Fast Bootstrap applied to LS-SVM for Long Term Prediction of Time Series

Amaury Lendasse
HUT, CIS, FI-02015, Finland
E-mail: lendasse@cis.hut.fi

Vincent Wertz
CESAME, UCL, LLN 1348, Belgium
Email : wertz@auto.ucl.ac.be

Geoffroy Simon, Michel Verleysen
DICE, UCL, LLN 1348, Belgium
Emails:
{simon,Verleysen}@dice.ucl.ac.be

Abstract - **Time series forecasting is usually limited to one-step ahead prediction. This goal is extended here to longer-term prediction, obtained using the least-square support vector machines model. The influence of the model parameters is observed when the time horizon of the prediction is increased and for various prediction methods. The model selection to optimize the design parameters is performed using the Fast Bootstrap methodology introduced in previous works.**

I. Introduction

Time series forecasting is a general problem encountered in many field as engineering (electrical consumption, gas consumption, ...), finance (share or stock evolution, ...), environment (river flood, ...) to give only a few examples. The general problem of time series forecasting can be rephrased as the problem of finding a model able to forecast the future evolution of a time series given its past evolution. Most of the time, the forecasting problem is limited to a short-term time series prediction. In other words, as one tries to model the future evolution of a time series, the usual goal is to be able to perform a one-step ahead prediction. The main reason motivating such approach is reliability of the predicted values. One-step ahead predictions can be reasonably reliable, while the uncertainty on future values increases with the time horizon. The idea is thus to see how models that have been used to perform one-step ahead predictions behave in the more general framework of multiple steps ahead predictions (where multiple steps can mean a relatively large number of future values). Furthermore, as these models are parameterised, the relative importance of their parameters will be observed on longer-term prediction. The influence of these parameters will thus be underlined as the time horizon of the prediction increases. Since one chooses a family of parameterised models, there exists as many models as there are different values for the parameters. The problem is thus to be able to choose the best one among a family of models, according to some criterion (usually the generalisation error).

Many techniques have been developed in the general framework of model selection. Some of them are based on a penalisation of the model complexity, as AIC, BIC, MDL [1, 2, 3], while others are based on resampling, as k-fold cross-validation, leave-one-out, and bootstrap [4]. Although they differ in their approach, these methods, either based on complexity penalty or resampling, have been proved to be asymptotically equivalent [5]. Within the resampling methods, the bootstrap will be used here, as it provides a more robust estimate of the generalisation error [6]. Nevertheless, the bootstrap has an awkward limitation. The computation time needed to obtain an estimate of the generalisation error can be very large when using nonlinear models.

An improvement to the bootstrap, namely the Fast Bootstrap, will be extended here to the case of least-squares support vector machines (LS-SVM) [7, 8]. This improvement has already been applied to radial basis function networks [9, 10]. In the following of this paper, we first recall some basic concepts about LS-SVM. The principle of the bootstrap will also be recalled in section III. The Fast Bootstrap improvement will be introduced for LS-SVM in section IV. Section V deals with various approaches for long-term forecasting. The LS-SVM will then be applied to the SantaFe A time series in order to observe the influence of the model in the case of long-term forecasting. This influence will be finally discussed in the conclusion.

II. Least-Square Support Vector Machines

Consider a given training set of N data points $\{x_k, y_k\}$ with x_k a n-dimensional input and y_t a 1-dimensional output. In feature space SVM models take the form:

$$y(x) = \omega^T \varphi(x) + b \,,$$

where the nonlinear mapping $\varphi(.)$ maps the input data into a higher dimensional feature space. In least squares support vector machines for function estimation, the following optimization problem is formulated:

$$\min_{\omega,e} J(\omega,e) = \frac{1}{2}\omega^T\omega + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 \,,$$

subject to the equality constraints:

$$y(x) = \omega^T \varphi(x) + b + e_k, \; k = 1,...,N.$$

This corresponds to a form of ridge regression. The Lagrangian is given by:

