Financial Time Series Prediction Using Support Vector Machines

Kai Chen

Abstract

This paper mainly reports the work of Van Gestel *et al*: *Financial Time Series Prediction Using Least Squares Support Vector Machines Within the Evidence Framework* [1]. First, we present some background knowledge on Support Vector Machine (SVM) and Bayesian inference framework. Then the methodology of their work is reported. Finally, an application of the model is given.

Keywords: Financial time series prediction, LS-SVM, Bayesian inference

1. Introduction

Time series prediction is a dynamic research area in financial economics. A good prediction is important for the formulation of economic policies (e.g. bank lending rate) and the construction of financial models (e.g. investment strategies). The main process of time series prediction is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model which describes the inherent structure of the series. Time series forecasting thus can be termed as the act of predicting the future by understanding the past. Mathematically stated:

$$\hat{x}(t + \Delta_t) = f(x(t - a), x(t - b), x(t - c), \dots) \tag{1}$$

where, in this simple example [2], $\hat{x}$ is the predicted value of a discrete time series x. The object of prediction is to find a function $f(x)$ such that $\hat{x}$ at a future time point is *unbiased* and *consistent*. It should be mentioned that another measure of a predictor's goodness is *efficiency* which is introduced in [3].

However, prediction is never an easy task especially for financial field since the market data are highly unstable and full of noises. In addition, the high frequency [4] and long memory [5] properties of stock market require more complex methods to model.

In general, time series models can be divided into two classes: stochastic models and supervised machine learning based models. In stochastic field, numerous classical models, such as ARIMA (Autoregressive Integrated Moving Average), are proposed, both linear [6] and nonlinear [7].

Email address: `kai.chen@rutgers.edu` (Kai Chen)

Recently, as more data are available with powerful analysis machines, a substantial amount of research works have been carried out towards the application of machine learning approaches for time series modeling and prediction. As a prominent representative of multilayer perceptrons (MLPs), artificial neural networks (ANNs) have attracted increasing attentions in the domain of time series forecasting [8]. However, the practical design of MLSs suffers from drawbacks like the non-convex optimization problem and the choice of the number of hidden units. A major breakthrough in the area of time series forecasting occurred with the development of Vapniks support vector machine (SVM) concept [9].

In SVM, the regression problem is formulated and represented as a convex quadratic programming (QP) problem [10]. Following this idea, numerous SVM forecasting models have been developed by researchers. In this paper, we mainly present the use of an LS-SVM combined with a Bayesian evidence framework proposed by Van Gestel *et al* [1], which is the basis of most following works that involve more complex methods.

2. Preliminary

2.1. Least Squares Support Vector Machines (LS-SVMs)

The Support Vector Machine (SVM), developed by Vapnik [9] and others in 1995, is used for many machine learning tasks such as pattern recognition, object classification, and in the case of time series prediction, regression analysis. Similar to most of the machine learning methods, SVM gives a prediction after it is trained by observed data. This makes SVM different from more traditional time series prediction methodologies in the sense there is no "model" since the data drives the prediction.

Basically, if the data is not linear in its "input" space, the SVM regression maps the inputs into a higher dimensional "feature" space in which a linear regression is constructed by minimizing an appropriate cost function. Using Mercers theorem [11], the regression is obtained by solving a finite dimensional QP problem in the dual space. This avoids explicit knowledge of the high dimensional mapping and uses only the related kernel functions.

To illustrate this idea more concretely, consider a nonlinear regression $y_i = f(x_i) + e_i$ in the following form:

$$y_i = w^T \varphi(x_i) + b + e_i \tag{2}$$

where $w \in \mathbf{R}^{n_f}$ is the weight vector; $b \in \mathbf{R}$ is the constant term; e_i is additive noise. For financial time series, the output $y_i \in \mathbf{R}$ is typically a return of an asset or exchange rate, or some measure of the volatility at the time index i. The input vector $x_i \in \mathbf{R}^n$ may consist of lagged returns, volatility measures and economic explanatory variables. The mapping $\varphi(\cdot) : \mathbf{R}^n \mapsto \mathbf{R}^{n_f}$ is a nonlinear function that maps the input vector x into a higher dimensional feature space.

