Quantitative Finance

ISSN: 1469-7688 (Print) 1469-7696 (Online) Journal homepage: http://www.tandfonline.com/loi/rquf20

Practical Bayesian support vector regression for financial time series prediction and market condition change detection

T. Law & J. Shawe-Taylor

To cite this article: T. Law & J. Shawe-Taylor (2017): Practical Bayesian support vector regression for financial time series prediction and market condition change detection, Quantitative Finance, DOI: 10.1080/14697688.2016.1267868

To link to this article: http://dx.doi.org/10.1080/14697688.2016.1267868

Published online: 09 Mar 2017.

Quantitative Finance, 2017
http://dx.doi.org/10.1080/14697688.2016.1267868

Routledge
Taylor & Francis Group

Practical Bayesian support vector regression for financial time series prediction and market condition change detection

T. LAW* and J. SHAWE-TAYLOR

Department of Computer Science, University College London, Gower Street, London WC1E 6BT, United Kingdom

(*Received 8 April 2016; accepted 18 November 2016; published online 9 March 2017*)

Support vector regression (SVR) has long been proven to be a successful tool to predict financial time series. The core idea of this study is to outline an automated framework for achieving a faster and easier parameter selection process, and at the same time, generating useful prediction uncertainty estimates in order to effectively tackle flexible real-world financial time series prediction problems. A Bayesian approach to SVR is discussed, and implemented. It is found that the direct implementation of the probabilistic framework of Gao *et al*. returns unsatisfactory results in our experiments. A novel enhancement is proposed by adding a new kernel scaling parameter μ to overcome the difficulties encountered. In addition, the multi-armed bandit Bayesian optimization technique is applied to automate the parameter selection process. Our framework is then tested on financial time series of various asset classes (i.e. equity index, credit default swaps spread, bond yields, and commodity futures) to ensure its flexibility. It is shown that the generalization performance of this parameter selection process can reach or sometimes surpass the computationally expensive cross-validation procedure. An adaptive calibration process is also described to allow practical use of the prediction uncertainty estimates to assess the quality of predictions. It is shown that the machine-learning approach discussed in this study can be developed as a very useful pricing tool, and potentially a market condition change detector. A further extension is possible by taking the prediction uncertainties into consideration when building a financial portfolio.

Keywords: Support vector machines regression; Kernel scaling; Machine learning; Bayesian inference; Multi-armed bandit Bayesian optimization; Gaussian process

JEL Classification: C44, C45, C61

1. Introduction

Support vector regression (SVR) has been one of the most popular Machine-learning algorithms for over a decade. Its ability to predict financial time series has been demonstrated in various studies (Müller *et al*. 1997, Tay and Cao 2001, Cao and Tay 2003, Kim 2003, Lu *et al*. 2009, Gündüz and Uhrig-Homburg 2011) with satisfactory empirical results. Although the base assumption of most machine-learning models requires i.i.d. data, there is literature (Mohri and Rostamizadeh 2008, Ralaivola *et al*. 2010) suggesting non-i.i.d. data can be used to train the statistical learning system by increasing sample sizes. This supports SVR as a relevant tool for time series prediction. Comparing to other linear time series models, which require careful designing of inputs, SVR allows flexible mapping of high dimensional features to capture non-linear relationships and at the same time with a regularization technique to reduce over-fitting. However, on the other hand, financial experts in the industry may step back from using this algorithm for two reasons: (1) it is not easy, and sometimes computationally expensive to determine the parameters for the algorithm, (2) it does not provide an estimate of prediction uncertainties.

We propose to investigate extensions to the approach that make up for these two disadvantages by generalizing it within a Bayesian approach. This allows a more efficient parameter selection procedure as well as the derivation of a prediction uncertainty estimate. This is particularly important in the financial context to avoid decisions based on unreliable predictions. A Bayesian approach to SVR (Gao *et al*. 2002) is discussed, and implemented. It is found in our experiments that direct implementation of the probabilistic framework proposed by Gao *et al*. (2002) gives unsatisfactory results. A novel enhancement is proposed by adding a new kernel scaling parameter μ to overcome the difficulties encountered. In addition, because the gradient optimization approach proposed by Gao *et al*. (2002)

*Corresponding author. Email: timothy.law.14@ucl.ac.uk

is found to be impractical, we develop an alternative optimization method. Multi-arm bandit Bayesian optimization has been gaining popularity in recent years (Shahriari *et al*. 2016), mainly for its capability to actively and efficiently search for the global optimum for a complex or even unknown function. We make use of this technique to fully automate the parameter selection process.

We test our framework to predict financial time series of various asset classes to ensure its flexibility. The generalization performance from models selected by different parameter selection approaches (cross-validation, grid search, and Bayesian optimization) are compared. The practicality of the proposed framework is further examined by generating daily predictions, aiming to allow a thorough investigation of the prediction uncertainty estimates derived. A calibration process is established to show how these uncertainty estimates may be utilized to significantly improve predictions as well as to detect market condition changes.

The paper is organized as follows: section 2 gives an over view of SVR. Section 3 discusses the Bayesian probabilistic framework (Gao *et al*. 2002), and the rationale behind our novel enhancement to the probabilistic framework. Section 4 covers the theoretical background of the multi-armed bandit Bayesian optimization algorithm. In section 5, the implementation and results of various experiments are explained with the introduction of an adaptive calibration process. Section 6 concludes the findings and discusses possible research extensions.

2. Support Vector Regression

The concepts of support vector machine (SVM) can naturally be applied to handle regression problems. Instead of classifying testing examples into one of the two outcomes (class 1 or class 2), it is used to address target variables with real values. The ε-Support Vector Regression (ε-SVR) can be considered similar to the approach of applying a linear regression in the feature space. In contrast to the squared loss function used in the least squares regression, the error function for ε-SVR is the ε-insensitive loss function. This dictates that the errors smaller than ε are ignored. Figure 1 demonstrates this intuition. This method leads to sparsity similar to SVMs meaning that the form of the regression depends only on the support vectors.

Consider we have training examples $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n} \in \mathbb{R}$, and a target variable $y_i \in \mathbb{R}$ for each $\boldsymbol{x_i}$. The ε-insensitive loss function has the form

$$|y - f(\boldsymbol{x})|_\varepsilon := \max\{0, |y - f(\boldsymbol{x})| - \varepsilon\}$$

To estimate the linear regression $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$ with this loss function, we minimize

$$\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} |y_i - f(\boldsymbol{x}_i)|_\varepsilon \tag{1}$$

While the second term of equation (1) is the ε-insensitive loss function, the first term is considered the regularization constraint to prevent the ε-SVR from over-fitting. The parameter C controls the trade-off between the complexity of the model and the error. Introducing the "slack variables" (ξ_i^+, ξ_i^-) into this optimization problem, it becomes

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}(\xi_i^+ + \xi_i^-) \\ \text{subject to} \quad & y_i - \boldsymbol{w}\cdot\boldsymbol{x_i} - b \le \epsilon + \xi_i^+ \\ & \boldsymbol{w}\cdot\boldsymbol{x_i} + b - y_i \le \epsilon + \xi_i^- \\ & \xi_i^+,\ \xi_i^- \ge 0 \qquad \forall i \end{aligned} \tag{2}$$

This optimization is very similar to the one for SVMs. Smola and Schölkopf provide detailed derivation on how the dual problem can be obtained using the Lagrange multiplier method (Smola and Schölkopf 2004).

$$\begin{aligned} \max_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-} \quad & W(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-) = -\epsilon\sum_{i=1}^{n}(\alpha_i^+ + \alpha_i^-) + \sum_{i=1}^{n} y_i(\alpha_i^+ - \alpha_i^-) \\ & - \frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-)K(\boldsymbol{x_i}, \boldsymbol{x_j}) \\ \text{subject to} \quad & 0 \le \alpha_i^+, \alpha_i^- \le C, \quad \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-) = 0 \end{aligned} \tag{3}$$

Calculating the partial derivative of the Lagrangian $\mathcal{L}$ with respect to $\boldsymbol{w}$, we can calculate the optimal $\boldsymbol{w}$ using the optimal $\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-$.

$$\boldsymbol{w} = \sum_{i=1}^{n}(\alpha_i^+ - \alpha_i^-)\phi(\boldsymbol{x_i})$$

For a new data point z, the regression estimate can be obtained using the formula

$$f(z) = \boldsymbol{w}^* \cdot \phi(z) + b^* = \sum_{i=1}^{n}(\alpha_i^{+*} - \alpha_i^{-*})K(\boldsymbol{x}_j, z) + b^*$$

Similar to SVMs, this optimization problem satisfies the KKT conditions. The dual complementarity condition suggests that $\alpha_i^+, \alpha_i^- \ge 0$ only when $|f(x_i) - y| \ge \epsilon$, which equivalently means the corresponding training examples on or outside of the ϵ band. These training examples are the support vectors.

3. Probabilistic Framework for SVR

The selection of parameters has always been one of the most important tasks when training supervised learning algorithms, and SVR is no exception. In order to achieve good generalization performance, a process is required to fine tune the parameters in order to balance the trade-off between variance and bias. Traditionally, the multi-fold cross-validation process is employed. This process is generally very effective, but usually computationally heavy. In this section, a probabilistic framework for SVR is investigated aiming to accelerate the parameter selection process. Gao *et al*. (2002) demonstrates how MacKay's evidence framework (MacKay 1992) is used to determine the parameters. Also, the approximation of an error bar formula for the SVR predictions is derived.