0-7803-8359-1/04/$20.00 ©2004 IEEE

$$L(\omega,b,e,\alpha) = J(\omega,e) - \sum_{k=1}^{N}\alpha_k\left\{\omega^T\varphi(x_k) + b + e_k - y_k\right\}$$

with Lagrange multipliers α_k. The conditions for optimality are:

$$\begin{cases} \dfrac{\partial L}{\partial \omega} L = 0 \rightarrow \omega = \sum_{k=1}^{N}\alpha_k\varphi(x_k) \\ \dfrac{\partial L}{\partial e_k} L = 0 \rightarrow \alpha_k = \gamma e_k \\ \dfrac{\partial L}{\partial b} L = 0 \rightarrow \sum_{k=1}^{N}\alpha_k = 0 \\ \dfrac{\partial L}{\partial \alpha_k} L = 0 \rightarrow \omega^T\varphi(x_k) + b + e_k - y_k = 0 \end{cases},$$

for $k = 1..N$. After elimination of e_k and ω, the solution is given by the following set of linear equations:

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \Omega + \gamma^{-1}I \end{bmatrix}\begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix},$$

where $y = [y_1; \ldots; y_N]$, $\vec{1} = [1; \ldots; 1]$, $\alpha = [\alpha_1; \ldots; \alpha_N]$ and Ω follows Mercer's condition:

$$\Omega_{kl} = \frac{\varphi(x_k)^T\varphi(x_l)}{\psi(x_k, x_l)} \quad k,l = 1,\ldots,N.$$

This finally results into the following LS-SVM model for function estimation:

$$y(x) = \omega^T\varphi(x) + b,$$

where α and b are the solution to (6) and ω is given by (5). For the choice of the kernel function $\psi(.,.)$ one has several possibilities [7,8]. In this paper, Gaussian kernels are used: $\psi(x, x_k) = \exp\{-\|x-x_k\|^2/\sigma^2\}$. The remaining unknowns are σ and γ. These model hyperparameters will be selected according to a model selection procedure detailed in the following of this paper.

III. Bootstrap for Model Structure Selection

The bootstrap [4] is a resampling method that has been developed in order to estimate some statistical parameters (like the mean, the variance, etc). In the case of model structure selection, the parameter to be estimated is the generalization error (i.e. the average error that the model would make on an infinite-size and unknown test set). When using the bootstrap, this error is not computed directly. Rather the boorstrap estimates the difference between the generalization error and the training error calculated on the initial data set. This difference is called the *optimism*. The estimated generalization error will thus be the sum of the training error and of the estimated optimism. The training error is computed using all data from the training set. The optimism is estimated using a resampling technique based on drawing within the training set with replacement. Using notation $E_j^{A_j,A_j}$ where the first exponent A_j denotes the training set while the second exponent A_j indicates the set used to estimate the model error, the Bootstrap method can be decomposed in the following stages:

1. From the initial set I, one randomly draws N points with replacement. The new set A_j has thus the same size that the initial set and constitutes a new training set. This stage is called the resampling.
2. The training of the various model structures q is done on the same training set A_j. One can compute the training error on this single set:

$$E_j^{A_j,A_j}(q,\theta_j^*(q)) = \frac{\sum_{i=1}^{N}\left(h^q(x_i^{A_j},\theta_j^*(q)) - y_i^{A_j}\right)^2}{N}, \quad (9)$$

with θ_j^* the model parameters after learning, h^q the q^{th} model that is used, x_i^{Aj} the i^{th} input vector from set A_j, y_i^{Aj} the i^{th} output and N the number of elements in this set. Index j means that the error is evaluated on the j^{th} new sample.
3. One can also compute the validation error on the initial sample which now plays the role of the validation set V=I:

$$E_j^{A_j,V}(q,\theta_j^*(q)) = \frac{\sum_{i=1}^{N}\left(h^q(x_i^V,\theta_j^*(q)) - y_i^V\right)^2}{N}. \quad (10)$$

Here again index j means that the error is evaluated on the j^{th} new sample.
4. The difference between these two errors (9) and (10) is calculated and defined as the *optimism* by Efron [6]:

$$optmism_j(q,\theta_j^*(q)) = E_j^{A_j,V}(q,\theta_j^*(q)) - E_j^{A_j,A_j}(q,\theta_j^*(q)) \quad (11)$$

5. Steps 1 to 4 are repeated J times. The estimate of the optimism is then calculated as the average of the J values from (11):

$$\hat{optmism}(q) = \frac{\sum_{j=1}^{J} optmism_j(q,\theta_j^*(q))}{J}. \quad (12)$$