According to Mercer's theorem, $K(x_i, x) = \varphi(x_i)^T \varphi(x)$ is applied to relate the function $\varphi(\cdot)$ with the symmetric and positive definite kernel function K. Then with the introduction of a so called ε-sensitive loss (or cost) function, we can get the optimal weights by solving the following QP problem:

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} L(y(i), f(x(i), w)) \tag{3}$$

where $y(i)$ is the "truth" date (training set) and $L(\cdot)$ is the loss function which is given as:

$$L(y(i), f(x(i), w)) = \begin{cases} |y(i) - f(x(i), w)| - \varepsilon & if|y(i) - f(x(i), w)| \leq \varepsilon \\ 0 & otherwise \end{cases} \tag{4}$$

It is inherently assumed that a function $f(x)$ actually exists and the optimization problem is feasible. However, errors may have to be accepted to make the problem feasible. To account for errors, some slack variables are typically introduced, which makes the QP problem subject to a series of inequality conditions. Readers are referred to [10] to see how to set up the conditions for the *primal* problem that is given by Equation (3).

Usually, this QP problem is solved by using Lagrange multipliers and forming the dual optimization problem given by:

$$\max -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x(i), x(j)\rangle - \varepsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{N} y(i)(\alpha_i - \alpha_i^*) \tag{5}$$

$$\text{Subject to} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0; \alpha_i, \alpha_i^* \in [0, C] \tag{6}$$

Those input data points with non-zero Lagrange multipliers α are defined as *Support Vectors*. As we can see, the optimal weights w associated with having non-zero Lagrange multipliers is typically less than the entire data set. This means one does not need the entire data set to define $f(x)$. The sparseness of this solution is one of several advantages of using this methodology. Thus, the output prediction can be given by:

$$y_i = \sum_{i=1}^{N} \alpha_i K(x_i, x) + b \tag{7}$$

Notice that the prediction is made by a linear combination of weighted kernel functions. Here we present some well known kernels, K, used in SVM literature:

- The *Linear Kernel*: $K(x_i, x) = x_i^T x$
- The *Polynomial Kernel*: $K(x_i, x) = (x_i^T x + 1)^d$
- The *Radial Basis Function (RBF) Kernel*: $K(x_i, x) = exp(-||x - x_i||^2/2\sigma^2)$

The architecture of this whole process is given below in Figure 1 (reproduced here from Figure 1 in [2]):

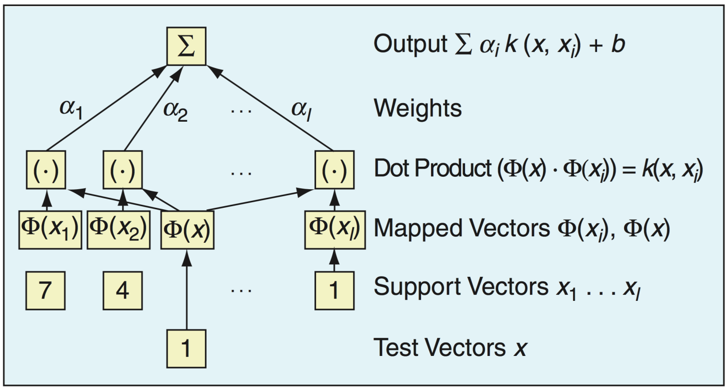

Σ
Output $\sum \alpha_i k(x, x_i) + b$
α_1
α_2
…
α_l
Weights
(·)
(·)
…
(·)
Dot Product $(\Phi(x) \cdot \Phi(x_i)) = k(x, x_i)$
$\Phi(x_1)$
$\Phi(x_2)$
$\Phi(x)$
$\Phi(x_l)$
Mapped Vectors $\Phi(x_i)$, $\Phi(x)$
7
4
…
1
Support Vectors $x_1 \dots x_l$
1
Test Vectors x

Figure 1: SVM architecture

So far, we have already discussed the basic background in terms of the traditional *Support Vector Machine (SVM)*. In *LS-SVM* formulation, one uses equality constraints instead of inequality constraints and a least squares error term (sum-squared error, SSE) in order to obtain a linear set of equations in the dual space. The main benefit of the LS-SVM technique is that it transforms the traditional QP problem to a simultaneous linear system problem, thus ensuring simplicity in computations, fast convergence and high precision [12].

