



METHOD AND SYSTEM TO PROVIDE A FORECASTING MODEL

by Inventors

Background of the Invention

1. Field of the Invention

The present invention relates generally to forecasting. More particularly, the present invention relates to a method and system to construct a forecasting model.

2. Description of the Related Art

Forecasting refers to the prediction or calculation of a future event or condition based on a study or analysis of known data based on past events. While forecasting is constantly performed by individuals to predict common events such the price of groceries or the amount of traffic on the commute to work, a much greater degree of sophistication is required to have any success at forecasting complicated events such as the weather or the performance of a stock market. Other complicated applications for forecasting include inventory control, production planning, marketing, as well as economic and financial planning.

Forecasting usually includes the construction of a forecasting model that predicts or estimates the value of a desired variable or a set of variables. The variables are computed from known data collected from past events, which are used to train the forecasting model. One form of known data is time series data, which comprises a set of data points recorded sequentially in time. In time series, each adjacent data point is typically dependent on adjacent data points.

Conventional forecasting models include linear regression models, Auto-Regressive Integrated Moving Average (ARIMA) models, Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). A linear regression model is a statistical model that assumes that a forecast can be decomposed into a weighted sum of a number of causal factors, plus a random error term. ARIMA models represent a general class of



linear models for forecasting a time series, which can be stationarized by transformations such as differencing and logging.

In contrast to linear regression and ARIMA models, ANN models are non-linear computational models that are designed to operate like a biological nervous system with the capability to learn. SVMs are also learning algorithms for pattern classification and regression. The basic principle behind SVMs is finding the optimal linear (or nonlinear) hyperplane (see above figure) such that the expected classification error for unseen test samples is minimized (i.e., good generalization performance).

Unfortunately, each of these models has limitations that are exposed by complicated forecasting applications. For example, because an ARIMA model and linear regression models are linear, these models are unable to capture non-linear relationships from training data (defined as known data used in training of forecasting models). Therefore, the use of ARIMA and linear regression models when training data is non-linear results in poor forecasting or generalisation performance.

A further problem with ARIMA models is that they assume the characteristics of the training data. This is undesirable as it introduces a bias into the forecasting model. For example, if a particular characteristic of the data to be forecasted is absent from the training data used in the construction of an ARIMA model, the ARIMA model will not be able to provide an accurate forecast since that particular characteristic will not be reflected in the forecast.

ANNs are able to make accurate forecasts for non-linear input, but must be trained by a complicated process. Specifically, an ANN is trained by presenting a set of training data to the network and thereafter, adjusting the weights in the network accordingly so that the network returns the desired estimate. However, before training of the ANN can commence, a large number of free parameters such as the number hidden layers and nodes in an ANN, the learning rate of the ANN, the initial weights, and the training algorithm to employ must be estimated.

Furthermore, because back-propagation of error using the gradient descent method is commonly employed to adjust the weights, the risk of error (such as locating a local minima instead of the global minima in a solution space) in an ANN is increased. The training of ANNs using the gradient descent method requires a substantial amount of computational time.

It is also often difficult to arrive at a network with an optimal degree of training. An ANN that is insufficiently trained will fail to fully capture the relationship in a complicated data set, resulting in underfitting. Conversely, an ANN that is over-trained can end up capturing not just the relationship between data, but also noise as well. This is referred to as overfitting. Both overfitting and underfitting may result in poor generalisation performance. Further, an ANN can be overfitted to respond to only one type of data. A separate ANN is then required for other types of data.

SVMs (such as the one discussed in US Patent No. 5,950,146 issued to Vapnik) are data-driven models having a solution that depends on subsets of the training data, also known as support vectors. Compared to the other conventional forecasting methods, SVM exhibits greater resilience against noise and non-stationary problems, which are the two main causes of forecasting error. However, the dependence of SVMs on a single set of training data is undesirable because the SVM is unable to account for the possibility that the set of training data may be flawed.