We propose a new parameter μ in addition to the SVR probabilistic framework of Gao *et al*. (2002). This parameter μ scales the kernel, and is defined to be the *a priori* estimate of the output variance. We show that adding such a parameter does not affect the SVR quadratic programming problem, but may act as a scaling parameter to allow the evidence function to be more flexible when handling data with different ranges. Because error bar estimate derived by Gao *et al*. (2002) is solely

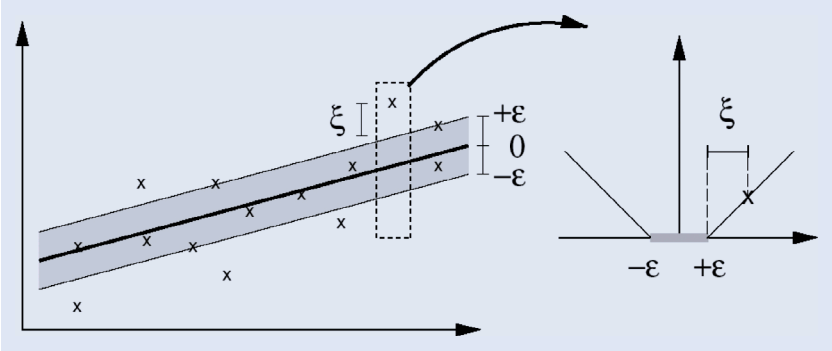

Figure 1. The loss function of ε-SVR.

from the kernel features, this parameter also adjusts the level of this variance to better match the range of the SVR prediction.

3.1. Bayesian Evidence Framework

In this section, the Bayesian evidence framework for SVR proposed by Gao *et al.* (2002) is discussed. SVR can be interpreted as a maximum a posteriori (MAP) solution to inference problems with Gaussian priors and an appropriate likelihood function based on a probabilistic interpretation. Gao *et al.* (2002) derived the evidence formula, and the error bar approximation under this framework.

In regression problems, the functional dependency $f(\cdot)$ between a set of sampled points $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}$ taken from $\mathbb{R}^d$, and target values $Y = \{y_1, y_2, \ldots, y_n\}$ with $y_i \in \mathbb{R}$ is defined as follows. The training data $D = \{X, Y\}$ are collected by randomly sampling from the model:

$$y_i = f(\boldsymbol{x_i}) + \delta_i \qquad i = 1, 2, \ldots, n$$

where $f(\cdot)$ is the underlying function, and δ_i are independent, identically distributed (i.i.d.) random noise. Regression aims to estimate the function f from the data-set D. In the Bayesian approach, the function $\boldsymbol{f}$ is regarded as the realization of a random field with a known *a priori* probability. Let $\boldsymbol{f} = [f(x_1), f(x_2), \ldots, f(x_n)]^T$. The *a posteriori* probability of $\boldsymbol{f}$ given the training data D can then be derived by Bayes' theorem:

$$P(\boldsymbol{f}|D) = \frac{P(D|\boldsymbol{f})P(\boldsymbol{f})}{P(D)} \tag{4}$$

where $P(\boldsymbol{f})$ is the *a priori* probability of the random field, and $P(D|\boldsymbol{f})$ is the conditional probability of the data D given the function values $\boldsymbol{f}$. This conditional probability $P(D|\boldsymbol{f})$, which is also the likelihood, can simply be interpreted as the noise model. $P(D|\boldsymbol{f})$ is evaluated by:

$$P(D|\boldsymbol{f}) = \prod_{i=1}^{n} P(\delta_i) = \prod_{i=1}^{n} P(y_i - f(\boldsymbol{x_i}))$$

Gao *et al.* (2002) suggested that the likelihood function be written as:

$$P(D|\boldsymbol{f}) = [G(C, \varepsilon)]^n \exp\left(-C \sum_{i=1}^{n} L_\varepsilon(y_i - f(\boldsymbol{x_i}))\right) \tag{5}$$

where $G(C, \varepsilon) = \frac{C}{2(\varepsilon C+1)}$ is the corresponding normalizing constant. For ε-SVR, as mentioned in section 2, the ε-insensitive loss function is given by:

$$L_\varepsilon(y_i - f(\boldsymbol{x_i})) = \begin{cases} 0, & \text{for} |y - f(\boldsymbol{x})| < \varepsilon, \\ |y - f(\boldsymbol{x})| - \varepsilon, & \text{otherwise.} \end{cases}$$

The *a priori* distribution $P(\boldsymbol{f})$ is assumed to be a multivariate Gaussian with a zero mean and covariance function $K(\cdot, \cdot)$.

$$P(\boldsymbol{f}) = \frac{1}{\sqrt{\det 2\pi\ K_{X,X}}} \exp\left\{-\frac{1}{2}\, \boldsymbol{f}^T K_{X,X}^{-1}\, \boldsymbol{f}\right\} \tag{6}$$

where $K_{X,X} = [K(\boldsymbol{x_i}, \boldsymbol{x_j})]$ is the covariance matrix at the training points X. Following Bayes' theorem given in equation (4), the *a posteriori* distribution is determined by combining equation (5) and (6).

$$P(\boldsymbol{f}|D) = \frac{[G(C, \varepsilon)]^n}{\sqrt{\det 2\pi\ K_{X,X}} P(D)} \times \exp\left\{-C \sum_{i=1}^{n} L_\varepsilon(y_i - f(\boldsymbol{x_i})) - \frac{1}{2}\, \boldsymbol{f}^T K_{X,X}^{-1}\, \boldsymbol{f}\right\} \tag{7}$$

where $G(C, \varepsilon) = \frac{C}{2(\varepsilon C+1)}$, and $P(D)$ is the evidence which is explained later in the section. The MAP estimate of f is computed by maximizing the *a posteriori* distribution provided in equation (7), which is the same as minimizing the risk function as follows:

$$R(f) = C \sum_{i=1}^{n} L_\varepsilon(y_i - f(\boldsymbol{x_i})) + \frac{1}{2}\, \boldsymbol{f}^T K_{X,X}^{-1}\, \boldsymbol{f} \tag{8}$$

This is equivalent to the primal problem for SVR (equation (2)) when $\frac{1}{2}\boldsymbol{f}^T K_{X,X}^{-1}\, \boldsymbol{f} = \frac{1}{2}\|\boldsymbol{w}\|^2$, which is suggested in the Lagrangian condition. As per the SVR optimization problem, the model that minimizes the risk function is defined as:

$$\boldsymbol{f}^* = \sum_{i=1}^{n} (\alpha_i^+ - \alpha_i^-) K(\boldsymbol{x_i}, \boldsymbol{x}) \tag{9}$$

With respect to the optimal solution $\boldsymbol{f}^*$, the training dataset X are divided into four parts:

$$\begin{aligned} X_0 &= \{\boldsymbol{x_i} | \left|y_i - f^*(\boldsymbol{x_i})\right| < \varepsilon \quad \text{with} \quad \alpha_i^+ = \alpha_i^- = 0\} \\ X_C &= \{\boldsymbol{x_i} | \left|y_i - f^*(\boldsymbol{x_i})\right| > \varepsilon \quad \text{with} \quad \alpha_i^+ = C, \alpha_i^- = 0, \\ &\qquad\qquad \text{or} \quad \alpha_i^+ = 0, \alpha_i^- = C\} \\ X_{M^-} &= \{\boldsymbol{x_i} | f^*(\boldsymbol{x_i}) - y_i - \varepsilon = 0 \quad \text{with} \quad 0 \le \alpha_i^- \le C\} \\ X_{M^+} &= \{\boldsymbol{x_i} | y_i - f^*(\boldsymbol{x_i}) - \varepsilon = 0 \quad \text{with} \quad 0 \le \alpha_i^+ \le C\} \end{aligned} \tag{10}$$

For later derivation, let us define $X_M = X_{M^+} \cup X_{M^-}$, and the support vectors $X_{sv} = X_M \cup X_C$. Also, denote that $\bar{X}_M = X \backslash X_M$.

In order to determine how to use the probabilistic framework to select the best parameters, the evidence $P(D)$ has to be determined. The evidence is simply defined as the marginal likelihood of the data for a particular model, and can be obtained by integrating the *a posteriori* distribution over the parameter space $\boldsymbol{f}$.

$$P(D) = \int P(\boldsymbol{f}) P(D|\boldsymbol{f}) \mathrm{d}\boldsymbol{f} \tag{11}$$

However, inserting equation (5) and (6) into (11) leads to analytically intractable integrals. Gao *et al.* (2002) used the Laplace method for approximation, and the detailed proof is provided in their study.

$$\begin{aligned} \ln P(D) \approx & -C \sum_{\boldsymbol{x_i} \in X_C} L_\varepsilon(y_i - f^*(\boldsymbol{x}_i)) \\ & - \frac{1}{2} (\alpha_i^+ - \alpha_i^-)^T K_{SV,SV} (\alpha_i^+ - \alpha_i^-) \\ & + l \ln G(C, \varepsilon) - \frac{1}{2} \ln \det(2\pi K_{X_M, X_M}) \\ & + \sum_{\boldsymbol{x_i} \in X_M} \ln \frac{C}{|\alpha_i|(C - |\alpha_i|)} \end{aligned} \tag{12}$$

To find the 'best' parameters for the SVR problem, the logarithmic evidence $\ln P(D)$ from different combinations of parameters are compared. Under this evidence framework, the set of parameters which give the optimal logarithmic evidence is determined to form the 'best' model. The number of support vectors required is implicitly fixed by the parameter selection under the Bayesian framework. In the study of Gao *et al.* (2002), this evidence formula is differentiated with respect to the parameter of interest and set to zero in order to find the optimal parameters. However, we realize that finding the gradient of equation (12) is nontrivial. The count and location of support vectors are implicitly related to the set of parameters selected, which implies that the terms that include the kernel K_{X_M,X_M}, $K_{SV,SV}$ or the loss function $\sum_{\boldsymbol{x_i} \in X_C} L_\varepsilon(y_i - f^*(\boldsymbol{x}_i))$ must be taken into account when taking the derivatives, and these are not considered in the original study. Such gradient approach to find the optimal parameters may not be practical, and we suggested using other optimization methods to maximize the log evidence of the model.

3.2. Error bar estimation

Gao *et al.* (2002) also introduced an error bar estimate for the SVR predictions under the evidence framework described in the previous section. It was suggested that the prediction error is made up of two terms. One of them is from the posterior uncertainty of the model $\boldsymbol{f}$, and the other is due to random noise in the data. The prediction model of a new test data point z is defined to be:

$$t = \boldsymbol{f}(z) + e(z) \tag{13}$$

where t is the target prediction, $\boldsymbol{f}(z)$ is the model output, and $e(z)$ is random noise, which is independent of $\boldsymbol{f}(z)$. The goal is to find the variance of the target t on the new test data point z.