2.2. Bayesian Inference Framework

Bayesian inference is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis as more evidence or information becomes available. In this framework, the overall aims are to find models that are well-matched to the data, and to use these models to make optimal predictions. Bayesian inference is widely used in supervised machine learning approaches.

Bayesian inference satisfies the likelihood principle [13]: our inferences depend only on the probabilities assigned to the data that were received, not on properties of other data sets which might have occurred but did not. Here, we introduce the background of this framework mostly based on MacKay's review [14].

We use the notation for *conditional probability* as: $P(A|B, \mathcal{H})$, which means "the probability of A, given B and $\mathcal{H}$". The statements B and $\mathcal{H}$ are the conditional assumptions on which this measure of plausibility is based. For example, imagine that Joe has a test for a disease; if A is the proposition "the test result is positive", and B is "Joe has the disease", then the quantity $P(A|B, \mathcal{H})$ is a number between 0 and 1 which expresses how likely we think the test would be to give the right answer, assuming that Joe did have the disease, and given the overall assumptions $\mathcal{H}$ about the reliability of the test.

Then according to Bayes' theorem, we get the following equation:

$$P(B|A, \mathcal{H}) = \frac{P(A|B, \mathcal{H})P(B|\mathcal{H})}{P(A|\mathcal{H})} \tag{8}$$

Here, our overall model of the situation, $\mathcal{H}$, is a conditioning statement on the right hand side of all the probabilities.

In addition, we can evaluate the plausibility of two alternative theories $\mathcal{H}_1$ and $\mathcal{H}_2$ in the light of data D. Using Bayes' theorem, we relate the plausibility of model $\mathcal{H}_1$ given the data, $P(\mathcal{H}_1|D)$, to the predictions made by the model about the data, $P(D|\mathcal{H}_1)$, and the prior plausibility of $\mathcal{H}_1$, $P(\mathcal{H}_1)$. This yields the following probability ratio between theory $\mathcal{H}_1$ and theory $\mathcal{H}_2$:

$$\frac{P(\mathcal{H}_1|D)}{P(\mathcal{H}_2|D)} = \frac{P(\mathcal{H}_1)}{P(\mathcal{H}_2)}\frac{P(D|\mathcal{H}_1)}{P(D|\mathcal{H}_2)} \tag{9}$$

where the first ratio on the right hand side measures how much our initial beliefs favored $\mathcal{H}_1$ over $\mathcal{H}_2$. The second ratio expresses how well the observed data were predicted by $\mathcal{H}_1$, compared to $\mathcal{H}_2$.

MacKay [14] gives detailed examples on how to implement Bayesian inference framework on neural networks with three levels of inference. In this paper, we will focus on the application to SVM.

3. Methodology

In this section, we present the main methodology of Gestel *et al*'s paper: *Financial Time Series Prediction Using Least Squares Support Vector Machines Within the Evidence Framework* [1]. In that paper, the Bayesian evidence framework is combined with least squares support vector machines (LS-SVMs) for nonlinear regression in order to infer nonlinear models of a time series and the corresponding volatility.

This review is organized as follows. The three levels for inferring the parameters of the LS-SVM time series model are described in Section 3.1. The inference of the volatility model is discussed in Section 3.2. An overview of the design of the LS-SVM time series and volatility model within the evidence framework is given in Section 3.3.

3.1. Inference of the Time Series Model

On the first level of inference, a statistical framework is related to the LS-SVM formulation which allows to include the time-varying volatility of the market by an appropriate choice of several hyperparameters. The hyperparameters of the model are inferred on the second level of inference. Model comparison is performed on the third level of inference in order to automatically tune the parameters of the kernel function and to select the relevant inputs.