In view of the foregoing, it is desirable to have a method and system to construct a forecasting model that gives good generalisation performance that is not dependent on a single set of training data. It is also desirable to have a method and system to construct a forecasting model in which the training data is augmented and which requires the estimation of only a few free parameters. It is further desired that the forecasting model be applicable to both linear and non-linear data.

Summary of the Invention

The present invention fills these needs by providing a method and system to construct a forecasting model. It should be appreciated that the present invention can be implemented in numerous ways, including as a process, an apparatus, a system or a device. Several inventive embodiments of the present invention are described below.

In one embodiment of the present invention, a method to construct a forecasting model is provided. The method begins by sampling a primary training data set to generate a plurality of sampled training data sets. Sampling is preferably conducted by bootstrapping. Thereafter, a plurality of estimators is trained with the sampled training data sets. Each sampled training data set is used to train a single estimator. The estimators are preferably SVMs. The forecasting model is then generated by combining the estimators.

In a preferred embodiment, a set of optimal free parameters is selected for each of the estimators. The selection of a set of optimal free parameters begins by selecting a set of free parameters for one of the estimators. The estimator is then trained with a sampled training data set. After which, a set of cross-validation data is applied to the estimator to generate a set of predicted values. A prediction error is then calculated with the set of predicted values. The set of free parameters is optimal if the prediction error is minimal. The set of free parameters is preferably selected again if the prediction error is not minimal.

It should be appreciated that the primary training data set may be constructed from a time series. The number of sampled training data points in each sampled training data set may be increased to obtain a better generalisation performance.

In another preferred embodiment, the combination of the estimators begins by constructing a predicted training data set from the primary training data set using the estimators. After which, the forecasting model is constructed from the predicted training data set. The forecasting model is preferably constructed from a shrinkage estimator. Preferably, a set of optimal free parameters is selected for the shrinkage estimator.

In yet another embodiment of the present invention, a system to construct a forecasting model is provided. The system comprises a sampler to generate a plurality of sampled training data sets from a primary training data set. A trainer coupled to the sampler trains a plurality of estimators with the sampled training data sets generated.

- 5 Each sampled training data set is used to train a single estimator. A combiner coupled to the trainer combines the estimators to generate the forecasting model.

Other aspects and advantages of the invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating by way of example the principles of the invention.

Brief Description of the Drawings

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings. To facilitate this description, like reference numerals designate like structural elements.

5 Figure 1 illustrates a method to construct a forecasting model in accordance with one embodiment of the present invention.

Figure 2 illustrates a sampling process to generate a plurality of sampled training data sets in accordance with one embodiment of the present invention.

10 Figure 3 illustrates a method to select optimal free parameters for an estimator in accordance with one embodiment of the present invention.

Figure 4 illustrates a method to combine a plurality of estimators in accordance with one embodiment of the present invention.

15 Figure 5 illustrates a bar chart comparing the performance of thirty individual SVM regressors and a forecasting model in accordance with one embodiment of the present invention.

Figure 6 illustrates a bar chart comparing the performance of a conventional SVM regressor and a forecasting model in accordance with one embodiment of the present invention.

20 Figure 7 illustrates a system to construct a forecasting model in accordance with one embodiment of the present invention.

Detailed Description of the Preferred Embodiments

A method and system to construct a forecasting model are provided. In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be understood, however, to one skilled in the art, that the present invention may be practiced without some or all of these specific details. In other instances, well known process operations have not been described in detail in order not to unnecessarily obscure the present invention.

Figure 1 illustrates a method 100 to construct a forecasting model in accordance with one embodiment of the present invention. A primary training data set is constructed in a block 150 from a set of raw data, such as the daily market closing price of a particular stock over a period of a year. This particular raw data set would be represented by a time series $\{y_i\}, i = 1, \dots, l$, however time independent data, such as the system performance of a computer, may also be used.

The primary training data set is subsequently sampled in a block 200, preferably by bootstrapping or Arc-x4, to generate a plurality of sampled training data sets. Then, in a block 250 the plurality of sampled training data sets is used to train an estimator or forecasting model. Examples of estimators include linear regression models, ARIMAs, ANNs, and SVMs. Each sampled training data set is used to train one estimator. Finally, a forecasting model is generated by combining the plurality of estimators in a block 300.