Gao *et al.* (2002) derives the estimate of the variance due to model uncertainty, let $\boldsymbol{f}(X)$ be the trained model with training data X, and the predictive distribution of $\boldsymbol{f}(z)$ corresponding to a new data point z is defined to be

$$\begin{aligned} P(\boldsymbol{f}(z)|D) &= \frac{1}{P(D)} \int p(D|\boldsymbol{f}(X)) P(\boldsymbol{f}(X), \boldsymbol{f}(z)) \mathrm{d}\boldsymbol{f}(X) \\ &\propto \int p(D|\boldsymbol{f}(X)) P(\boldsymbol{f}(X), \boldsymbol{f}(z)) \mathrm{d}\boldsymbol{f}(X) \end{aligned} \tag{14}$$

The covariance matrix K of the training dataset $X = X_M \cup \bar{X}_M$ is defined as

$$K_{X,X} = \begin{pmatrix} K_{X_M,X_M} & K_{X_M,\bar{X}_M} \\ K^T_{X_M,\bar{X}_M} & K_{\bar{X}_M,\bar{X}_M} \end{pmatrix}$$

and the covariance matrix between the training data X and the new test data z is

$$K_{[X,z],[X,z]} = \begin{pmatrix} K_{X,X} & K_{X,z} \\ K^T_{X,z} & K_{z,z} \end{pmatrix}$$

Similar to the evidence approximation derivation, Gao *et al.* (2002) again suggested to use the Laplace method to approximate the integral in equation (14). While the detailed steps can be found in their study, the predictive distribution has the form

$$\begin{aligned} P(\boldsymbol{f}(z)|D) \propto \exp \Big\{ & -\frac{1}{2} \big(\boldsymbol{f}(z) - \boldsymbol{f}(z)^*\big) \\ & \times \Big(K_{z,z} - K^T_{X_M,z} K^{-1}_{X_M,X_M} K_{X_M,z}\Big)^{-1} \\ & \big(\boldsymbol{f}(z) - \boldsymbol{f}(z)^*\big) \Big\} \end{aligned}$$

which has the form of a Gaussian distribution with mean $\boldsymbol{f}^*(z)$, and variance

$$\sigma^2_{\hat{f}}(z) = K_{z,z} - K^T_{X_M,z} K^{-1}_{X_M,X_M} K_{X_M,z} \tag{15}$$

Regarding the variance of the random noise $e(z)$, Gao *et al.* (2002) made use of the noise model suggested in equation (5), which is interpreted as $P(e(z)) \propto \exp\{-C\, L_\varepsilon(e(z))\}$ with

$$E(e(z)) = 0 \quad \text{and } \mathrm{Var}(e(z)) = \sigma^2_e(z) = \frac{2}{C^2} + \frac{\varepsilon^2(\varepsilon C + 3)}{3(\varepsilon C + 1)} \tag{16}$$

Considering the prediction model suggested earlier in equation (13), the model and noise components are additive. By combining equation (15) and (16), the prediction variance is

$$\begin{aligned} \sigma^2_t(z) &= \sigma^2_{\hat{f}} + \sigma^2_e \\ &= K_{z,z} - K^T_{X_M,z} K^{-1}_{X_M,X_M} K_{X_M,z} \\ &\quad + \frac{2}{C^2} + \frac{\varepsilon^2(\varepsilon C + 3)}{3(\varepsilon C + 1)} \end{aligned} \tag{17}$$

3.3. New scaling parameter μ

The effort of deriving this evidence framework by Gao *et al.* (2002) is highly appreciated. Not only does it give a very good methodology to choose the 'best' parameters, but also it provides an error bar estimate of the SVR prediction. This is implemented in our study, but the results were rather unsatisfying, which lead to our idea of introducing a new scaling parameter μ to the evidence framework. The problems observed, as well as the rationale behind this new parameter μ is explained in this section.

Let us begin this discussion by cracking open the final logarithmic evidence formula derived by Gao *et al*. (2002) as given in equation (12). It is not difficult to notice that the first part of the formula is equivalent to negative of the the optimal value from the objective function in the SVR optimization problem.

$$\begin{aligned}\ln P(D) &\approx -(\text{optimal objective value of SVR optimization problem}) \\ &\quad + l \ln \frac{C}{2(\varepsilon C+1)} - \frac{1}{2}\ln\det(2\pi K_{X_M,X_M}) \\ &\quad + \sum_{\boldsymbol{x_i}\in X_M} \ln \frac{C}{|\alpha_i|(C-|\alpha_i|)} \end{aligned} \tag{18}$$

The level of the optimal objective value heavily depends two main factors: the model structure, and the level of the slack variables ξ_i^+, ξ_i^-. While the model structure is what we want to determine through the evidence framework, the level of slack variables has huge impact on the parameters in order to compute the 'best' model. For example, if the range of the target variable in the training data is wider than the one of the kernel features, which is quite common because inputs are usually normalized, the level of the slack variables is high. In such case, the level of the parameter C has to be large in order to bring up the level of the kernel features to match the target variable given the $0 \le \alpha_i^+, \alpha_i^- \le C$ constraint in the optimization problem. Therefore, the optimal objective value of the optimization problem is inflated. When observing the remaining terms in the logarithmic evidence formula, there are no other terms to balance such effect. This implies that if the data experiment is set up such that the level of the target variable is much higher than the features, the evidence formula may not be able to handle the scale difference. As a result, the evidence formula may be dominated by the objective value of the optimization problem, and therefore not be able to select the 'best' model.

The same scaling effect is also observed in the error bar estimation. Equation (15) provides a very nice formula to estimate the variance of the SVR prediction. However, it is solely derived from the kernel features. This means that if the range of the target variable is different from the one of the kernel features, the prediction error obtained from the formula is only going to be proportional to the estimated variance.

Given the above observations, we introduces a new scaling parameter μ to the evidence framework, aiming to handle such effect caused by the data, with a view to enhance the evidence framework. This scaling parameter is simply a scalar to inflate the kernel such that

$$\widetilde{K} = \mu \cdot K$$

where $\mu > 0$ is a scalar interpreted as *a priori* of the output variance, and K is the kernel matrix in the SVR problem. First of all we show that adding such a parameter results in the same SVR quadratic programming problem. Consider the dual problem of SVR given in equation (3), scaling the kernel K with the parameter μ result in the following:

$$\begin{aligned}\max_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-} \widetilde{W}(\widetilde{\boldsymbol{\alpha}}^+,\widetilde{\boldsymbol{\alpha}}^-) &= -\epsilon\sum_{i=1}^{n}(\widetilde{\alpha}_i^+ + \widetilde{\alpha}_i^-) + \sum_{i=1}^{n} y_i(\widetilde{\alpha}_i^+ - \widetilde{\alpha}_i^-) \\ &\quad - \frac{1}{2}\sum_{i,j=1}^{n}(\widetilde{\alpha}_i^+ - \widetilde{\alpha}_i^-)(\alpha_j^+ - \widetilde{\alpha}_j^-)\widetilde{K}(\boldsymbol{x_i},\boldsymbol{x_j}) \\ \text{subject to } 0 \le \widetilde{\alpha}_i^+, \widetilde{\alpha}_i^- &\le \widetilde{C}, \quad \sum_{i=1}^{n}(\widetilde{\alpha}_i^+ - \widetilde{\alpha}_i^-) = 0 \end{aligned} \tag{19}$$

where $\widetilde{\alpha}_i = \frac{\alpha_i}{\mu}$, $\widetilde{C} = \frac{C}{\mu}$, and $\widetilde{W}(\widetilde{\boldsymbol{\alpha}}^+,\widetilde{\boldsymbol{\alpha}}^-) = \frac{1}{\mu}\cdot W(\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-)$. The new objective function $\widetilde{W}$ is simply a scaled version of the original W, which means that the optimization problem has exactly the same solution. Adding such a parameter does not change the MAP estimate computation, but has an impact when introduced into the evidence formulation. Consider the logarithmic evidence formula shown in equation (12), introducing μ results in the following.

$$\begin{aligned}\ln P(D) \approx & -\frac{1}{\mu}\Bigg[C\sum_{\boldsymbol{x_i}\in X_C} L_\varepsilon(y_i - f^*(\boldsymbol{x}_i)) \\ & - \frac{1}{2}(\alpha_i^+ - \alpha_i^-)^T K_{SV,SV}(\alpha_i^+ - \alpha_i^-)\Bigg] \\ & + l\ln\frac{C}{2(\varepsilon C+\mu)} - \frac{1}{2}\ln\det(2\pi\mu K_{X_M,X_M}) \\ & + \sum_{\boldsymbol{x_i}\in X_M}\ln\frac{\mu C}{|\alpha_i|(C-|\alpha_i|)} \end{aligned} \tag{20}$$

By simply observing the new evidence function, the first part of the formula, which is the optimal objective value of the SVR optimization as discussed earlier, is scaled by the parameter μ. However, μ is also in the remaining terms, so its impact to the overall evidence formulation is not obvious. Note that equation (20) is only used to demonstrate the impact of μ when comparing to the original evidence function. In practice, the parameters are implicitly altered once the kernel is scaled, and the new logarithmic evidence function for implementation is simply equivalent to the original one with the scaled kernel and parameters as below.

$$\begin{aligned}\ln P(D) \approx & -\widetilde{C}\sum_{\boldsymbol{x_i}\in X_C} L_\varepsilon(y_i - f^*(\boldsymbol{x}_i)) \\ & - \frac{1}{2}(\widetilde{\alpha}_i^+ - \widetilde{\alpha}_i^-)^T \widetilde{K}_{SV,SV}(\widetilde{\alpha}_i^+ - \widetilde{\alpha}_i^-) \\ & + l\ln G(\widetilde{C},\varepsilon) - \frac{1}{2}\ln\det(2\pi\widetilde{K}_{X_M,X_M}) \\ & + \sum_{\boldsymbol{x_i}\in X_M}\ln\frac{\widetilde{C}}{|\widetilde{\alpha}_i|(\widetilde{C}-|\widetilde{\alpha}_i|)} \end{aligned} \tag{21}$$