6. The training of the q model structures is done on the initial data set I and the training error is calculated on the same set. Two exponents I are used to indicate that the initial data set is used for both training and error estimation:

$$E^{I,I}(q,\theta^*(q)) = \frac{\sum_{i=1}^{N}\left(h^q(x_i^I,\theta^*(q)) - y_i^I\right)^2}{N}. \quad (13)$$

7. An approximation of the generalization error is finally obtained by:

$$\hat{E}_{gen}(q) = \hat{optmism}(q) + E^{I,I}(q,\theta^*). \quad (14)$$

$\hat{E}_{gen}(q)$ is an approximation of the generalization error for each model structure q. The best structure that will be selected is the one that minimizes this estimate of the generalization error. In this paper, the Bootstrap .632 will be used instead of the classical Bootstrap described above. In Bootstrap .632, the validation set V is made with the elements that are in the initial data set I but not the training

set A_j. Since the set V_j is different for each set A_j, one have to replace V by Vj in relations (10) and (11) while the remaining equations remain unchanged. Finally, (14) is replaced by:

$$\hat{E}_{gen}(q) = .632\, optimism(q) + .368\, E^{I,J}(q,\theta^*). \quad (15)$$

The main advantage of this version is that the estimate of the generalization error obtained by the Bootstrap .632 is unbiased [4].

IV. Fast Bootstrap and Toy Example

In this section, an improvement to the Bootstrap methods is presented. This improvement is called the Fast Bootstrap and allows reducing the computational time of the traditional Bootstraps [9-10]. This method is based on experimental observations and is presented on a function approximation example. In this example, 200 inputs x has been drawn using a uniform random law between 0 and 1. The output y has been generated by the function:

$$y = \sin(5x) + \sin(15x) + \sin(25x) + \varepsilon, \quad (16)$$

with ε a uniformly distributed random value in [-0.5 0.5]. This function is represented in Figure 1.

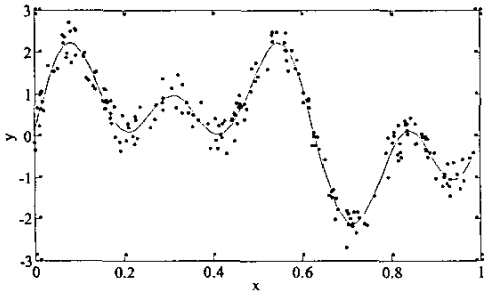

Figure 1: Example of function (dots) and its approximation (solid line).

A LS-SVM is used to approximate this function. Two parameters still have to be determined, namely σ and γ. For a fixed σ = 0.1, the optimal γ is determined using the Bootstrap method. The set of γ that is tested ranges from 0 to 100 with a 0.1 step. The number of resamplings in (12) is equal to 100. The apparent error defined in (13) is computed and represented in Fig.2. The optimism is computed using (12) and represented in Fig.3. The generalization error is computed using (15) and represented in Fig.4. The value of γ that minimizes the generalization error is equal to 11. In Fig.3, the optimism is very close from an exponential function of γ. This fact has been observed on other examples and benchmarks. Then, using this information, the number of values of γ to be tested can be considerably reduced. In this example, this set is indeed reduced to 5 to 100 with an incremental step of 5. An exponential approximation of the optimism is used. Thanks to the approximation, the number of Bootstraps is also reduced by a factor 10 in (12). The new optimism and generalization error are represented as dotted lines in Fig.3 and Fig.4 respectively. The optimum is close to the one that has been selected by the Bootstrap method. This new method, denoted Fast Bootstrap, is in this toy example 500 times quicker than the traditional Bootstrap. In other examples, the Fast Bootstrap is at least 100 times quicker than traditional Bootstrap for the selection of the γ parameter for a LS-SVM, without loss of precision.

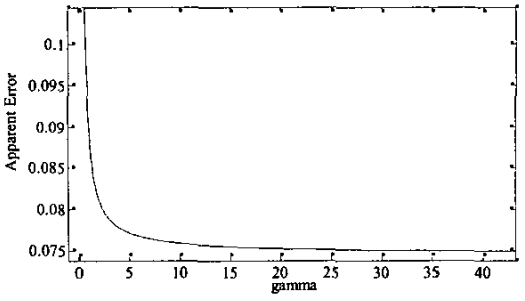

Figure 2: Apparent Error with respect to γ.