The primary training data set can be constructed in block 150 with the following training data patterns:

$$\{x_i, y_i\}, i = 1, \dots, l$$

where y_i is time-independent and l represents the total number of training data in the set; and

$$\{x_i = (y_{i-d}, \dots, y_{i-1}, e_i), y_i\}, i = d+1, \dots, t$$

where y_i is a time series.

Each training data pattern comprises a vector of an input variable x_i and an output variable y_i . In cases where y_i is a time series, the vector of the input variable x_i comprises d lagged values of the time series ($y_{i-d}, \dots, y_{i-1}$) and an exogenous variable e_i , which is used to compensate for outside factors. The exogenous variable e_i may be omitted in instances where there are no related or proxy data to be taken into consideration. The next value of the time series y_i is the output variable.

Figure 2 illustrates sampling block 200 in greater detail in accordance with one embodiment of the present invention. In general, sampling involves selecting a number of training data points from the primary training data set to form a sampled training data set. Examples of sampling techniques, which can be applied in sampling block 200, include Arc-x4 and bootstrapping. In the bootstrapping process, a primary training data set 202 is repeatedly random sampled to generate a plurality of sampled training data sets 204, 206, 208, 210, and 212. Primary training data set 202 is made up of the following training data: $t_1, t_2, t_3, t_4, t_5, t_6$.

Each sampled training data set 204, 206, 208, 210, and 212 includes a random combination of training data from primary training data set 202. Each sampled training data set 204, 206, 208, 210, and 212 may contain a number of training data points equal to, greater than, or less than the six training data points in primary training data set 202. The number of training data points in each sampled training data set is chosen to balance generalization performance with computational time. A greater number of training data points selected will result in a better generalisation performance, but increases computational time. When a smaller number of training data is selected, computational time is reduced at the expense of the generalisation performance.

Because each of the sampled training data sets are obtained by sampling the same primary training data set, the statistical distribution of the sampled training data sets will mimic that of the primary training data set. Therefore, because the sampled training data sets share the same statistical characteristics as the primary training data set, the sampled training data sets can be used to train SVMs to estimate the primary training data set.

Referring back to Figure 1, each sampled training data set 204, 206, 208, 210, or 212 generated by sampling in block 200, is used to train an estimator such as a linear regression model, an ARIMA model, an ANN, or a SVM in block 250. For example, training the plurality of estimators using SVM regression is effected by mapping data from an input space onto a feature space of a higher dimension. A linear estimator is then constructed in the feature space. The linear estimator in the feature space corresponds to a non-linear estimator in the input space.

Given a sampled training data set $(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i)$, where $x_i \in X \subseteq R^n$ represents a vector of an input variable and $y_i \in Y \subseteq R$ represents an output variable, an estimator derived using SVM regression approximates the following linear estimator:

$$f(x_i) = w \cdot \phi(x_i) + b \quad (1)$$

where $\phi(x_i)$ represents a feature space of a higher dimension on which training data x_i from an input space is mapped. w and b are coefficients, which are estimated by minimising the following regularised risk function:

$$\text{Minimise } \frac{1}{2} \|w\|^2 + C \frac{1}{l} \sum_{i=1}^l L_\varepsilon(y_i, f(x_i)) \quad (2)$$

where $\|w\|^2$ is known as the regularised term. Because it is important to restrain the ability of the function to estimate the training set (also referred to as function capacity), $\|w\|^2$ is minimised. In this manner, the estimator attains maximum possible flatness to control function capacity and minimize overfitting.

The second term $\frac{1}{l} \sum_{i=1}^l L_\varepsilon(y_i, f(x_i))$ represents an empirical error measured by the following ε -insensitive loss function:

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} |y_i - f(x_i)| - \varepsilon & |y_i - f(x_i)| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

By applying equation (3), the linear estimator in equation (1) may be represented even with only a sparse number of data points. ε is commonly referred to as the tube size and C is known as the regularisation constant. Both ε and C are user-defined free parameters and are determined empirically.