3.1.1. Inference of the Model Parameters (Level 1)

Given the data points $D = \{(x_i, y_i)\}_{i=1}^{N}$ and the hyperparameters μ and $\zeta_{1:N} = [\zeta_1, \zeta_2, \cdots, \zeta_N]$ of the model $\mathcal{H}$ (LS-SVM with kernel function K), we obtain the model parameters by maximizing the posterior $P(w, b|D, \log\mu, \log\zeta_{1:N}, \mathcal{H})$. Application of Bayes' rule at the first level of inference [15] gives:

$$P(w, b|D, \log\mu, \log\zeta_{1:N}, \mathcal{H}) = \frac{P(D|w, b, \log\mu, \log\zeta_{1:N}, \mathcal{H})P(w, b|\log\mu, \log\zeta_{1:N}, \mathcal{H})}{P(D|\log\mu, \log\zeta_{1:N}, \mathcal{H})} \tag{10}$$

where $P(D|\log\mu, \log\zeta_{1:N}, \mathcal{H})$ follows from normalization and is independent of w and b.

The weight parameters w are assumed to have a Gaussian distribution. A uniform distribution for the prior on b is taken, which can also be approximated as a Gaussian distribution. Then assuming Gaussian distributed additive noise $e_i (i = 1, \cdots, N)$ with zero mean and variance ζ_i^{-1}, we obtain the likelihood corresponds to the error term $\sum_{i=1}^{N} \zeta_i E_{D,i}$. Finally, by substituting all the parameters into Equation (10) and neglecting all constants, Bayes' rule yields (the derivation in the paper is omitted):

$$P(w, b|D, \log \mu, \log \zeta_{1:N}, \mathcal{H}) \propto \exp(-\frac{\mu}{2} w^T w) \exp(-\sum_{i=1}^{N} \frac{\zeta_i}{2} e_i^2) \tag{11}$$

Taking the negative logarithm, the maximum *posterior* model parameters w_{MP} and b_{MP} are obtained as the solution to the following optimization problem:

$$\min_{w,b} \mathcal{J}_1(w, b) = \mu E_W + \sum_{i=1}^{N} \zeta_i E_{D,i} \tag{12}$$

with

$$E_W = \frac{1}{2} w^T w \tag{13}$$

$$E_{D,i} = \frac{1}{2} e_i^2 = \frac{1}{2} (y_i - w^T \varphi(x_i) - b)^2 \tag{14}$$

The uncertainty on the estimated model parameters results into an additional uncertainty for the one step ahead prediction $\hat{y}_{MP,N+1} = z_{MP} = w_{MP}^T \varphi(x) + b_{MP}$ with variance $\sigma_{\hat{y}_{N+1}}^2 = \zeta_{N+1}^{-1} + \sigma_z^2$. The first term ζ_{N+1}^{-1} corresponds to the volatility at the next time step and has to be predicted by volatility model, which will be introduced in next section. The second term σ_z^2 is due to the Gaussian uncertainty on the estimated model parameters w and b in the linear transform $z = w^T \varphi(x) + b$.

In order to obtain a practical expression in the dual space, one solves the following optimization problem corresponding to (12):

$$\min_{w,e} \mathcal{J}_1(w, e) = \frac{\mu}{2} w^T w + \frac{1}{2} \sum_{i=1}^{N} \zeta_i e_i^2 \tag{15}$$

$$\text{s.t.} \quad y_i = w^T \varphi(x_i) + b + e_i, \quad i = 1, \ldots, N \tag{16}$$

