathbf{u}}^T\mathbf{g}(\bar{\mathbf{x}}) = 0$

$$L(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = f(\bar{\mathbf{x}}) + \bar{\mathbf{u}}^T\mathbf{g}(\bar{\mathbf{x}}) = f(\bar{\mathbf{x}}) \text{ concludes the proof.}$$

■

A.2 Standard Method to Solve SVR

In the following, i indicates $1, \ldots, N$. $(*)$ indicates the variable with and without asterisk.

Definition 11 (Primal Optimization Problem of SVR)

$$\min_{\mathbf{w},b,\boldsymbol{\xi}^{(*)}} \quad \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{A.3}$$

subject to

$$\begin{aligned} y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - b &\leq \varepsilon + \xi_i, \\ \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b - y_i &\leq \varepsilon + \xi_i^*, \\ \xi_i^{(*)} &\geq 0. \end{aligned}$$

Solution 12 At first, we construct the Lagrange function as,

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\xi}^{(*)}) &= \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y_i + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle + b) \\ &- \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}_i)\rangle - b) - \sum_{i=1}^{N}(\mu_i\xi_i + \mu_i^*\xi_i^*), \end{aligned} \tag{A.4}$$

where $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\mu}, \boldsymbol{\mu}^* \geq \mathbf{0}$, are the corresponding Lagrange multipliers.

At the saddle point, the differentiation of L with respect to $\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*$ is equal to zero. Therefore, we have,

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i) &= 0,\\
\frac{\partial L}{\partial b} &= \sum_{i=1}^{N}\alpha_i - \sum_{i=1}^{N}\alpha_i^* &= 0,\\
\frac{\partial L}{\partial \xi_i} &= C - \alpha_i - \mu_i &= 0,\\
\frac{\partial L}{\partial \xi_i^*} &= C - \alpha_i^* - \mu_i^* &= 0.
\end{aligned}$$

We can rewrite the above equations as,

$$\begin{aligned}
\mathbf{w} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i), \qquad \sum_{i=1}^{N}\alpha_i = \sum_{i=1}^{N}\alpha_i^*, \qquad (A.5)\\
\alpha_i^{(*)} \in [0, C].
\end{aligned}$$

Substituting Eq. (A.5) into Eq. (A.4) and applying the Dual Theory in Appendix A.1, we change the original primal optimization problem of SVR in Eq. (A.3) to the following QP problem,

$$\begin{aligned}
\min \Phi(\boldsymbol{\alpha}^{(*)}) &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle\boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}_j)\rangle \qquad (A.6)\\
&\quad + \sum_{i=1}^{N}(\varepsilon - y_i)\alpha_i + \sum_{i=1}^{N}(\varepsilon + y_i)\alpha_i^*,
\end{aligned}$$

subject to

$$\sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0, \quad \alpha_i^{(*)} \in [0, C].$$

Bibliography

[1] M. Aoki. *State Space Modeling of Time Series.* Springer-verlag; New York: Springer-Verlag, 2nd edition, 1990.

[2] G. M. De Athayde. Building a Mean-Downside Risk Portfolio Frontier. In F. A. Sortino and S. E. Satchell, editors, *Managing Downside Risk in Financial Markets: Theory, Practice and Implementation*, pages 194–211. Oxford, Boston:Butterworth-Heinemann, 2001.

[3] D. E. Baestaens. *Neural Network Solutions for Trading in Financial Markets.* London: Financial Times: Pitman Pub., 1994.

[4] I. S. Baird and H. Thomas. What Is Risk Anyway? Using and Measuring Risk in Strategic Management. In Richard A. Bettis and Howard Thomas, editors, *Risk, Strategy and Management*, pages 21–51. Greenwich, Conn.: JAI Press, 1990.

[5] M. S. Bazaraa. *Nonlinear Programming: Theory and Algorithms.* New York: Wiley, 2nd edition, 1993.

[6] K. Bennett and E. Bredensteiner. Duality and Geometry in SVM Classifiers. In P. Langley, editor, *Proc. of Seventeenth Intl. Conf. on Machine Learning*, pages 57–64, San Francisco, 2000. Morgan Kaufmann.

[7] Dimitri P. Bertsekas. *Nonlinear Programming.* Belmont, Mass.: Athena Scientific, 2nd edition, 1999.

[8] T. Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Econometrics*, 31:307–327, 1986.