5 To obtain estimates of w and b , equation (2) is transformed into the following primal objective function through the introduction of positive slack variables ξ_i, ξ_i^* :

$$\text{Minimise } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*)$$

(4)

subject to the following optimisation constraints:

$$\begin{aligned} y_i - w \cdot \phi(x_i) - b &\leq \varepsilon + \xi_i \\ w \cdot \phi(x_i) + b - y_i &\leq \varepsilon + \xi_i^*, \quad i = 1, \dots, l \\ \xi_i^{(*)} &\geq 0 \end{aligned}$$

A Lagrange function is then constructed from both equation (4) and the corresponding optimisation constraints through the introduction of a dual set of variables. Because the optimal solution occurs where the Lagrange function has a saddle point with respect to the primal and dual variables, the partial derivatives of the Lagrange function with respect to the primal variables w , b , ξ_i and ξ_i^* equate to zero for optimality.

Exploiting the Lagrange function with the saddle point constraints yields a dual optimization problem. In the process of deriving the dual optimization problem, a set of the dual variables are eliminated, leaving only one set, that is, a_i, a_i^* , otherwise known as the Lagrange multipliers. The Lagrange multipliers a_i, a_i^* satisfy the following conditions: $a_i a_i^* = 0$, $a_i \geq 0$ and $a_i^* \geq 0$, where $i = 1, \dots, l$.

By maximising the dual optimization problem, one arrives at:

$$W(a_i^{(*)}) = \sum_{i=1}^l y_i (a_i - a_i^*) - \varepsilon \sum_{i=1}^l (a_i + a_i^*) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (a_i - a_i^*) (a_j - a_j^*) K(x_i, x_j) \quad (5)$$

subject to the following constraints:

$$\sum_{i=1}^l (a_i - a_i^*) = 0$$

$$0 \leq a_i^{(*)} \leq C, \quad i = 1, \dots, l$$

In equation (5), $K(x_i, x_j)$ is a kernel function that satisfies Mercer's condition with a value defined by $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. The use of kernel functions allows the present invention to take into account feature spaces of arbitrary dimensions without the need to compute a map $\phi(x)$. Kernel functions, which satisfy this condition, include the polynomial kernel, the Gaussian kernel, and the two-layer tangent kernel.

Upon further exploitation, the following explicit form of the linear estimator in equation (1) is arrived at:

$$f(x_j) = \sum_{i=1}^l (a_i - a_i^*) K(x_i, x_j) + b \quad (6)$$

The training of an estimator using SVM regression can be likened to the solving of a linearly constrained quadratic programming problem, represented by equation (5), having twice the number of variables as the number of training data points. Because the generalization performance of an estimator is affected by the empirically chosen free parameters, such as ε and C , the selected kernel function, and the free parameters, which define the kernel function in the preceding example, the estimator will give a better generalization performance if optimal free parameters are selected.

A method to select the optimal free parameters for an estimator is illustrated in Figure 3. A set of free parameters for the estimator is empirically selected in a block 352. The estimator is then trained in a block 354 with a sampled training data set. A set

of cross-validation data, which is a data set that is not used for training, is applied in a block 356 to the trained estimator to generate a set of predicted values. A prediction error is then calculated in a decision block 358 with the set of predicted values. If the prediction error calculated is minimal, the set of free parameters selected in block 352 is accepted as the optimal free parameters for the estimator. On the other hand, if the prediction error calculated is not minimal, the set of free parameters selected in block 352 is chosen again.

Prediction error can be quantified using the equation for normalised mean squared error (NMSE) given below:

$$NMSE = \frac{1}{\sigma^2 l_v} \sum_{i=1}^{l_v} (y_i - f(x_i))^2 \quad (7)$$

where l_v represents the total number of data in the set of cross-validation data and σ is a measure of the standard deviation of the actual values in the set of cross-validation data. Other criteria to measure or quantify the prediction error include mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE) and normalised root mean squared error (NRMSE).