To solve the minimization problem, one constructs the Lagrangian:

$$\mathcal{L}_1(w, b, e; \alpha) = \mathcal{J}_1(w, e) - \sum_{i=1}^{N} \alpha_i [w^T \varphi(x_i) + b + e_i - y_i] \tag{17}$$

where $\alpha_i \in \mathbf{R}$ are the Lagrange multipliers (also called support values). Recall that the input data points with non-zero Lagrange multipliers α are defined as *Support Vectors*. The conditions for optimality are give by:

$$\begin{cases} \frac{\partial \mathcal{L}_1}{\partial w} = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \frac{\partial \mathcal{L}_1}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i = 0 \\ \frac{\partial \mathcal{L}_1}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma_i e_i, \quad i = 1, \ldots, N \\ \frac{\partial \mathcal{L}_1}{\partial \alpha_i} = 0 \rightarrow b = y_i - w^T \varphi(x_i) - e_i, \quad i = 1, \ldots, N \end{cases} \tag{18}$$

with $\gamma_i = \zeta_i / \mu (i = 1, \ldots, N)$. Eliminating w and e, one obtains the following linear Karush-Kuhn-Tucker system in α and b:

$$\left[\begin{array}{c|c} 0 & 1_v^T \\ \hline 1_v & \Omega + D_\gamma^{-1} \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline y \end{array} \right] \tag{19}$$

with $y = [y_1; \ldots; y_N]$, $1_v = [1; \ldots; 1]$, $e = [e_1; \ldots; e_N]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$ and $D_\gamma = \mathrm{diag}([\gamma_1; \ldots; \gamma_N])$. Then apply Mercer's theorem with the Ω matrix:

$$\Omega_{ij} = \varphi(x_i)^T \varphi(x_j) = K(x_i, x_j) \tag{20}$$

The LS-SVM regressor is then obtained as:

$$z_{MP} = \hat{y}_{MP,N+1} = \sum_{i=1}^{N} \alpha_i K(x, x_i) + b_{MP} \tag{21}$$

3.1.2. Inference of the Hyperparameters (Level 2)

The hyperparameters μ and $\zeta_i (i = 1, \ldots, N)$ are inferred from the data D by applying Bayes' rule on the second level:

$$\begin{aligned} P(\log \mu, \log \zeta_{1:N} | D, \mathcal{H}) &= \frac{P(D | \log \mu, \log \zeta_{1:N}, \mathcal{H}) P(\log \mu, \log \zeta_{1:N} | \mathcal{H})}{P(D | \mathcal{H})} \\ &\propto P(D | \log \mu, \log \zeta_{1:N}, \mathcal{H}) \end{aligned} \tag{22}$$

where a flat, non-informative prior is assumed on the hyperparameters μ and ζ_i. Again, we assume Gaussian distributions for the parameters. This gives us:

$$P(\log \mu, \log \zeta_{1:N} | D, \mathcal{H}) \propto \sqrt{\frac{\mu^{n_f} \Pi_{i=1}^{N} \zeta_i}{\det H}} \exp\left(-\mathcal{J}_1(w_{MP}, b_{MP})\right) \tag{23}$$

Using the expression for $\det H$ from Appendix A [1] and taking the negative logarithm, we find the maximum *posterior* estimates μ_{MP} and $\zeta_{MP,i}$ by minimizing the level 2 cost function:

$$\begin{aligned} \mathcal{J}_2(\mu, \zeta_{1:N}) = \mu E_W(w_{MP}) &+ \sum_{i=1}^{N} \zeta_i E_{D,i}(w_{MP}, b_{MP}) \\ &+ \frac{1}{2} \sum_{i=1}^{N_{eff}} \log(\mu + \lambda_{G,i}) - \frac{N_{eff}}{2} \log \mu \\ &- \frac{1}{2} \sum_{i=1}^{N} \log \zeta_i + \frac{1}{2} \log(\sum_{i=1}^{N} \zeta_i) \end{aligned} \tag{24}$$

Notice that this is an optimization problem in $N + 1$ unknowns and may require long computations. Therefore, we will first discuss the inference in the case of constant $\zeta_i = \zeta$. This value for the hyperparameters will then be used to infer the non-constant ζ_i.