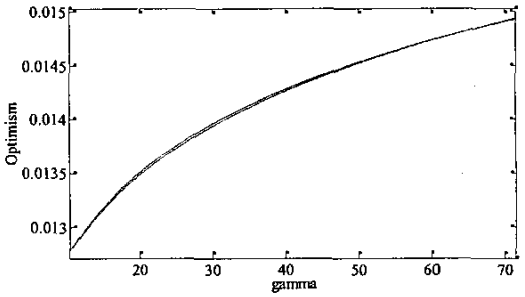

Figure 3: Optimism with respect to γ using Bootstrap (solid line) and Fast Bootstrap (dotted line).

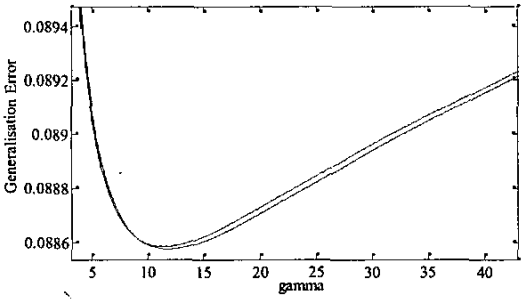

Figure 4: Generalization Error with respect to γ using Bootstrap (solid line) and Fast Bootstrap (dashed line).

V. Long-term forecasting strategies

A. *Definition of the problem*

Long-term forecasting is just an extension of the usual one-step ahead prediction that could be called short-term forecasting. More formally, having at disposal a time series of inputs x_t and exogenous variables u_t, with t between 1 and n, the one-step ahead prediction problem is usually defined as:

$$\hat{x}_{t+1} = h(x_t, x_{t-1}, ..., x_{t-p}, u_t, u_{t-1}, ..., u_{t-q}, \theta) \quad (17)$$

where $h(.)$ is the model used to predict the time series, x_t are the model inputs, u_t are some exogenous variables, θ the set of model parameters and $\hat{x}_{t+1}$ is the output at instant $t+1$ to be predicted. p represents the amount of past inputs used in the model while q is the number of past exogenous variables used in the model. Note that model $h(.)$ could be either a linear model or a non-linear one. In our case this model will be a LS-SVM.

Long-term forecasting can be defined in a similar way, as a straightforward extension of (17):

$$(\hat{x}_{t+h}, ..., \hat{x}_{t+2}, \hat{x}_{t+1}) = h(x_t, x_{t-1}, ..., x_{t-p}, u_t, u_{t-1}, ..., u_{t-q}, \theta) \quad (18)$$

where h denotes the final time horizon of the long-term forecasting. In this last relation, the goal of the long-term forecasting is clearly illustrated: obtaining the whole set of h future values at time t (current time). The question arising now is how such a long-term forecasting can be obtained. Two methods are proposed here: the rolling forecasting, and the block forecasting.

B. *The rolling forecasting strategy*

This first strategy is a recursive one. Consider one has a one-step ahead prediction model. The idea to obtain long-term forecasting with this short-term forecasting model is straightforward. At time t, all what has to be done is to predict the output at time $t+1$ as usually. Then the prediction at time $t+1$ can be used to predict the output at $t+2$; this processus is repeated recursively up to the final time horizon h. For example, if we consider here $h = 5$, we can write:

$$\begin{aligned} \hat{x}_{t+1} &= h(x_t, x_{t-1}, x_{t-2}, ..., x_{t-p}, \theta), \\ \hat{x}_{t+2} &= h(\hat{x}_{t+1}, x_t, x_{t-1}, ..., x_{t-p+1}, \theta), \\ &... \\ \hat{x}_{t+5} &= h(\hat{x}_{t+4}, \hat{x}_{t+3}, \hat{x}_{t+2}, ..., x_{t-p+4}, \theta), \end{aligned} \quad (19)$$

where the exogenous variable have been omitted for the sake of simplicity.

C. *The block forecasting strategy*

This second strategy is a direct one. The idea is an immediate application of relation (18), i.e. the use of a multiple output model. The number of outputs is the same as the time horizon h. For the above example with $h = 5$, the model has 5 outputs.