[9] B. E. Boser, I. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Computational Learing Theory*, pages 144–152, 1992.

[10] G.E.P. Box and G.M. Jenkins. *Time-Series Analysis, Forecasting and Control*. San Francisco: Holden-Day, third edition, 1994.

[11] R. G. Brown. *Smoothing, Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs, N.J.: Prentice Hall, 1963.

[12] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[13] C. Burges and D. Crisp. Uniqueness of the SVM Solution. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 223–229, Cambridge, MA, 2000. MIT Press.

[14] C. Campbell. An Introduction to Kernel Methods. In R. J. Howlett and L. C. Jain, editors, *Radial Basis Function Networks: Design and Applications*, chapter 7, pages 155–192. Physica Verlag, Berlin., 2000.

[15] C. Campbell and N. Cristianini. Simple Learning Algorithms for Training Support Vector Machines, 1998. University of Bristol of Technical Report.

[16] Li Juan Cao, Kok Seng Chua, and Lim Kian Guan. c-Ascending Support Vector Machines for Financial Time Series Forecasting. In *International Conference on Computational Intelligence for Financial Engineering (CIFEr2003)*, pages 329–335, 2003.

[17] Li Juan Cao, Kok Seng Chua, and Lim Kian Guan. Combining KPCA with Support Vector Machine for Time Series Forecasting. In *International Conference on Computational Intelligence for Financial Engineering (CIFEr2003)*, pages 337–341, 2003.

[18] Siu-Ming Cha and Laiwan Chan. Trading Signal Prediction. In *International Conference on Neural Information Processing — ICONIP 2000*, pages 842–846, 2000.

[19] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a Library for Support Vector Machines (version 2.31), 2001.

[20] O. Chapelle and V. N. Vapnik. Model Selection for Support Vector Machines. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[21] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1-3):131–159, 2002.

[22] C. Chatfield. *The Analysis of Time Series: An Introduction.* Chapman and Hall, fifth edition, 1996.

[23] C. Chatfield. *Time-Series Forecasting.* Chapman and Hall/CRC, 2001.

[24] B. Cheng and D. M. Titterington. Neural Networks: A Review from a Statistical Perspective. *Statistical Science*, 9:2–54, 1994.

[25] V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory and Mehods.* Wiley, New York, 1998.

[26] C. Cortes and V. N. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

[27] D. J. Crisp and C. Burges. A Geometric Interpretation of ν-svm Classifiers. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[28] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines(and Other Kernel-based Learning Methods)*. Cambridge University Press, Cambridge, U.K.; New York, 2000.

[29] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. N. Vapnik. Support Vector Regression Machines. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 155–161. The MIT Press, 1997.

[30] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, London; New York, 1973.

[31] R. F. Engle., editor. *ARCH: Selected Readings*. Oxford; New York: Oxford University Press, 1995.

[32] R.F. Engle. Autoregresssive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflations. *Econometrica*, 50:987–1007, 1982.

[33] E.F. Fama. The Behavior of Stock Market Prices. *Journal of Business*, January:34–105, 1965.

[34] E.F. Fama. Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, May:383–417, 1970.

[35] A. Fan and M. Palaniswami. Selecting Bankruptcy Predictors Using a Support Vector Machine Approach. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference*, volume 6, pages 354 – 359. The MIT Press, 2000.

[36] T. Friess, N. Cristianini, and C. Campbell. The Kernel Adatron Algorithm: A Fast and Simple Learning Procedure for Support Vector Machine. In *Proceedings of 15th International Conference on Machine Learning*, 1998.

[37] J.B. Gao, S.R. Gunn, C.J. Harris, and M. Brown. A Probabilistic Framework for SVM Regression and Error Bar Estimation. *Machine Learning*, 46(1–3):71–89, March 2002.

[38] C. Gourieroux. *ARCH Models and Financial Applications*. Springer-Verlag, 1997.

[39] C. W. J. Granger and A. P. Andersen. *Introduction to Bilinear Time Series*. Göttingen: Vandenhoeck and Ruprecht, 1978.

[40] C. W. J. Granger and R. Joyeux. An Introduction to Long-Memory Time Series Models and Fractional Differencing. *Journal of Time Series Analysis*, 1, 1980.

[41] S. Gunn. Support Vector Machines for Classification and Regression. Technical Report NC2-TR-1998-030, Faculty of Engineering and Applied Science, Department of Electronics and Computer Science, University of Southampton, May 1998.