Same as the original evidence framework, the parameters with the largest evidence value is chosen to give the 'best' model. On top of the original parameters, the 'best' scaling parameter μ is also determined through maximizing the evidence value. We test this altered evidence framework in our experiment, and the performance is found to be much better than the original evidence derivation. The results of the experiments are shown in section 5. Similarly, this new parameter μ also has its influence on the error bar estimation. Consider introducing μ to the original error bar estimation as shown in equation (17), the new error bar estimate becomes:

$$\begin{aligned}\widetilde{\sigma}_t^2(z) &= \widetilde{\sigma}_f^2 + \widetilde{\sigma}_e^2 \\ &= \mu\left(K_{z,z} - K_{X_M,z}^T K_{X_M,X_M}^{-1} K_{X_M,z}\right) \\ &\quad + \frac{2\mu^2}{C^2} + \frac{\varepsilon^2(\varepsilon C + 3\mu)}{3(\varepsilon C + \mu)}\end{aligned} \tag{22}$$

From simply observing the new error bar estimate in equation (22), the prediction error is scaled by the parameter μ when comparing to the original formulation. Therefore, we interpret μ as the *a priori* variance of the output to scale the kernel features to match the variance of the target variable. Similarly, equation (22) is only provided for the purpose of illustrating the impact of μ when comparing to the original error bar estimate. In practice, the parameters are implicitly altered once the kernel is scaled, and the new error bar formulation for implementation is simply equivalent to original one with the scaled kernel and parameters as follows, and the results are discussed in section 5.

$$\begin{aligned}\widetilde{\sigma}_t^2(z) &= \widetilde{\sigma}_f^2 + \widetilde{\sigma}_e^2 \\ &= \widetilde{K}_{z,z} - \widetilde{K}_{X_M,z}^T \widetilde{K}_{X_M,X_M}^{-1} \widetilde{K}_{X_M,z} \\ &\quad + \frac{2}{\widetilde{C}^2} + \frac{\varepsilon^2(\varepsilon\widetilde{C} + 3)}{3(\varepsilon\widetilde{C} + 1)}\end{aligned} \tag{23}$$

Preprocessing the target variable may be an alternative approach to the problem, but it requires careful reverse engineering of the output to ensure appropriate interpretability. Building the new parameter into the Bayesian framework avoid this extra step, and therefore fit better to the idea of developing an automated flexible prediction framework.

4. Bayesian optimization for parameters selection

In section 3.1, we demonstrated that using the gradient method to optimize the log evidence is impractical, and seeking for another optimization method is necessary. Recently, the popularity of Bayesian optimization has grown significantly in the artificial intelligence and machine learning community (Srinivas *et al*. 2010, Shahriari *et al*. 2016). Its capability to actively and efficiently search for the global optimum for a complex or even unknown function greatly enhance the automatic tuning process for many powerful machine learning algorithms (Snoek *et al*. 2012). In this study, we borrow such technique to optimize the log evidence function in order to obtain the 'best' parameters. Bayesian optimization is explained in this section, and are used in our experiments.

4.1. Multi-armed Bandit

Multi-armed bandit (MAB) problem is a problem in which a gambler faces k slot machines, a.k.a. 'one-armed bandits'. Each machine has unknown probability of winnings. The gambler is allowed to play one machine each round, and eventually has to define strategy to maximize the total winnings. The key here is to trade-off between exploration (try a new machine), and exploitation (continue to play with an already observed winning machine). This problem has been widely studied in various areas such as machine learning, operational research and control etc.

Recently, the task of globally optimizing a complex, sometimes unknown function is being formulated in a MAB setting (Srinivas *et al*. 2010). The problem becomes a sequential optimization of an objective function $f: \boldsymbol{D} \to \mathbb{R}$, considering each input as one of k arms in the data $\boldsymbol{D}$. In each time step t, an input $\boldsymbol{x_t} \in D$ is chosen, and observe the corresponding noisy function value $y_t = f(\boldsymbol{x_t}) + \delta_t$, where δ_t is the noise perturbation. The goal is to maximize the 'sum of rewards' $\sum_{t=1}^{T} f(\boldsymbol{x_t})$ as rapidly as possible, which is essentially searching for $\boldsymbol{x}^* = \text{argmax}_{\boldsymbol{x}\in D} f(\boldsymbol{x})$.

To efficiently perform the trade-off between exploration and exploitation, the dependencies across arms are assumed and modelled (Dorard *et al*. 2009). Such dependencies allow exploration to be faster. When an arm is explored, knowledge is gained on that arm as well as similar arms. The rewards of arms are assumed to be correlated, which means that the resulting rewards are similar if the arms pulled are similar. The correlations are modelled by assuming that f is a function drawn from a Gaussian Process (GP).

The GP prior distribution is initialized with mean μ_0 and the correlation between arms are described by the GP covariance matrix. Each entry (i, j) of the matrix specifies the covariance between arm i and j. The covariance can be defined using a kernel to measure the similarity between data points. After each inputs point is drawn, the corresponding reward is observed, and the GP posterior distribution is updated. An acquisition function is then used to determined the next point to examine. The algorithm iterates until a near optimal value is reached. A few common acquisition functions are listed in section 4.3. Figure 2 provides a clear overview of the process, and the pseudo code is listed in algorithm 1.

Algorithm 1: Pseudo-code for Bayesian optimization

Input: Input space $\boldsymbol{D_0}$; GP Prior μ_0, Σ_0
for $\boldsymbol{t} = 1, 2, \ldots$ **do**
  Select new $\boldsymbol{x_{t+1}}$ by optimizing acquisition function α
    $\boldsymbol{x_{t+1}} = \underset{x\in D_t}{\text{argmax}}\ \alpha(\boldsymbol{x}; D_t)$
  Query objective function to obtain y_{t+1}
  Augmented data $\boldsymbol{D_{t+1}} = \{D_t, (\boldsymbol{x_{t+1}}, y_{t+1})\}$
  Perform Bayesian update to obtain μ_{t+1} and Σ_{t+1}
end

4.2. Gaussian process

GP is a widely used method in machine learning. It defines a distribution over the function f, which maps some input space X to $\mathbb{R}$. For n input data points $x_{1:n}$, the corresponding function values and noisy observations are respectively $f_i := f(\boldsymbol{x_i})$, and y_i, with $i = 1, 2, \ldots, n$. In the GP regression setting, the function value $\boldsymbol{f} := f_{1:n}$ is assumed to be jointly Gaussian with mean $\boldsymbol{m}$ and covariance $\boldsymbol{K}$.

$$P(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}) \tag{24}$$

Suppose noisy observations are samples from the distribution over the function perturbed by noise $\delta_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)$. The noisy observations $\boldsymbol{y}$ are normally distributed given $\boldsymbol{f}$, and the likelihood is

$$P(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i=1}^{n} \mathcal{N}(f_i, \sigma^2) = \mathcal{N}(\boldsymbol{f}, \sigma^2\boldsymbol{I}) \tag{25}$$

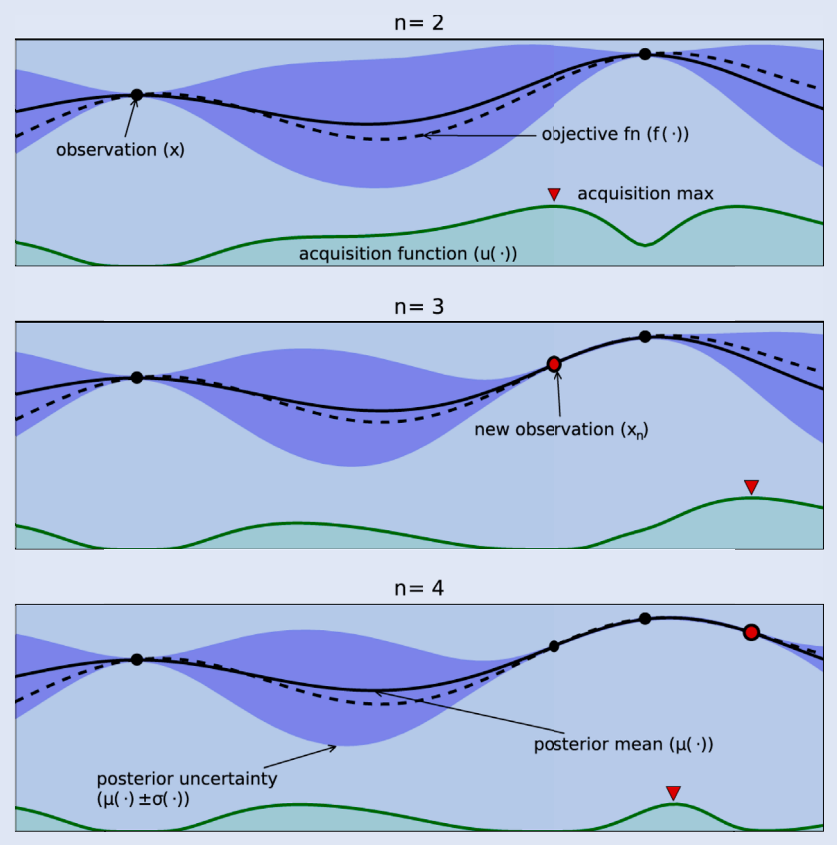

Figure 2. GP Bayesian optimization (Shahriari *et al*. 2016).