As discussed previously with reference to Figure 1, each sampled training data set generated by sampling in block 200 is used to train a single estimator in block 250. Because each estimator is trained with a different sampled training data set having a different statistical distribution, each estimator is likely to generate a different prediction error. To minimise the prediction error, the plurality of estimators trained in block 250 is combined in block 300 by a shrinkage estimator, which outputs a final prediction. Examples of shrinkage estimators include principal component regression (PCR), partial least squares (PLS), ridge regression (RR), kernel principal component regression (KPCR), kernel partial least squares (KPLS), and kernel ridge regression (KRR).

Shrinkage estimators are required because the sampled training data sets used to train the plurality of estimators in block 250 originated from the same primary training

data set, causing a correlation between each estimator. If not dealt with properly by the shrinkage estimator, this correlation between each of the plurality of estimators can result in multi-collinearity, which has the undesired effect of deteriorating generalisation performance when the plurality of estimators is combined in block 300.

5 By way of example, a method to combine in block 300 the plurality of estimators using KPCR, a non-linear PCR method developed by generalising the kernel method into a linear PCR method, is illustrated in detail with reference to Figure 4. A predicted training data set $(f(x_1), y_1), (f(x_2), y_2), \dots, (f(x_l), y_l))$ is first constructed in a block 302. The predicted training data set comprises $f(x_i) \subseteq R^n$, which is a vector of predicted values
 10 estimated by applying the primary training data set to the plurality of estimators trained in block 250, as the input vector and $y_i \subseteq R$ as the output variable.

The vector of predicted values $f(x_i)$ is then mapped in a block 304 onto a feature space $\phi(f(x_i))$ of a dimension greater than the number of training data l . In doing so, the following eigenvalue problem is solved:

15
$$\lambda_i u_i = \tilde{C} u_i, i = 1, \dots, l$$

 (8)

where $\tilde{C} = \frac{1}{l} \sum_{i=1}^l \phi(f(x_i)) \phi(f(x_i))^T$ represents a sample covariance matrix of $\phi(f(x_i))$, λ_i represents a non-zero eigenvalue of $\tilde{C}$ and u_i represents a corresponding eigenvector.

20 Equation (8) is then transformed into a Kernel equation and expressed as follows:

$$\tilde{\lambda}_i \alpha_i = K \alpha_i, i = 1, \dots, l$$

(9)

where $K_{ij} = \phi(f(x_i))^T \cdot \phi(f(x_j))$, representing a $l \times l$ kernel matrix. $\tilde{\lambda}_i$ represents an eigenvalue of K satisfying the condition $\tilde{\lambda}_i = l\lambda_i$. α_i represents a corresponding eigenvector of K satisfying the condition:

$$u_i = \sum_{j=1}^l \alpha_i(j) \phi(f(x_j))$$

5 where $\alpha_i(j), j = 1, \dots, l$ are components of α_i . To ensure that the eigenvectors of $\phi(f(x_j))$ are of unit length, that is, $u_i \cdot u_i = 1$, each α_i must be normalised with a corresponding eigenvalue by applying the equation $\tilde{\alpha}_i = \alpha_i / \sqrt{\tilde{\lambda}_i}$.

Estimated values of $\tilde{\alpha}_i$ are then used to calculate the principal components s of $f(x_i)$ with the following equation in a block 306:

$$10 \quad s_j(i) = u_i^T \phi(f(x_j)) = \sum_{k=1}^l \tilde{\alpha}_i(k) K(f(x_k), f(x_j)), \quad i = 1, \dots, l$$

(10)

In order to center $f(x_i)$ in $\phi(f(x_i))$, that is, $\sum_{i=1}^l \phi(f(x_i)) = 0$, in equation (10), the kernel matrix of the training data set K and that the testing data set K_t are modified as follows:

$$15 \quad \tilde{K} = (I - \frac{1}{l} \mathbf{1}_l \mathbf{1}_l^T) K (I - \frac{1}{l} \mathbf{1}_l \mathbf{1}_l^T)$$

(11)

$$\tilde{K}_t = (K_t - \frac{1}{l} \mathbf{1}_t \mathbf{1}_l^T K) (I - \frac{1}{l} \mathbf{1}_t \mathbf{1}_t^T)$$

(12)

where I represents an l dimensional identity matrix and l_t represents the number of data points for testing. 1_l and 1_{l_t} represent vectors whose elements are all 1 and whose lengths are l and l_t respectively. K_t represents a $l_t \times l$ kernel matrix of the data points for testing.