[42] A. C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge: Cambridge Univ. Press, 1989.

[43] S. Haykin. *Neural Networks : A Comprehensive Foundation*. Upper Saddle River, N.J.: Prentice Hall, 2nd edition, 1999.

[44] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55–67, 1970.

[45] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[46] T. Joachims. Making Large-Scale Support Vector Machine Learning Practical. In A. Smola B. Schölkopf, C. Burges, editor, *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA, 1998.

[47] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[48] E.M. Jordaan and G.F. Smits. Estimation of the Regularization Parameter for Support Vector Regression. In *The 2002 Internation Joint Conference on Neural Network, IJCNN'02*, volume 3, pages 2192–2197, 2002.

[49] Ingber L. Statistical Mechanics of Nonlinear Nonequilibrium Financial Markets: Applications to Optimized Trading. *Mathematical Computer Modelling*, 1996.

[50] J.-H. Lee and C.-J. Lin. Automatic Model Selection for Support Vector Machines, November 2000. Implementation available in looms.

[51] Y.-J. Lee and O.L. Mangasarian. RSVM: Reduced Support Vector Machines. In *the First SIAM International Conference on Data Mining*, 2001.

[52] K.-M. Lin and C.-J. Lin. A Study on Reduced Support Vector Machines. *IEEE Transactions on Neural Networks*, 2003. To appear.

[53] Andrew W. Lo and A. Craig MacKinlay. *A Non-Random Walk Down Wall Street*. Princeton University Press, 1999.

[54] B.G. Malkiel. *Efficient Market Hypothesis*. Macmillan, London, 1987.

[55] O. L. Mangasarian. *Nonlinear Programming*. Philadelphia: Society for Industrial and Applied Mathematics, 2nd edition, 1994.

[56] H. Markowitz. Portfolio Selection. *Journal of Finance*, 7:77–91, 1952.

[57] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines. In J. Principe, L. Giles, N. Morgan, and E. Wilson, editors, *IEEE Workshop on Neural Networks for Signal Processing VII*, pages 511–519. IEEE Press, 1997.

[58] K. R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting Time Series with Support Vector Machines. In W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, editors, *ICANN*, pages 999–1004. Springer, 1997.

[59] Ian T. Nabney. *Netlab: Algorithms for Pattern Recognition*. Springer, London; New York, 2002.

[60] D.F. Nicholls and A. Pagan. Varying Coefficient Eegression. In E.J. Hannan, P.R. Krishnaiah, , and M.M. Rao, editors, *Handbook of Statistics*, volume 5, pages 413–449, North Holland, Amsterdam, 1985.

[61] E. Osuna, R. Freund, and F. Girosi. Improved Training Algorithm for Support Vector Machines. In *NNSP'97*, 1997.

[62] E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: Training and applications. Technical Report AIM-1602, MIT, 1997.

[63] B. Schölkopf Pai-Hsuen Chen, Chih-Jen Lin. A Tutorial on ν-Support Vector Machines. Technical report, National Taiwan University, 2003.

[64] J. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

[65] M. B. Priestley. *Spectral Analysis and Time Series*. New York: Academic Press, London, 1981.

[66] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1:81–106, 1986.

[67] Baldev Raj and Aman Ullah. *Econometrics: A Varying Coefficients Approach*. New York: St. Martin's Press, 2nd edition, 1981.

[68] B.D. Ripley. Statistical Aspects of Neural Networks. In O.E.Barndorff-Nielsen, J.L. Jensen, and W.S.Kendall, editors, *Network and Chaos – Statistical and Probablistic Aspects*, pages 40–123, London, 1993. Chapman and Hall.

[69] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Support Vector Regression with Automatic Accuracy Control. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of ICANN'98 Perspectives in Neural Computing*, pages 111–116, Berlin, 1998. Spring.

[70] B. Schölkopf, P. Bartlett, A. Smola, and R. Williamson. Shrinking the Tube: A New Support Vector Regression Algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 330 – 336, Cambridge, MA, 1999. MIT Press.

[71] B. Schölkopf, C. Burges, and A. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, Cambridge, Massachusetts, 1999.

[72] B. Schölkopf, P. Y. Simard, A. Smola, and V. N. Vapnik. Prior Knowledge in Support Vector Kernels. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural information processings systems*, volume 10, pages 640–646, Cambridge, MA, 1998. MIT Press.