Equation (24) represents the prior distribution $p(\boldsymbol{f})$ with mean $m_i := \mu_0(\boldsymbol{x_0})$ and $K_{i,j} = k(\boldsymbol{x_i}, \boldsymbol{x_j})$, and $K_{i,j}$ is a positive definite kernel to measure the covariance between point i and j. The likelihood is given in equation (25). The posterior distribution is proportional to the product of equations (24) and (25).

$$P(\boldsymbol{f}|\boldsymbol{y}, X) \propto P(\boldsymbol{y}|\boldsymbol{f})\, P(\boldsymbol{f}, X) \tag{26}$$

Since the function is assumed to be jointly Gaussian, the joint distribution over the observations $\boldsymbol{y}$ and a new test point f^* can be defined as

$$P\begin{pmatrix}\boldsymbol{y}\\ f^*\end{pmatrix} = \mathcal{N}\left(\begin{matrix}\boldsymbol{m}\\ \mu(\boldsymbol{x}_*)\end{matrix}, \begin{bmatrix}K(X,X)+\sigma^2\boldsymbol{I} & K(X,\boldsymbol{x}_*)\\ K(\boldsymbol{x}_*,X) & K(\boldsymbol{x}_*,\boldsymbol{x}_*)\end{bmatrix}\right) \tag{27}$$

Given the conditioning property of a joint Gaussian distribution, the predictive posterior distribution for the test point f^* is derived to be

$$\begin{aligned}P(f^*|\boldsymbol{x}_*, \boldsymbol{y}, X) &= \int P(f^*|\boldsymbol{f}, \boldsymbol{x}_*, \boldsymbol{y}, X)\, P(\boldsymbol{f}|\boldsymbol{y}, X)\, \mathrm{d}\boldsymbol{f}\\ &= \mathcal{N}\left(\mu^*, \Sigma^*\right)\end{aligned}$$

where,

$$\begin{aligned}\mu^* &= \mu(\boldsymbol{x}_*) + K(\boldsymbol{x}_*, X)\left(K(X,X)+\sigma^2\boldsymbol{I}\right)^{-1}(\boldsymbol{y}-\boldsymbol{m})\\ \Sigma^* &= K(\boldsymbol{x}_*,\boldsymbol{x}_*) - K(\boldsymbol{x}_*,X)\left(K(X,X)+\sigma^2\boldsymbol{I}\right)^{-1}K(X,\boldsymbol{x}_*)\end{aligned} \tag{28}$$

In the context of Bayesian optimization as described in section 4.1, the process begins by fitting a GP on the function values from a small subset of inputs, and sequentially add points to the input set while updating the posterior distribution. The Bayesian updating process iterates until the function optimum is reached. The order of sequence in selecting input points is decided by the acquisition function that is going to be discussed in section 4.3.

4.3. Acquisition method

The acquisition function, which is also the expected utility in decision theory, is the key to define the order of sequence in selecting input points. It is designed to trade off exploration of the search space and exploitation of the currently known area. The inputs that correspond to the optimum of the acquisition function are usually the location of the next input. Some widely used acquisition function in the current academic and industrial research including but not limited to probability of improvement (PI), expected improvement (EI), upper confidence bound (UCB), and Thompson sampling (TS).

PI is a policy that makes use of the mean μ_n and variance σ_n of the GP posterior distribution to define the highest probability in leading to an improvement over the best current value $f(\boldsymbol{x}_{\text{best}})$, thus, becomes the next input to be selected in the sequence. Φ is the standard normal cumulative distribution function.

$$\begin{aligned}\alpha_{PI}(\boldsymbol{x}; D_n) &= \Phi(\gamma(\boldsymbol{x}))\\ \gamma(\boldsymbol{x}) &= \frac{\mu_n(\boldsymbol{x}) - f(\boldsymbol{x}_{\text{best}})}{\sigma_n(\boldsymbol{x})}\end{aligned}$$

While PI is proven to exploit quite aggressively in some cases (Bergstra and Bengio 2012), EI is a great alternative.

Because the random variable $f(\boldsymbol{x})$ is normally distributed, the expectation can be computed analytically (Brochu *et al*. 2010). ϕ is defined to be the standard normal probability density function.

$$\alpha_{EI}(\boldsymbol{x}; D_n) = \begin{cases} (\mu_n(\boldsymbol{x}) - f(\boldsymbol{x}_{\text{best}}))\, \Phi(\gamma(\boldsymbol{x})) + \sigma_n(\boldsymbol{x})\, \phi(\gamma(\boldsymbol{x})), & \text{if } \sigma_n > 0. \\ 0, & \text{if } \sigma_n = 0. \end{cases}$$

UCB is relatively more recent approach to trade off exploration and exploitation (Srinivas *et al*. 2010). It is constructed to minimize regret for the optimization. κ is a user-defined parameter to balance exploration against exploitation.

$$\alpha_{UCB}(\boldsymbol{x}; D_n) = \mu_n(\boldsymbol{x}) + \kappa \sigma_n(\boldsymbol{x})$$

While conceptually different from other acquisition methods mentioned, recently TS has gained wide interest among the MAB community. Multiple theoretical and empirical evaluations have been performed (Chapelle and Li 2011, Scott 2010, Agrawal and Goyal 2013, Kaufmann *et al*. 2012, Russo and Van Roy 2014), and there are claims about its advantage over the other methods. A sample of the reward function is drawn from the posterior distribution, and the arm with the highest simulated reward is selected. This is the method employed in the experiments of this study.

$$\alpha_{TS}(\boldsymbol{x}; D_n) = f^{(n)}(\boldsymbol{x})$$
$$\text{where } f^{(n)} \sim GP(\mu_0, k|D_n)$$

5. Experiments

SVR has long been proven to have good prediction performance. The focus of this study is to outline a Bayesian probabilistic framework for the algorithm mainly for two purposes. (1) Multi-fold cross validation process is well known to be an effective method to select parameters for machine learning algorithms, but is usually computationally expensive. Borrowing the MAP approach from Bayesian statistics, it may be possible to achieve similar performance as the original SVR with less computation to select the 'best' parameters. (2) Introducing the probabilistic framework allows assessment of prediction uncertainties. This is extremely important especially in the financial context, where decisions are better based on predictions with some level of confidence.

Before describing the implementation details, it may be useful to provide an overview of the prediction system to bring the proposed methodologies together. Figure 3 presents the complete setup of the prediction process. The SVR model is first trained with the parameters defined (i.e. CV, Grid Search or BayesOpt). Under the Bayesian framework, while making use of the model output, the predictive mean and variance are then computed. The quality of the prediction is evaluated using the corresponding error bar. This, at the same time, is interpreted as an assessment of whether the market condition has changed. This complete process is iterated in a rolling window manner.

5.1. Data

Eight financial time series from various asset classes (i.e. Equity Index, CDS spreads, Bond yields, Commodity Futures) are used in this study to assess the performance and flexibility of the Bayesian SVR framework proposed in this study. The details of each time series are listed in appendix 1. They are gathered from the Reuters Eikon platform at the Thomson-Reuters Laboratory, University College London. Daily quotes are used in the analyses. The time frame used for each time series may vary slightly depending on the availability of data.

5.2. Implementation

In this first set of experiments, the original RBF kernel SVR algorithm is first trained and tested with parameters selected through five-fold cross validation. The MATLAB package *libsvm* (Chang and Lin 2011) is used for SVR implementation. The results are used as the benchmark performance when comparing to the ones from the Bayesian SVR where parameters are selected through the MAP approach. The following combination of parameters is used for all experiments that involve cross validation procedure to select parameters.

$$C = 2^{(-3,-1,1,3,5,7,9,11,13,15)} \qquad \varepsilon = 2^{(-9,-8,-7,-6,-5,-4,-3)}$$
$$\gamma = 2^{(-8,-7,-6,-5,-4,-3,-2,-1)}$$

The original Bayesian probabilistic framework by Gao *et al*. (2002) is implemented. However, for the theoretical reasons explained in section 3.3, the results are poor. We have therefore not included these results. The same sets of data are then processed using the amended Bayesian probabilistic framework with parameters selected by optimizing the logarithmic evidence formula suggested in equation (21). As mentioned previously, the gradient method to optimize the log evidence is impractical, and seeking for another optimization method was necessary. The easiest optimization method is grid search. Although grid search is still quite computationally expensive, and its performance is highly dependent on the design of the grid, it is very easy to implement and the results are easy to assess. In section 3.3, we introduces the new parameter μ to scale the RBF kernel K. This new parameter is incorporated into the grid search process in maximizing the evidence. Note that because the parameter μ is a factor that scales the kernel, we control the kernel width parameter γ to be a fixed small value in order to avoid the interactive effect of these two parameters within the probabilistic framework. In our experiments, the small γ value chosen works consistently, and the same value is used in all experiments. The parameters C and ε values are the ones considered in the cross validation process.

$$C = 2^{(-3,-1,1,3,5,7,9,11,13,15)} \qquad \varepsilon = 2^{(-9,-8,-7,-6,-5,-4,-3)}$$
$$\mu = 2^{(-3,-1,1,3,5,7,9,11,13,15,17,19)} \qquad \gamma = 2^{(-8)}$$

Lastly, a similar exercise is carried out with the logarithmic evidence formula being optimized with the multi-armed bandit Bayesian optimization approach, aiming to automate and accelerate the parameter selection process, while achieving comparable performance. The MATLAB package *bayesopt* (Martinez-Cantin 2014) is used. TS is used as the acquisition method for our experiments, and observation noise (σ^2 in equation (28)) for the GP is chosen to be a very small number (1e-14) to reflect the fact that the log evidence is a deterministic function. For Bayesian optimization, only the upper and lower bound of the parameter search space is required.