The optimal constant hyperparameters μ_{MP} and ζ_{MP} are obtained by solving the optimal problem:

$$\min_{\mu,\zeta} \mathcal{J}_3(\mu,\zeta) = \mathcal{J}_1(w_{MP}, b_{MP}) + \frac{1}{2}\sum_{i=1}^{N_{eff}} \log(\mu + \zeta\lambda'_{G,i}) - \frac{N_{eff}}{2}\log\mu - \frac{N-1}{2}\log\zeta \tag{25}$$

Then the non-constant hyperparameters $\zeta_{MP,i}$ is given by (detailed derivation is in [1]):

$$\zeta_{MP,i}^{-1} \simeq e_i^2 + \sigma_{z_i}^2, \quad i = 1, \ldots, N \tag{26}$$

where both e_i and $\sigma_{z_i}^2$ are obtained from the LS-SVM model with constant ζ_{MP} that is given by solving problem (25).

The estimates $\zeta_{MP,i}$ will not be used to infer the LS-SVM time series model directly. Instead, they are used to infer the LS-SVM volatility model in later section. The modeled $\hat{\zeta}_{MP,i}$ of the LS-SVM volatility model are far less noisy estimates of the corresponding volatility and will be used to infer the LS-SVM time series model using a weighted least squares error term.

3.1.3. Model Comparison (Level 3)

In this Section Bayes rule is applied to rank the evidence of different models $\mathcal{H}_j$. Different models $\mathcal{H}_j$ correspond to different choices for the kernel function; e.g., for an RBF kernel with tuning parameter σ_j. The probability of the corresponding models $\mathcal{H}_j$ is calculated in order to select the tuning parameter σ_j with the greatest model evidence.

By applying Bayes rule on the third level, we obtain the posterior for the model $\mathcal{H}_j$:

$$P(\mathcal{H}_j|D) \propto P(D|\mathcal{H}_j)P(\mathcal{H}_j) \tag{27}$$

Since the model selection is performed before inferring $\hat{\zeta}_{MP,i}$, we will assume a constant $\zeta_i = \zeta$ in this section. Then the posterior $P(D|\mathcal{H}_j)$ is given by:

$$P(D|\mathcal{H}_j) \propto P(D|\log\mu_{MP}, \log\zeta_{MP}, \mathcal{H}_j) \cdot \frac{\sigma_{\log\mu|D}\sigma_{\log\zeta|D}}{\sigma_{\log\mu}\sigma_{\log\zeta}} \tag{28}$$

where $\sigma_{\log\mu|D}$ and $\sigma_{\log\zeta|D}$ are error bars which can be obtained from problem (24).

Ranking of models according to model quality $P(D|\mathcal{H}_j)$ is thus based on the goodness of the fit and the Occam factor [14], which punishes for over-parameter models.

3.2. Volatility Modeling

We use the inferred hyperparameters $\zeta_{MP,i}$ from Equation (26) to train the LS-SVM volatility model. Instead of modeling and predicting the inferred $\zeta_{MP,i}$ or $\zeta_{MP,i}^{-1}$ directly, we will model $\zeta_{MP,i}^{-1/2}$, which corresponds to the prediction of absolute returns [16].

The inference of volatility model uses the same approach as the inference of the LS-SVM time series model. Readers are referred to the original paper [1] to see more details on how to set up the QP problem.

3.3. Design of the Bayesian LS-SVM

By applying the theory from the previous Sections, the design of the LS-SVM time series and volatility model within the evidence framework is shown in Figure 2.