- 5 The mapping of $f(x_i)$ in block 304 onto $\phi(f(x_i))$ and the calculation in block 306 of the principal components s of $f(x_i)$ are in essence kernel principal component analysis (KPCA).

10 With the calculated principal components s , a forecasting model can then be constructed in a block 308. By projecting $\phi(f(x_i))$ onto s , the standard regression model in feature space can be expressed as follows:

$$y = sb + \xi \quad (13)$$

15 where y represents a vector of l observations of a dependent variable, s represents a $l \times m$ matrix of principal components whose row comprises the first m principal components $s(i), i = 1, \dots, m$, b represents a vector of regression coefficients and ξ represents a vector of error terms.

A least square estimate of the coefficient b can be calculated using:

$$b = \Lambda^{-1} S^T y \quad (14)$$

where $\Lambda = \text{diag}(\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_m)$.

- 20 Using the first m non-linear principal components, sorted in descending order of their eigenvalues, the forecasting model can be constructed as follows:

$$f_{model}(f(x_j), b) = \sum_{i=1}^l c_i K(f(x_i), f(x_j)) \quad (15)$$

where $\{c_i = \sum_{k=1}^m b_k \alpha_i^k\}_{i=1}^l$.

To improve generalisation performance, the set of cross-validation data and method 350 to select the optimal free parameters for an estimator, illustrated in Figure 3, may be employed to choose the optimal free parameters such as the kernel function, the kernel parameters and the optimal number of principal components m , for KPCR. The free parameters that produce the smallest error are used to construct the forecasting model.

A study involving a well-known sunspot time series was conducted to evaluate the accuracy of forecasting models obtained in accordance with one embodiment of the present invention. Sunspots are dark spots that appear from time to time on the sun's surface and are usually visible only through a telescope. The set of data used included two hundred and eighty yearly averaged sunspots recorded between 1700 and 1979. Data obtained in the period between 1700 and 1955 were used to form a primary training data set, while the remaining data obtained in the period between 1956 and 1979 were used to form a set of test data.

Thirty-five randomly selected data from the primary training data set were used to form a set of validation data for the selection of optimal free parameters. However, this set of validation data was not used to train the estimators. The primary training data set comprised twelve previously recorded sunspots. Thirty sets of sampled training data generated by bootstrapping were then used to train a corresponding set of thirty SVM regressors. The SVM regressors were optimised by selecting free parameters such as kernel parameters, C and ε , which produced the smallest NMSE. The value of these free parameters varies depending on the sampled training data set employed.

Three types of kernel functions were investigated: the polynomial kernel, the Gaussian kernel and the two-layer tangent kernel. Values of C ranging from 0.1 to 1000 and values of ε ranging from 0.0001 to 0.1 were studied in increments of multiples of 10. The thirty SVM regressors were then combined via KPCR to construct a forecasting model. In the process of combining the thirty SVM regressors, a vector of predicted

values was estimated from the thirty SVM regressors using the primary training data set. A predicted training data set was then constructed from the vector of predicted values and data from the primary training data set. Optimal free parameters were selected for KPCR.

5 The results of this study are illustrated in Figure 5. Figure 5 is a bar chart comparing the performance of thirty individual SVM regressors with that of the forecasting model. The forecasting error NMSE of each SVM regressor was calculated and charted. As evident from Figure 5, the forecasting model produces the smallest NMSE as compared to the other individual SVM regressors. This demonstrates that the
10 forecasting model achieves better generalisation performance than the thirty individual SVM regressors and that the problem posed by multi-collinearity can be overcome.