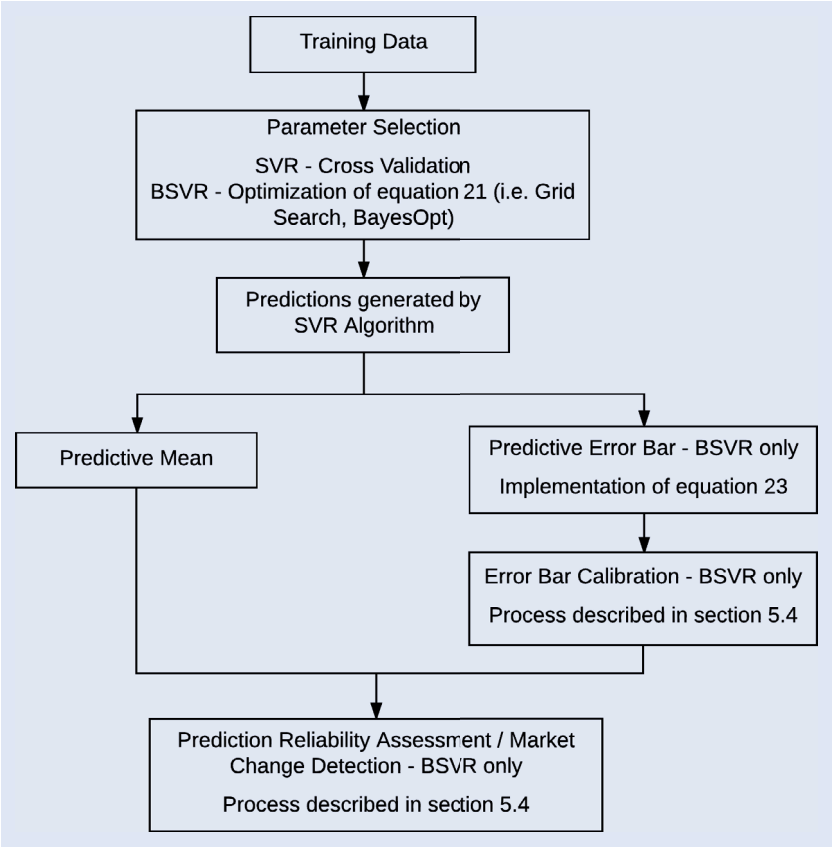

Figure 3. Overview of the proposed methodologies setup.

$$C = 2^{(-3:15)} \qquad \varepsilon = 2^{(-9:-3)}$$
$$\mu = 2^{(-3:19)} \qquad \gamma = 2^{(-8)}$$

To reduce the computational cost, algorithms are trained with 2.5 years of data (630 quotes), and tested on the following 0.5 year (125 quotes), assuming 252 trading days in a year. Predictions are performed and assessed in a rolling window manner to cover the entire time series.

In the second set of experiments, only Bayesian SVR with Bayesian optimization selected parameters is examined. Because of the reduction in computational cost for the parameters selection process, the algorithm is trained with 3 months of data (63 quotes), and tested daily. Providing the rapid-changing nature of financial time series, this greatly enhances the sensitivity of predictions and their uncertainty measures. The error bar estimates are computed using equation (23). These are closely examined and practical applications are discussed.

In all the above mentioned experiments, quotes from the last 14 days are chosen to capture the patterns in order to predict a one day ahead quote. The same 14-day window was used in another study (Gündüz and Uhrig-Homburg 2011), which we follow the same approach. However, the Bayesian SVR framework proposed is generalizable to use any reasonable features. This study aims to define a flexible framework for the algorithm rather than to provide the best features for predictions. It is also important to mention that each input variable in the training samples are linearly scaled to the range [0, 1] by

$$z_{ij} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)}$$

where x_{ij} is the jth sample of the ith input variable, and z_{ij} is the corresponding scaled input value. The testing samples are also scaled in the same manner use the maximum and minimum from the training set.

Mean Absolute Percentage Errors (MAPE) are used to assess the prediction performance.

$$\text{MAPE}(\%) = \frac{1}{T}\sum_{t=1}^{T}\frac{\left|\hat{y}_t^{pred} - y_t^{obs}\right|}{y_t^{obs}} \tag{29}$$

where $\hat{y}_t^{pred}$ is the predicted quote of the asset on day t, and $t = 1, \ldots, T$ is the number of prediction days. This metric is computed in all experiments to compare prediction performances.

5.3. Results

In the first set of experiments, the original RBF kernel SVR algorithm is first trained and tested with parameters selected through five-fold cross validation. The results are then used as the benchmark performance when comparing to the ones from the Bayesian SVR where parameters are selected through the evidence maximization approach using grid search or Bayesian optimization. MAPEs (equation (29)) are computed to assess the predictive performance of each model for each times series. Algorithms are trained with 2.5 years of data (630 quotes), and test on the following 0.5 year (125 quotes), assuming 252 trading days in a year. Predictions are performed and assessed in a rolling window manner to cover the entire time series.

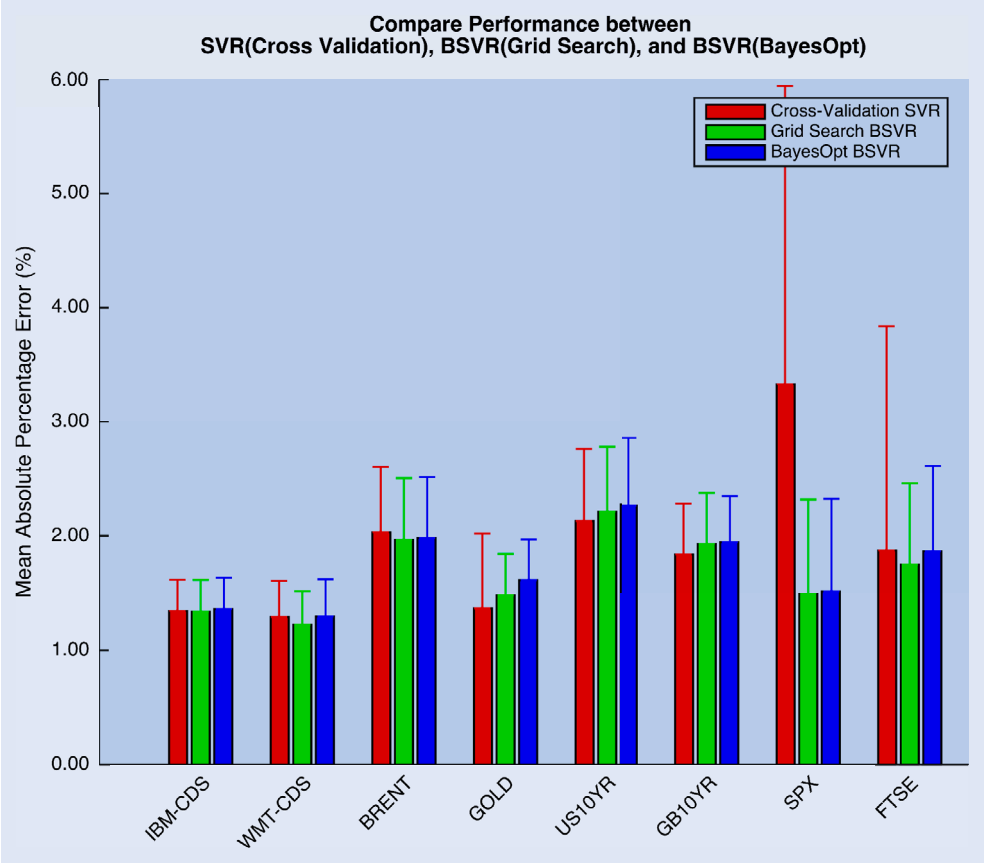

Figure 4. MAPE for SVR(Cross Validation), BSVR(Grid Search), and BSVR(BayesOpt).

The bars in figure 4 show the averaged MAPEs for all the rolling windows, and the arm describes the corresponding coefficient of variation (CoV) which is computed by dividing the mean by the standard deviation (SD). CoV acts as the SD standardized by the level of the mean, which allows comparison across different models and assets. The longer CoV arms suggest prediction performance inconsistencies in some of the rolling windows.

Figure 4 shows that the predictive performance varies across assets mainly due to difference in individual behaviour such as volatility level, regime change etc. The MAPE values are provided in appendix 2. It is interesting to observe that with S&P500 (SPX), the original SVR with five-fold cross validation shows inconsistent prediction performance. It may be a sign suggesting, in this case, cross-validation is incapable to generalize the training data distribution to the testing data. It may be fixed by fine tunes such as changing the size of training or testing window, using different number of folds in the cross-validation process etc. This is purposely left unfixed to ensure the consistency in our experiments as well as to demonstrate that evidence maximization approach in Bayesian SVR may sometimes allow better parameter selection than cross validation. Other than SPX, the models generated from the three different parameters selection methods give comparable performance in most of the cases. This is encouraging as it suggests that it may be worthwhile to generalize SVR to Bayesian SVR to gain the additional features such as a faster and easier parameter selection process, as well as an estimate of prediction uncertainty, without sacrificing much predictive performance.

Results from the last experiment provide initiatives to spend more effort in examining further the performance and use of Bayesian SVR. It is shown that Bayesian optimization accelerates the log evidence optimization process while retaining similar performance as using grid search. Hence, in the second set of experiments, only Bayesian SVR with Bayesian optimization selected parameters is applied. S&P500 Equity Index is used as an example to demonstrate the practical use of this Bayesian SVR framework. Since the computational cost for the parameters selection process has been significantly reduced, we are able to increase the number of iterations by training the algorithm with three months of data (63 quotes), and testing daily. This greatly enhances the sensitivity of predictions and their uncertainty measures in better adapting to the rapid-changing nature of financial time series.

The S&P500 Equity Index time series is examined. The prediction error drops from roughly 1.5% to 1.2%. While this is quite encouraging, the more important findings are yet to be discussed. In figure 5, the top graph shows the actual vs. predicted quotes as well as the predictive variance. The middle graph provides the log return of the daily asset quotes to display the volatility in different regimes. At the bottom, the graph plots the prediction error of everyday quotes. It is quite obvious that the Bayesian SVR predictions are less reliable during the high volatility regimes. This matches expectation as the data distribution may not be well represented in the training data during high volatility circumstances. Therefore, it is important to take into account prediction uncertainties to assess the quality of predictions.