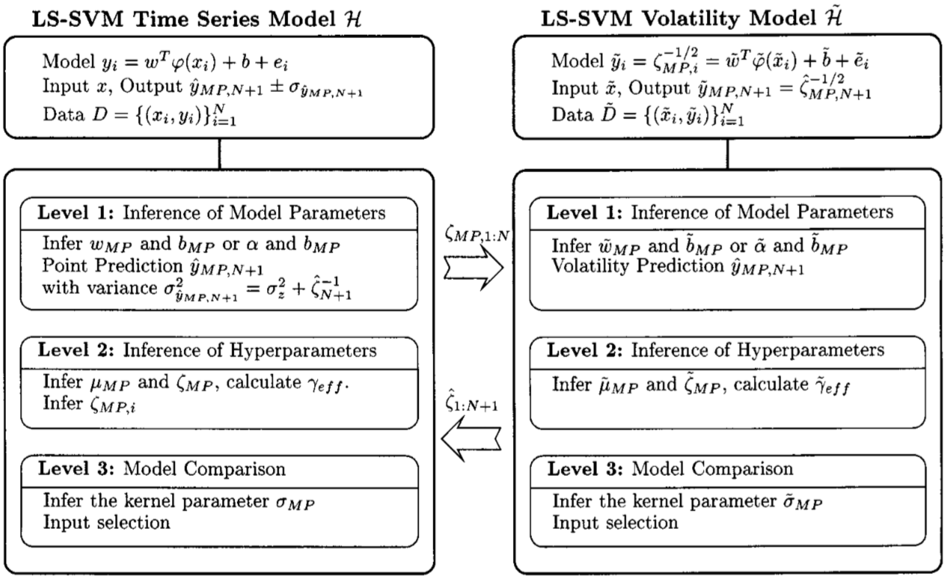

LS-SVM Time Series Model $\mathcal{H}$
Model $y_i = w^T \varphi(x_i) + b + e_i$
Input x, Output $\hat{y}_{MP,N+1} \pm \sigma_{\hat{y}_{MP,N+1}}$
Data $D = \{(x_i, y_i)\}_{i=1}^N$
Level 1: Inference of Model Parameters
Infer w_{MP} and b_{MP} or α and b_{MP}
Point Prediction $\hat{y}_{MP,N+1}$
with variance $\sigma^2_{\hat{y}_{MP,N+1}} = \sigma_z^2 + \hat{\zeta}_{N+1}^{-1}$
Level 2: Inference of Hyperparameters
Infer μ_{MP} and ζ_{MP}, calculate γ_{eff}.
Infer $\zeta_{MP,i}$
Level 3: Model Comparison
Infer the kernel parameter σ_{MP}
Input selection
$\zeta_{MP,1:N}$
$\hat{\zeta}_{1:N+1}$
LS-SVM Volatility Model $\tilde{\mathcal{H}}$
Model $\tilde{y}_i = \zeta_{MP,i}^{-1/2} = \tilde{w}^T \tilde{\varphi}(\tilde{x}_i) + \tilde{b} + \tilde{e}_i$
Input $\tilde{x}$, Output $\tilde{y}_{MP,N+1} = \hat{\zeta}_{MP,N+1}^{-1/2}$
Data $\tilde{D} = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^N$
Level 1: Inference of Model Parameters
Infer $\tilde{w}_{MP}$ and $\tilde{b}_{MP}$ or $\tilde{\alpha}$ and $\tilde{b}_{MP}$
Volatility Prediction $\hat{y}_{MP,N+1}$
Level 2: Inference of Hyperparameters
Infer $\tilde{\mu}_{MP}$ and $\tilde{\zeta}_{MP}$, calculate $\tilde{\gamma}_{eff}$
Level 3: Model Comparison
Infer the kernel parameter $\tilde{\sigma}_{MP}$
Input selection

Figure 2: Design of the Bayesian LS-SVM

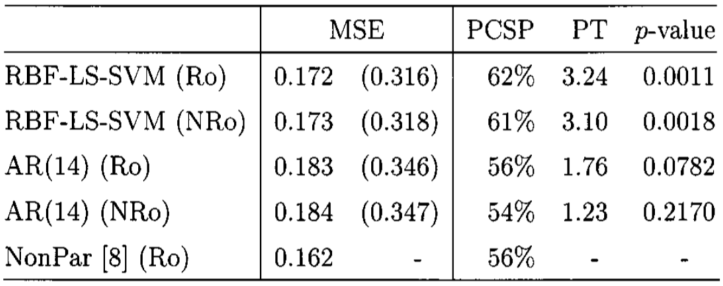
The model parameters w, b, the hyperparameters μ, ζ_i and the kernel parameter and relevant inputs of the time series model $\mathcal{H}$ are inferred from the data D on different levels of inference. The inferred hyperparameters $\zeta_{MP,i}$ are used to estimate the parameters $\tilde{w}$, $\tilde{b}$, $\tilde{\mu}$, $\tilde{\zeta}$ and $\tilde{\sigma}$ of the volatility model $\tilde{\mathcal{H}}$. The predicted volatility is used to calculate error bars $\sigma^2_{\tilde{y}_{MP,N+1}} = \hat{\zeta}^{-1}_{MP,N+1} + \sigma_z^2$ on the point prediction $\tilde{y}_{MP,N+1} = z_{MP}$ given by (21).