 A further comparison was carried out between the performance of a conventional SVM regressor and the forecasting model. The optimal free parameters of the conventional SVM regressor were determined with the same set of validation data as in
15 the previous study. Likewise, the polynomial kernel, the Gaussian kernel and the two-layer tangent kernel were studied. Values of c ranging from 0.1 to 1000 and values of ε ranging from 0.0001 to 0.1 were studied in increments of multiples of 10. The results of this study are illustrated in Figure 6.

 Figure 6 illustrates a bar chart comparing the performance of a conventional
20 SVM regressor with that of the forecasting model. Once again the forecasting model produced a significantly smaller NMSE than the conventional SVM regressor, demonstrating that the forecasting model achieves a better generalisation performance than the conventional SVM regressor.

 A system 400 to construct a forecasting model is illustrated in Figure 7. System
25 400 comprises a trainer 402 coupled to a sampler 404 and a combiner 406. A primary training data set is first input to sampler 404, which samples the primary training data set to generate a plurality of sampled training data sets. Thereafter, trainer 402 trains a plurality of estimators with the plurality of sampled training data sets generated by sampler 404. Each sampled training data set preferably trains a single estimator. The

plurality of estimators is then combined in combiner 406 to generate the forecasting model.

As discussed previously, the primary training data set is constructed from a set of raw data. This set of raw data can be a time series or time-independent. Further, each sampled training data set can contain greater, fewer or equal numbers of training data as compared to the primary training data set. As described above, sampling techniques such as bootstrapping and Arc-x4 can be employed in sampler 404 to generate the plurality of sampled training data sets. Similarly, estimators such as ARIMA models, linear regression models, ANNs and SVMs are examples of estimators that can be trained in trainer 402.

Because the generalization performance of an estimator is affected by empirically chosen free parameters such as C , ε , the kernel function selected and the free parameters, which define the kernel function, optimal free parameters for the estimators are selected in a selector 408 incorporated in trainer 402.

Combiner 406 comprises a first constructor 410 and a second constructor 412 coupled to first constructor 410. First constructor 410 constructs a predicted training data set from the primary training data set using the plurality of estimators. This predicted training data set is then fed into second constructor 412 for the construction of the forecasting model. The forecasting model can be constructed from an estimator such as, for example, ARIMA models, SVMs and linear regression models. To overcome the problem of multi-collinearity, shrinkage estimators such as PCR, PLS, RR, KPCR, KPLS and KRR techniques may be employed instead.

As the generalization performance of the forecasting model is also affected by empirically chosen free parameters such as the kernel function, the kernel parameters and the optimal number of principal components m , optimal free parameters for the forecasting model are selected in a selector 414 incorporated in second constructor 412. System 400 can be implemented on a single device such as an integrated circuit chip or a handheld device.

An advantage of the present invention is that the forecasting model constructed with the present forecasting method or system gives a better generalization performance than the conventional forecasting models. Selection of optimal free parameters for the estimators by determining the set of free parameters that give a minimal prediction error also improves the generalization performance of the constructed forecasting model.

Additionally, the present invention overcomes the dependency on a single set of training data by sampling the primary training data set to generate a plurality of sampled training data sets and combining the estimators trained with the sampled training data sets. The present invention also avoids the problem of multi-collinearity through the use of a shrinkage estimator to combine the estimators. Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention. Furthermore, certain terminology has been used for the purposes of descriptive clarity, and not to limit the present invention. The embodiments and preferred features described above should be considered exemplary, with the invention being defined by the appended claims.

CLAIMS

1. A method to construct a forecasting model, said method comprising:

sampling a primary training data set to generate a plurality of sampled training
5 data sets;

training a plurality of estimators with said plurality of sampled training data sets,
wherein each of said plurality of sampled training data sets is used to train one of said
plurality of estimators; and

combining said plurality of estimators to generate said forecasting model.

10

2. A method to construct a forecasting model as recited in claim 1, wherein
sampling is conducted by bootstrapping.

3. A method to construct a forecasting model as recited in claim 2, wherein
15 each of said plurality of estimators is a Support Vector Machine.