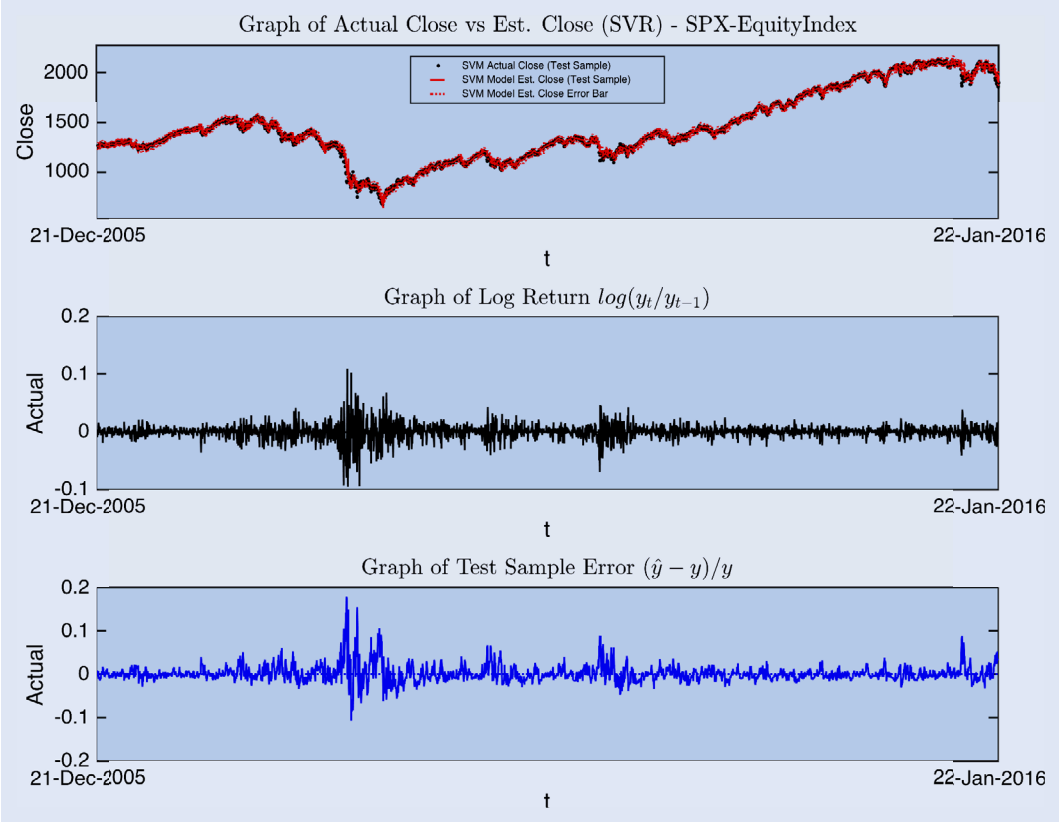

Figure 5. Actual quotes vs predicted quotes (S&P500 Equity Index).

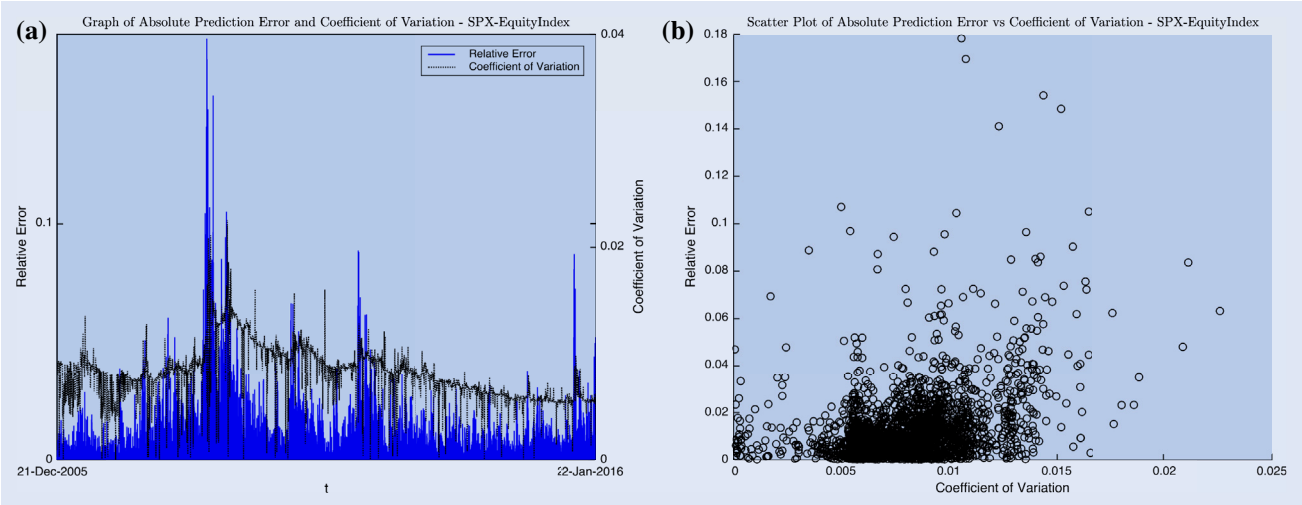

Figure 6. The absolute relative error and the prediction CoV (S&P500 Equity Index).

As mentioned previously, another useful functionality of Bayesian SVR is prediction uncertainty estimation. In the top graph of figure 5, some dotted lines are found. These are the error bar estimates for the SVR predictions provided in equation (23). These are used to assess the level of confidence for the predictions made by the Bayesian SVR algorithm. Given the scale of the figure, the relationship between the error bar estimates and the corresponding predictive performance can hardly be observed. A better demonstration is provided in figure 6.

In figure 6(a), the absolute value of the relative errors ($|Predicted/Actual-1|$) is the vertical bars plotted on the left axis, and the prediction coefficient of variance (σ_{pred}/μ_{pred}) is the dotted line overlaid on the right axis. The coefficient of variance is used instead of the error bar estimates to account for the different level of prediction mean across the time series. It is easy to see that when the coefficient of variance is high, the prediction errors are higher. Similarly, figure 6(b) shows a scatter plot of the absolute value of the relative errors against the coefficient of variance, and the Pearson correlation is 34.06% with p-value close to zero. This suggests a significant relationship between size of the error bars and the prediction quality. In the financial context, when higher coefficient of variance is observed, the quote prediction may not be as reliable, and

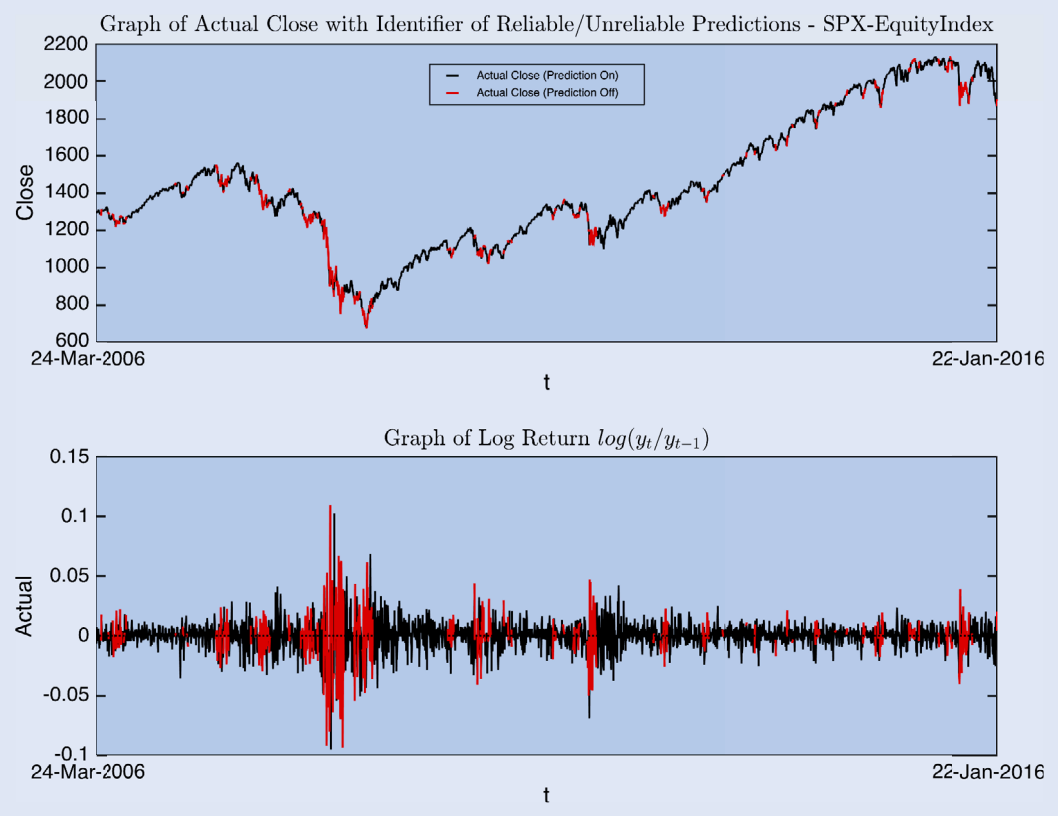

Figure 7. Actual Quotes with identifier of 'Reliable' predictions (S&P500 Equity Index).

careful considerations should be made before basing decisions on such predictions.

5.4. Calibration

It is definitely useful to have such prediction uncertainty estimates. However, again because of the rapid changing nature of financial time series, it is not easy to identify one coefficient of variation cut-off value that is suitable to consistently distinguish 'reliable' predictions over time. It is equally difficult to identify such a value across different time series. Therefore, we introduce an automated calibration procedure to hopefully define this cut-off value adaptively in order to best utilize the prediction uncertainty estimates.

We begin by defining a rolling window procedure similar to the one in the previous experiments. In this case, to be consistent, three months of data (63 quotes) are used as training data, and are applied one day forward. The predictions generated in the last three months are assessed. A criterion is arbitrary defined suggesting that only the predictions that have error less than the average daily change in quotes are considered 'reliable'. Among these prediction points, considering that the CoV and prediction error are not perfectively correlated, the cut-off value is set to be the 80th percentile of all CoV values to exclude outliers.