4. Examples

The design of the LS-SVM regressor in the evidence framework is applied to two cases in the original paper [1]. Here, we only present one of them.

The performance of the LS-SVM time series model is compared with results from the literature [17], [18] for the case of one step ahead prediction of the US short term interest rate. The results are listed in the table below.

These are all out of sample tests. The models used are: LS-SVM with RBF-Kernel (RBF-LS-SVM), AR(14) model and the non-parametric model (NonPar), using both rolling (Ro) and Non-rolling (NRo) approaches. First, the sample MSE and corresponding sample standard deviation are reported. Then the directional accuracy is assessed by the Percentage of Correct Sign Predictions (PCSP), the Pesaran-Timmermann statistic (PT) and the corresponding p-value. We can see that the LS-SVM with RBF-Kernel model clearly performs better than the other models with respect to the directional accuracy criterion.



	MSE		PCSP	PT	p-value
RBF-LS-SVM (Ro)	0.172	(0.316)	62%	3.24	0.0011
RBF-LS-SVM (NRo)	0.173	(0.318)	61%	3.10	0.0018
AR(14) (Ro)	0.183	(0.346)	56%	1.76	0.0782
AR(14) (NRo)	0.184	(0.347)	54%	1.23	0.2170
NonPar [8] (Ro)	0.162	-	56%	-	-

References

[1] T. Van Gestel, J. A. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, J. Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Transactions on neural networks 12 (4) (2001) 809–821.

[2] N. I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Computational Intelligence Magazine 4 (2).

[3] S. J. Orfanidis, Optimum signal processing: an introduction, Macmillan publishing company, 1988.

[4] R. Gençay, M. Dacorogna, U. A. Muller, O. Pictet, R. Olsen, An introduction to high-frequency finance, Elsevier, 2001.

[5] Z. Ding, C. W. Granger, R. F. Engle, A long memory property of stock market returns and a new model, Journal of empirical finance 1 (1) (1993) 83–106.

[6] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, Time series analysis: forecasting and control, John Wiley & Sons, 2015.

[7] R. Perrelli, Introduction to arch & garch models, University of Illinois Optional TA Handout (2001) 1–7.

[8] J. Kamruzzaman, Artificial neural networks in finance and manufacturing, IGI Global, 2006.

[9] V. Vapnik, Statistical learning theory. 1998, Wiley, New York, 1998.

[10] N. Cristianini, J. Shawe-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.

[11] S. Kumar, Neural networks: a classroom approach, Tata McGraw-Hill Education, 2004.

[12] J. A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural processing letters 9 (3) (1999) 293–300.

[13] J. O. Berger, R. L. Wolpert, M. Bayarri, M. DeGroot, B. M. Hill, D. A. Lane, L. LeCam, The likelihood principle, Lecture notes-Monograph series 6 (1988) iii–199.

[14] D. J. MacKay, Probable networks and plausible predictionsa review of practical bayesian methods for supervised neural networks, Network: Computation in Neural Systems 6 (3) (1995) 469–505.

[15] C. M. Bishop, Neural networks for pattern recognition, Oxford university press, 1995.

[16] C. W. Granger, C.-y. Sin, Modelling the absolute returns of different stock indices: exploring the forecastability of an alternative measure of risk.

[17] J. T. Barkoulas, C. F. Baum, J. Onochie, A nonparametric investigation of the 90-day t-bill rate, Review of Financial Economics 6 (2) (1997) 187–198.

[18] J. G. De Gooijer, D. Zerom, Kernel-based multistep-ahead predictions of the us short-term interest rate, Journal of Forecasting 19 (4) (2000) 335–353.