[73] B. Schölkopf and A. Smola, editors. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, Massachusetts, 2002.

[74] B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett. New Support Vector Algorithms. Technical Report NC2-TR-1998-031, GMD and Australian National University, 1998.

[75] Taylor S.J. *Modeling Fiancial Time Series*. J.Wiley & Sons, Chichester, 1994.

[76] A. Smola, N. Murata, B. Schölkopf, and K.-R. Müller. Asymptotically Optimal Choice of ε-Loss for Support Vector Machines. In *Proc. of Seventeenth Intl. Conf. on Artificial Neural Networks*, 1998.

[77] A. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.

[78] C. Bhattacharyya S.S. Keerthi, S.K. Shevade and K.R.K. Murthy. A Fast Iterative Nearest Point Algorithm for Support Vector Machine Classifier Design. Technical Report TR-ISL-99-03, Intelligent Systems Lab, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore, India, 2000.

[79] J. A. K. Suykens and J. Vandewalle. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[80] J. A. K. Suykens, J. Vandewalle, and B. De Moor. Optimal Control by Least Squares Support Vector Machines. *Neural Networks*, 14(1):23–35, 2001.

[81] Van Gestel T., Suykens J., Baestaens D., Lambrechts A., Lanckriet G., Vandaele B., De Moor B., and Vandewalle J. Financial Time Series Prediction Using Least Squares Support Vector Machines within the Evidence Framework. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks in Financial Engineering*, 12(4):809–821, 2001.

[82] E. H. Tay and L. J. Cao. Application of Support Vector Machines to Financial Time Series Forecasting. *Omega*, 29:309–317, 2001.

[83] H. Tong. *Non-Linear Time Series.* Clarendon Press, Oxford, 1990.

[84] T.B. Trafalis and H. Ince. Support Vector Machine for Regression and Applications to Financial Forecasting. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000)*, volume 6, pages 348–353. IEEE, 2000.

[85] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley, October 2001.

[86] George Tsibouris and Matthew Zeidenberg. Testing the Efficient Markets Hypothesis with Gradient Descient Algorithms. In A.Refenes, editor, *Neural Networks in the Capital Markets*. John Wiley and Sons, 1995.

[87] V. N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. (in Russian), Nauka, Moscow, 1979.

[88] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[89] V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[90] V. N. Vapnik and O. Chapelle. Bounds on Error Expectation for Support Vector Machines. *Neural Computation*, 12(9):2013–2036, 2000.

[91] V. N. Vapnik, S. Golowich, and A. Smola. Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. In M. Mozer, M. Jordan, and T. Petshe, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 281–287, Cambridge, MA, 1997. MIT Press.

[92] S. S. Keerthi W. Chu and C. J. Ong. Bayesian Inference in Support Vector Regression. Technical Report CD-01-15, Control Division, Department of Mechanical Engineering, National University of Singapore, 2001a.

[93] Benjamin W. Wah and Minglun Qian. Constrained Formulations and Algorithms for Stock-Price Predictions Using Recurrent FIR Neural Networks. In *AAAI/IAAI*, pages 211–216, 2002.

[94] C.J.C.H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, England, 1989.

[95] M West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. New York: Springer-Verlag, 1997.

[96] Haiqin Yang, Laiwan Chan, and Irwin King. Support Vector Machine Regression for Volatile Stock Market Prediction. In Hujun Yin, Nigel Allinson, Richard Freeman, John Keane, and Simon Hubbard, editors, *Intelligent Data Engineering and Automated Learning — IDEAL 2002*, volume 2412 of *LNCS*, pages 391–396. Springer, 2002.

[97] Haiqin Yang, Laiwan Chan, and Irwin King. Margin Settings in Support Vector Regression for the Stock Market Prediction, 2003. To be submitted.

[98] Haiqin Yang, Irwin King, and Laiwan Chan. Non-fixed and Asymmetrical Margin Approach to Stock Market Prediction Using Support Vector Regression. In *International Conference on Neural Information Processing — ICONIP 2002*, page cr1968, 2002.

[99] Zheng Rong Yang. Support Vector Machines for Company Failure Prediction. In *International Conference on Computational Intelligence for Financial Engineering (CIFEr2003)*, pages 47–54, 2003.