The closing quotes of the S&P500 Equity Index in the past 10 years are plotted in the top graph of figure 7. The prediction points that are considered 'unreliable' are coloured in red, while the rest are in black. The bottom graph shows the corresponding log return of the quotes as a demonstration of market volatility. It is quite easy to spot that the predictions are coloured in red when the training data are not reflecting the current market condition. This is particularly obvious during 2008 financial crisis, when there is high volatility and prediction is extremely difficult. Some red points are also spotted when there are sudden changes in the market. If the 'unreliable' predictions are neglected, the prediction error drops further to about 1.1%, while keeping 77% of the predictions. The calibration not only allows one to avoid basing decisions on unreliable predictions, it can possibly act as a market condition change detection tool. However, it is important to note that the algorithm, in its current form, is only capable to detect when the market condition deviates from the training period. Without further modifications, it is not a tool to cluster the time series into different regimes (i.e. 'high' or 'low' volatility etc.).

There are two user-defined parameters in this calibration procedure. The first parameter is the reliability criteria definition. We define predictions to be considered 'reliable' when error is less than the average daily quote changes in the last three month. However, this can be adjusted based on the user's preference. For example, if this prediction is used as part of a trading strategy, this reliability criteria can be defined as some minimum return of the strategy over a period of time. The second parameter is the quantile used to exclude the outliers in the coefficient of variation estimates. The parameter may vary, but can be easily optimized by setting an objective function to consider other factors such as the number of predictions to be included, the target magnitude of error reduction, etc. While this calibration process is efficient and simple to implement, other clustering methods (i.e. KNN, Naive Bayes, SVM, Logistic Regression etc.) may be considered to differentiate the 'reliable' and 'unreliable' predictions.

6. Conclusions and future extensions

With the intention to adopt the powerful SVR in predicting financial time series, this study introduces a framework to generalize the algorithm with a Bayesian approach. This gives a more efficient parameter selection procedure as well as a prediction uncertainty estimate. A Bayesian approach to SVR (Gao *et al*. 2002) is discussed, and implemented. It is found that direct implementation of the probabilistic framework proposed by Gao *et al*. (2002) returns poor results in our experiments. A novel enhancement is proposed by adding a new kernel scaling parameter μ to overcome the difficulties encountered. In addition, the multi-armed bandit Bayesian optimization technique is applied to automate the parameter selection process.

The framework is then tested on financial time series of various asset classes to ensure its flexibility. In the experiments, it is shown that the generalization performance of the model selected through our framework can reach or sometimes surpass the ones from the model selected through the computationally more expensive cross-validation process. While taking advantage of the reduction in computational cost, iterations are increased to generate daily predictions. It is shown with the S&P500 Equity Index that the prediction error has decreased, while generating a sensible prediction uncertainty estimate. An adaptive calibration process is then presented to demonstrate the practical use of the prediction uncertainty estimates to identify 'unreliable' predictions, which greatly enhances the prediction performance. The machine learning approach discussed in this study can be developed as a pricing tool, and possibly as a market condition change detector.

While the framework is quite effective as it is, there are a few directions that may justify further investigations. The multi-armed bandit optimization technique employed at the moment is sequential. Although it is quite efficient, it can be further improved by parallelizing the algorithm. This has been discussed in a few studies (Ginsbourger and Riche 2011 Snoek *et al*. 2012 Desautels *et al*. 2014). Also, as mentioned, the calibration process introduced is efficient and simple to implement. However, it may be interesting to find out if other clustering algorithms give better results, or possibly lead to a regime clustering algorithm. Furthermore, it may be interesting to relax the Gaussian prior distribution assumption of the probabilistic framework proposed by Gao *et al*. (2002) . A good example is to extend the SVR model to a Multi-Kernel Learning formulation which assumes a non-Gaussian prior distribution. However, this may lead to significant complexity in the reconstruction of the Bayesian evidence approximation. From the financial point of view, the prediction uncertainty estimates may be extended as inputs to build financial portfolios.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC).

ORCID

T. Law http://orcid.org/0000-0003-0482-6697
J. Shawe-Taylor http://orcid.org/0000-0002-2030-0073

References

Agrawal, S. and Goyal, N., Thompson sampling for contextual Bandits with linear payoffs. *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA, 2013, 127–135.

Bergstra, J. and Bengio, Y., Random search for hyper-parameter optimization. *J. Mach. Learn. Res*., 2012, **13**(1), 281–305.

Brochu, E., Cora, V. and de Freitas, N., A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. 2010, preprint, arXiv:1012.2599.

Cao, L.J. and Tay, F.E., Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Trans. Neural Networks*, 2003, **14**(6), 1506–1518.

Chang, C. and Lin, C., LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol*., 2011, **2**(3), 1–27.

Chapelle, O. and Li, L., An empirical evaluation of thompson sampling. *Adv. Neural Inf. Proc. Syst* 2011, **24**, 2249–2257.

Desautels, T., Krause, A. and Burdick, J., Parallelizing exploration-exploitation tradeoffs with Gaussian process bandit optimization. *J. Mach. Learn. Res*., 2014, **15**(1), 3873–3923.

Dorard, L., Glowacka, D. and Shawe-Taylor, J., Gaussian process modelling of dependencies in multi-armed bandit problems. *Proceedings of the 10th International Symposium on Operations Research*, Nova Gorica, Slovenia, 2009, 77–84.

Gao, J., Gunn, S., Harris, C. and Brown, M., A probabilistic framework for SVM regression and error bar estimation. *Mach. Learn*., 2002, **46**(1), 71–89.

Ginsbourger, D. and Riche, R., Dealing with asynchronicity in parallel Gaussian process based global optimization. 4th International Conference of the ERCIM WG on Computing & Statistics (ERCIM'11), London, UK, 2011.

Gündüz, Y. and Uhrig-Homburg, M., Predicting credit default swap prices with financial and pure data-driven appraoches. *Quant. Finance*, 2011, **11**(12), 1709–1727.

Kaufmann, E., Korda, N. and Munos, R., Thompson sampling: An asymptotically optimal finite-time analysis. *Algorithmic Learn. Theory* 2012, **7568**, 199–213.

Kim, K., Financial time series forecasting using support vector machines. *Neurocomputing*, 2003, **55**(1), 307–319.

Lu, C., Lee, T. and Chiu, C., Financial time series forecasting using independent component analysis and support vector regression. *Decis. Support Syst*., 2009, **47**(2), 115–125.

MacKay, D., Bayesian interpolation. *Neural Comput*., 1992, **4**(3), 415–447.

Martinez-Cantin, R., BayesOpt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits. *J. Mach. Learn. Res*., 2014, **15**(1), 3735–3739.

Mohri, M. and Rostamizadeh, A., Stability bounds for non-iid processes. *Adv. Neural Inf. Process. Syst*., 2008, **20**, 1025–1032.

Müller, K., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. Predicting time series with support vector machines. In *Artificial Neural Networks: ICANN'97*, Vol. 20, edited by W. Gerstner , A. Germond , M. Hasler and J.D. Nicoud, pp. 999–1004, 1997 (Springer: Berlin).

Ralaivola, L., Szafranski, M. and Stempfel, G., Chromatic PAC-Bayes bounds for non-iid data: Applications to ranking and stationary-mixing processes. *J. Mach. Learn. Res*. 2010, **11**, 1927–1956.

Russo, D. and Van Roy, B., Learning to optimize via posterior sampling. *Math. Oper. Res*., 2014, **39**(4), 1221–1243.

Scott, S., A modern Bayesian look at the multi-armed bandit. *Appl. Stochastic Models Bus Ind*., 2010, **26**(6), 639–658.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. and de Freitas, N., Taking the human out of the loop: A review of bayesian optimization. *Proc. IEEE*, 2016, **104**(1), 148–175.

Smola, A. and Schölkopf, B., A tutorial on support vector regression. *Statistics Comput*., 2004, **14**(3), 199–222.
Snoek, J., Larochelle, H. and Adams, R., Practical Bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst*., 2012, 2951–2959.
Srinivas, N., Krause, A., Kakade, S. and Seeger, M., Gaussian process optimization in the bandit setting: No regret and experimental design. *Proceedings of the 27th International Conference on Machine Learning, Haifa, Israel*, 2010, 1015–1022.
Tay, F.E. and Cao, L., Application of support vector machines in financial time series forecasting. *Omega*, 2001, **29**(4), 309–317.

Appendix 1. Financial time series dataset

Asset class	Symbol	Name	Start date	End date
Equity Index	FTSE	FTSE 100	02 September 2005	22 January 2016
Equity Index	SPX	S&P 500	02 September 2005	22 January 2016
Commodity Futures	BRENT	ICE BRENT Crude Oil Front Month Futures	02 September 2005	22 January 2016
Commodity Futures	GOLD	CME GOLD Front Month Futures	02 September 2005	22 January 2016
Bond Yield	GB10YR	UK Gilt 10YR	02 September 2005	22 January 2016
Bond Yield	US10YR	US Treasury 10YR	02 September 2005	22 January 2016
CDS Spread	IBM-CDS	INTERNATIONAL BUSINESS MACHINES CORPORATION 5YR	21 July 2008	21 January 2016
CDS Spread	WMT-CDS	WAL-MART STORES INC. 5YR	21 July 2008	21 January 2016

Appendix 2. MAPE (630 quotes training, 126 quotes testing)

Symbol	SVR (Cross-Validation)	BSVR (Grid Search)	BSVR (BayesOpt)
FTSE	1.88 (3.69)[†]	1.76 (1.25)	1.88 (1.40)
SPX	3.34 (8.69)	1.50 (1.24)	1.52 (1.23)
BRENT	2.03 (1.16)	1.96 (1.06)	1.98 (1.06)
GOLD	1.37 (0.89)	1.48 (0.54)	1.61 (0.58)
GB10YR	1.85 (0.82)	1.94 (0.86)	1.95 (0.79)
US10YR	2.13 (1.34)	2.21 (1.25)	2.27 (1.35)
IBM-CDS	1.34 (0.36)	1.34 (0.37)	1.36 (0.37)
WMT-CDS	1.29 (0.41)	1.22 (0.36)	1.29 (0.42)

[†]Average of MAPEs from all rolling windows with standard deviation within parentheses.