D. *Comments*

The main problem of the rolling forecasting strategy is that there is a certain amount of error between the prediction $\hat{x}_{t+1}$ and the true next value x_{t+1}. As the first prediction is taken as input to obtain the second one, this error is propagated through the model. The second prediction has potentially twice more error: the difference between $\hat{x}_{t+2}$ and x_{t+2} plus the propagated error. With an increasing time horizon h, this accumulation can be important. The block forecasting strategy avoids this problem since the recursive step is avoided. On the other hand, the main problem with the block forecasting strategy is that a single model can perform badly on multiple outputs. Indeed, it has to model various dynamics since the relation with the p last inputs x_t (and potentially the q last exogenous variables u_t) is not necessarily the same for x_{t+1}, x_{t+2} and x_{t+h}. The model should be able to capture various dynamics with a parameter set θ of limited size, which is obviously a difficult task. The rolling forecasting approach is not subject to this problem since only the next value dynamics is modelled. Advantages and drawbacks of the two proposed methods have been discussed briefly. The aim now is to observe how this intuition is verified when applying those two strategies with a specific model, namely the LS-SVM. Furthermore a particular attention will be given to the influence of the model parameters σ and γ while using both long-term forecasting strategies.

VI. SantaFe A Time Series

The Time Series used here is a well-known benchmark: the SantaFe A time series [11]. The number of samples is 1000. This series is represented in Fig. 5.

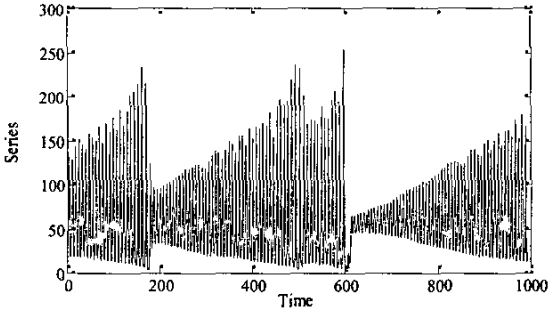

Figure 5: SantaFe A Time Series.

The model based on relation (18) that is used here is:

$$\hat{x}_{t+1} = h(x_t, x_{t-1}, x_{t-2}, \theta). \quad (20)$$

The number of inputs (number of past values of x in (20)) has been selected using the Fast Bootstrap .632 according to the procedure described in the following. The generalization error (defined as the sum of squared errors on all predictions) with respect to the number of inputs is

represented in Fig.6. For each value in Fig.6, parameters σ and γ have been optimized. To obtain the optimized σ and γ, the γ parameter is optimized by Fast Bootstrap .632 for each value of σ, in a selected range. Note that the generalization error used here is a one-step ahead prediction error. The minimum of the generalization error in Fig.3 corresponds to a regressor of size 3 (as already mentioned in (20)). For this size of the regressor, the generalization error with respect to σ is represented in Fig.7 (each value on this curve is the result of a Fast Bootstrap .632 optimization on γ). The minimum is situated at σ = 60. Fig.8 shows the generalization error with respect to γ, for a fixed value of $\sigma = 60$.

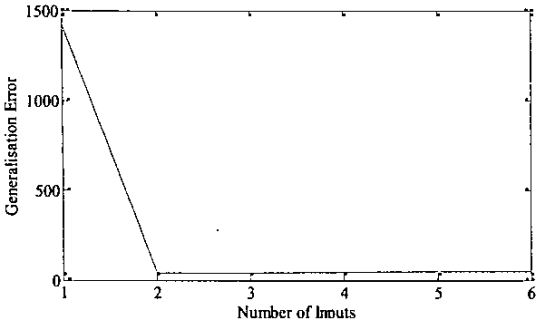

Figure 6: Generalization error with respect to the number of inputs.

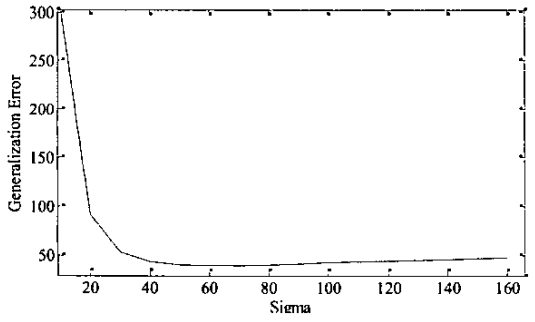

Figure 7: Generalization error with respect to σ.