4. A method to construct a forecasting model as recited in claim 3, further
comprising:

selecting a set of free parameters for each of said plurality of estimators;

20 training each of said plurality of estimators;

applying a set of cross-validation data to each of said plurality of estimators to
generate a set of predicted values; and

calculating a prediction error with said set of predicted values, wherein said set of
free parameters is accepted as optimal if said prediction error is minimal.

25

5. A method to construct a forecasting model as recited in claim 4, wherein said set of free parameters is selected again if said prediction error is not minimal.

6. A method to construct a forecasting model as recited in claim 5, wherein said primary training data set is constructed from a time series.

5

7. A method to construct a forecasting model as recited in claim 6, wherein the number of sampled training data points in each of said plurality of sampled training data sets is increased to obtain a better generalisation performance.

10 8. A method to construct a forecasting model as recited in claim 1, further comprising:

selecting a set of free parameters for each of said plurality of estimators;

training each of said plurality of estimators;

15 applying a set of cross-validation data to each of said plurality of estimators to generate a set of predicted values; and

calculating a prediction error with said set of predicted values, wherein said set of free parameters is accepted as optimal if said prediction error is minimal.

20 9. A method to construct a forecasting model as recited in claim 8, wherein said set of free parameters is selected again if said prediction error is not minimal.

10. A method to construct a forecasting model as recited in claim 1, wherein said step of combining comprises:

25 constructing a predicted training data set from said primary training data set using said plurality of estimators; and

constructing said forecasting model from said predicted training data set.

11. A method to construct a forecasting model as recited in claim 10, wherein said forecasting model is constructed from a shrinkage estimator.

12. A method to construct a forecasting model as recited in claim 11, further comprising:

selecting a set of free parameters for said shrinkage estimator;

training said shrinkage estimator;

applying a set of cross-validation data to said shrinkage estimator to generate a set of predicted values; and

calculating a prediction error with said set of predicted values, wherein said set of free parameters is accepted as optimal if said prediction error is minimal.

13. A method to construct a forecasting model as recited in claim 12, wherein said set of free parameters is selected again if said prediction error is not minimal.

14. A forecast generated from said forecasting model constructed with the method recited in claim 1.

15. A system to construct a forecasting model, said system comprising:

a sampler to generate a plurality of sampled training data sets from a primary training data set;

a trainer coupled to said sampler, wherein said trainer is to train a plurality of estimators with said plurality of sampled training data sets, wherein each of said plurality of sampled training data sets is used to train one of said plurality of estimators; and

a combiner coupled to said trainer, wherein said combiner is to combine said plurality of estimators to generate said forecasting model.

16. A system to construct a forecasting model as recited in claim 15, wherein each of said plurality of estimators is a Support Vector Machine.

17. A system to construct a forecasting model as recited in claim 16, wherein said trainer further comprises a selector to select a set of optimal free parameters for each of said plurality of estimators.

18. A system to construct a forecasting model as recited in claim 17, wherein said plurality of sampled training data sets is generated by bootstrapping.

19. A system to construct a forecasting model as recited in claim 18, wherein said primary training data set is constructed from a time series.

20. A system to construct a forecasting model as recited in claim 19, wherein the number of sampled training data points in each of said plurality of sampled training data sets is increased to obtain a better generalisation performance.

21. A system to construct a forecasting model as recited in claim 15, wherein said combiner comprises:

a first constructor to construct a predicted training data set from said primary training data set using said plurality of estimators; and

a second constructor coupled to said first constructor, wherein said second constructor is to construct said forecasting model from said predicted training data set.

22. A system to construct a forecasting model as recited in claim 21, wherein said forecasting model is constructed from a shrinkage estimator.

23. A system to construct a forecasting model as recited in claim 22, wherein said second constructor includes a selector to select a set of optimal free parameters for said shrinkage estimator.

5 24. A system to construct a forecasting model as recited in claim 15, wherein said system is implemented on an integrated circuit chip.

25. A system to construct a forecasting model as recited in claim 15, wherein said system is implemented on a handheld device.

10

26. A forecast generated from said forecasting model constructed with the system recited in claim 15.