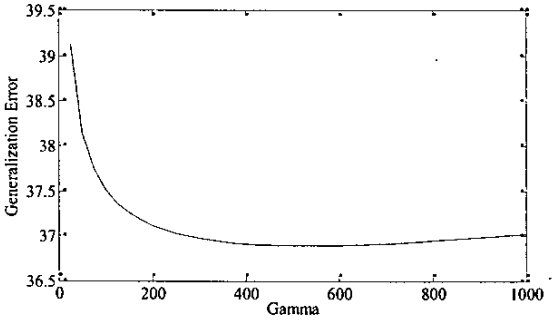

Figure 8: Generalization error with respect to γ.

The generalization error is minimum for σ = 60 and γ = 525. To explain Fig.8 and justify the use of the Fast Bootstrap .632 procedure, the apparent error and the optimism (for $\sigma = 60$) are represented in Fig.9 and Fig.10 respectively.

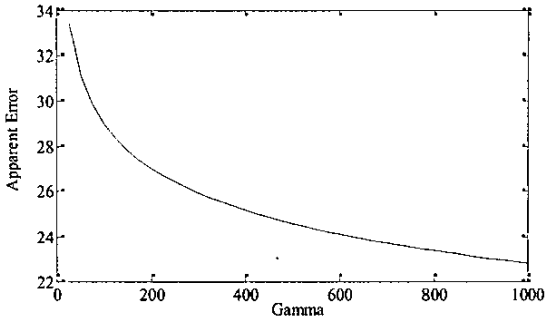

Figure 9: Apparent Error with respect to γ.

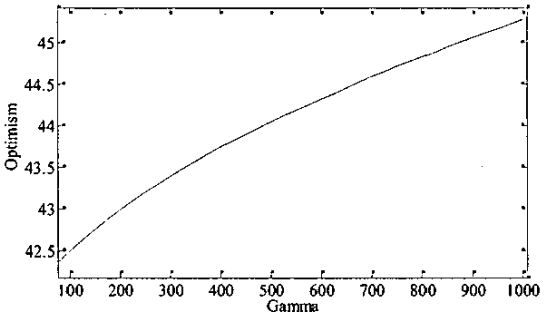

Figure 10: Optimism with respect to γ.

In Fig.10, the optimism is an exponentially increasing function of γ (similarly to Fig.3). The whole procedure above to select σ, γ and the size of the regressor is based on one-step ahead predictions and errors. In order to extend the procedure to long-term forecasting, the same methodology is applied for t+2, t+3, ..., t+10, first with the block forecasting strategy:

$$\begin{aligned} \hat{x}_{t+1} &= h_1(x_t, x_{t-1}, x_{t-2}, \theta), \\ \hat{x}_{t+2} &= h_2(x_t, x_{t-1}, x_{t-2}, \theta), \\ &\cdots \\ \hat{x}_{t+10} &= h_{10}(x_t, x_{t-1}, x_{t-2}, \theta), \end{aligned} \quad (21)$$

In (21), each model h_i corresponds to a LS-SVM. We assume that σ = 60 remains valid and only γ is optimized using Fast Bootstrap. According to equation (19), the rolling forecast methodology is also used:

$$\begin{aligned} \hat{x}_{t+1} &= h(x_t, x_{t-1}, x_{t-2}, \theta), \\ \hat{x}_{t+2} &= h(\hat{x}_{t+1}, x_t, x_{t-1}, \theta), \\ &\cdots \\ \hat{x}_{t+10} &= h(\hat{x}_{t+9}, \hat{x}_{t+8}, \hat{x}_{t+7}, \theta), \end{aligned} \quad (22)$$

We again assume that $\sigma = 60$ remains valid and only γ is optimized using Fast Bootstrap. In this last case, it is interesting to note that the optimal γ that is selected is very different from the one selected in the one-step ahead prediction case. This is shown in Fig.11.

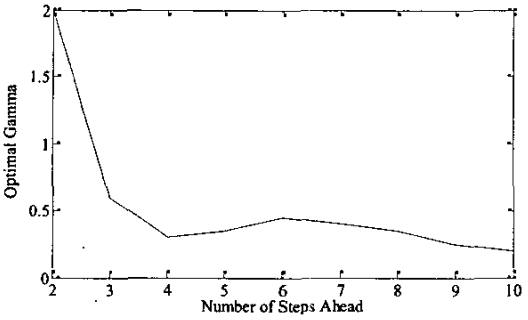

Figure 11: Optimal γ with respect to the number of steps ahead.

The hyperparameter γ is decreasing with respect to the horizon of prediction. The model that is selected is thus a less complex one. This result is in accordance to the results obtained in [3] for linear models. The results of the block forecast and rolling forecast strategies are presented in Fig.12.

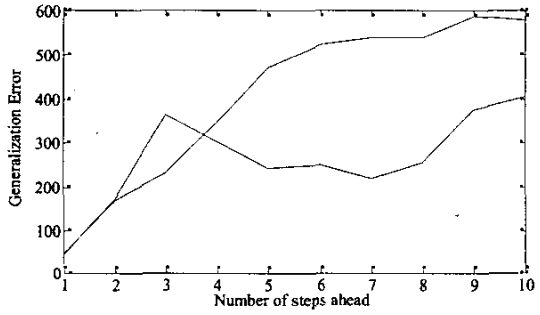

Figure 12: Generalization error with respect to the number of steps ahead: for rolling forecast (solid line) and block forecast (dashed line).

For prediction horizons less or equal to 4, the rolling forecast performs better than the block forecast strategy, while the opposite becomes true for larger horizons. This result is in accordance to the intuition presented in section V.

VII. Conclusions

In this paper, the ability of the Fast Bootstrap to select the hyperparameters of a LS-SVM has been shown on Toy example and a time series prediction benchmark. The main limitation of the Bootstrap is its computational load; the Fast Bootstrap is 10 to 100 times faster.

Two strategies for long-term prediction have been presented: the rolling forecasting strategy and the block forecasting one. Both of them have been applied to the SantaFe A prediction benchmark. Firstly, the rolling strategy provides better predictions for short horizons while the opposite becomes true for larger ones. Secondly, for the rolling forecasting strategy, the hyperparameter γ is decreasing with the horizon of prediction. These two results are obtained in a fast and efficient manner with the Fast Bootstrap methodology.

VIII. Acknowledgements

Michel Verleysen is Senior Research Associate of the Belgian National Fund for Scientific Research (FNRS). G. Simon is funded by the Belgian F.R.I.A. Part the work of V. Wertz is supported by the Interuniversity Attraction Poles (IAP), initiated by the Belgian Federal State, Ministry of Sciences, Technologies and Culture. Part the work of A. Lendasse is supported by the project New Information Processing Principles, 44886, of the Academy of Finland. The scientific responsibility rests with the authors.

IX. References

[1] H. Akaike, Information theory and an extension of the maximum likelihood principle, 2nd Int. Symp. on information Theory, 267-81, Budapest, 1973.

[2] G. Schwarz, "Estimating the dimension of a model", Ann. Stat. 6, 461-464, 1978.

[3] L. Ljung, System Identification - Theory for the user, 2nd Ed, Prentice Hall, 1999.

[4] B. Efron, R. J. Tibshirani, An introduction to the Bootstrap, Chapman & Hall, 1993.

[5] M. Stone, An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, J. Royal. Statist. Soc., B39, 44-7, 1977.

[6] R. Kohavi, A study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection, Proc. of the 14th Int. Joint Conf. on A.I., Vol. 2, Canada, 1995.

[7] J.A.K. Suykens, L. Lukas, J. Vandewalle, Sparse approximation using least squares support vector machines', in Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS 2000), Geneva, Switzerland, May 2000, pp. II757-II760.

[8] J.A.K. Suykens, L. Lukas, J. Vandewalle, Sparse Least Squares Support Vector Machine Classifiers", in Proc. of the European Symposium on Artificial Neural Networks (ESANN'2000), Bruges, Belgium, 2000, pp. 37-42.

[9] Fast Approximation of the Bootstrap for Model Selection
G. Simon, A. Lendasse, V. Wertz, M. Verleysen, ESANN 2003, European Symposium on Artificial Neural Networks, Bruges (Belgium), 23-25 April 2003, pp. 99-106.

[10] Bootstrap for Model Selection: Linear Approximation of the Optimism
G. Simon, A. Lendasse, M. Verleysen, IWANN 2003, International Work-Conference on Artificial and Natural Neural Networks, Mao, Menorca (Spain), June 3-6, 2003. Computational Methods in Neural Modeling, J. Mira, J.R. Alvarez eds, Springer-Verlag, Lecture Notes in Computer Science 2686, 2003, pp. I182-I189.

[11] A.S. Weigend, N.A. Gershenfeld, Times Series Prediction: Forecasting the future and Understanding the Past, Addison-Wesley, Reading, MA, 1994.