2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Short Term Chaotic Time Series Prediction using Symmetric LS-SVM Regression

Marcelo Espinoza, Johan A.K. Suykens and Bart De Moor

Katholieke Universiteit Leuven, Department of Electrical Engineering (ESAT), SCD/SISTA
Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium
Email: {marcelo.espinoza,johan.suykens}@esat.kuleuven.be

Abstract—In this article, we illustrate the effect of imposing symmetry as prior knowledge into the modelling stage, within the context of chaotic time series predictions. It is illustrated that using Least-Squares Support Vector Machines with symmetry constraints improves the simulation performance, for the cases of time series generated from the Lorenz attractor, and multi-scroll attractors. Not only accurate forecasts are obtained, but also the forecast horizon for which these predictions are obtained is expanded.

1. Introduction

In applied nonlinear time series analysis, the estimation of a nonlinear black-box model in order to produce accurate forecasts starting from a set of observations is common practice. Usually a time series model is estimated based on available data up to time t, and its final assessment is based on the simulation performance from $t+1$ onwards. Due to the nature of time series generated by chaotic systems, where the series not only shows nonlinear behavior but also drastic regime changes due to local instability of attractors, this is a very challenging task. For this reason, chaotic time series have been used as benchmark in several time series competitions [10, 9].

The modelling of chaotic time series can be improved by exploiting some of its properties. If the true underlying system is symmetric, this information can be imposed to the model as prior knowledge [3], in which case it is possible to obtain better forecasts than those obtained with a general model [1]. In this article, short term predictions for chaotic time series are generated using Least-Squares Support Vector Machines (LS-SVM) regression. We show that LS-SVM with symmetry constraints can produce accurate predictions. Not only accurate forecasts are obtained, but also the forecast horizon for which these predictions are obtained is expanded, when compared with the unconstrained LS-SVM formulation.

This paper is structured as follows. Section 2 describes the LS-SVM technique for regression, and how symmetry can be imposed in a straightforward way. Section 3 describes the applications for the cases of the $x-$coordinate of the Lorenz attractor, and the data generated by a nonlinear transformation of multi-scroll attractors.

2. LS-SVM with Symmetry Constraints

Least-Squares Support Vector Machines (LS-SVM) is a powerful nonlinear black-box regression method, which builds a linear model in the so-called feature space where the inputs have been transformed by means of a (possibly infinite dimensional) nonlinear mapping $\boldsymbol{\varphi}$ [7]. This is converted to the dual space by means of the Mercer's theorem and the use of a positive definite kernel, without computing explicitly the mapping $\boldsymbol{\varphi}$. The LS-SVM formulation, solves a linear system in dual space under a least-squares cost function [8], where the sparseness property can be obtained by e.g. sequentially pruning the support value spectrum [6] or via a fixed-size subset selection approach [7]. The LS-SVM training procedure involves the selection of a kernel parameter and the regularization parameter of the cost function, which can be done e.g. by cross-validation, Bayesian techniques [4] or others. The inclusion of a symmetry constraint (odd or even) to the nonlinearity within the LS-SVM regression framework can be formulated as follows [2]. Given the sample of N points $\{\boldsymbol{x}_k, y_k\}_{k=1}^N$, with input vectors $\boldsymbol{x}_k \in \mathbb{R}^p$ and output values $y_k \in \mathbb{R}$, the goal is to estimate a model of the form

$$y = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}) + b + e, \tag{1}$$

where $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^{n_h}$ is the mapping to a high dimensional (and possibly infinite dimensional) feature space, and the residuals e are assumed to be i.i.d. with zero mean and constant (and finite) variance. The following optimization problem with a regularized cost function is formulated:

$$\min_{\boldsymbol{w},b,e_k} \quad \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2$$
$$\text{s.t.} \quad \begin{cases} y_k = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) + b + e_k, & k = 1,\ldots,N, \\ \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_k), & k = 1,\ldots,N, \end{cases} \tag{2}$$

where a is a given constant which can take either -1 or 1. The first restriction is the standard model formulation in the LS-SVM framework. The second restriction is a shorthand for the cases where we want to impose the nonlinear function $\boldsymbol{w}^T\boldsymbol{\varphi}(x_k)$ to be even (resp. odd) by using $a = 1$ (resp. $a = -1$). The solution is formalized in the following lemma.

Lemma 1 *[2] Given the problem (2) and a positive definite kernel function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ satisfying the assumptions $K(\boldsymbol{x}_k, -\boldsymbol{x}_l) = K(-\boldsymbol{x}_k, \boldsymbol{x}_l)$ and $K(-\boldsymbol{x}_k, -\boldsymbol{x}_l) = K(\boldsymbol{x}_k, \boldsymbol{x}_l)$ $\forall k, l = 1, \ldots, N$, the solution to (2) is given by the system*

$$\left[\begin{array}{c|c} \frac{1}{2}(\boldsymbol{\Omega} + a\boldsymbol{\Omega}^*) + \frac{1}{\gamma}\mathbf{I} & \mathbf{1} \\ \hline \mathbf{1}^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \hline 0 \end{array}\right], \quad (3)$$

with $\boldsymbol{\Omega}_{k,l} = K(\boldsymbol{x}_k, \boldsymbol{x}_l)$ and $\boldsymbol{\Omega}^*_{k,l} = K(-\boldsymbol{x}_k, \boldsymbol{x}_l)$ $\forall k, l = 1, \ldots, N$.

Proof. The Lagrangian for (2) is given by $\mathcal{L}(\boldsymbol{w}, b, e_k, \alpha_k, \beta_k) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\frac{1}{2}\sum_{k=1}^N e_k^2 - \sum_{k=1}^N(\boldsymbol{w}^T\boldsymbol{\varphi}(x_k) + b + e_k - y_k) - \sum_{k=1}^N(\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) - a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_k))$, with $\alpha_k, \beta_k \in \mathbb{R}$ the Lagrange multipliers. Taking the optimality conditions $\frac{\partial\mathcal{L}}{\partial\boldsymbol{w}} = 0$, $\frac{\partial\mathcal{L}}{\partial b} = 0$, $\frac{\partial\mathcal{L}}{\partial e_k} = 0$, $\frac{\partial\mathcal{L}}{\partial\beta_k} = 0$ $\frac{\partial\mathcal{L}}{\partial\alpha_k} = 0$, yields the following system of equations: $\boldsymbol{w} = \sum_{l=1}^N (\alpha_l + \beta_l)\boldsymbol{\varphi}(\boldsymbol{x}_l) - a\sum_{l=1}^N \beta_i\boldsymbol{\varphi}(-\boldsymbol{x}_i)$, $\sum_{l=1}^N \alpha_i = 0$, $\gamma e_k = \alpha_k$, $y_k = \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) + b + e_k, \boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) = a\boldsymbol{w}^T\boldsymbol{\varphi}(-\boldsymbol{x}_k)$, $k = 1, \ldots, N$.

Applying Mercer's theorem, $\boldsymbol{\varphi}(\boldsymbol{x}_k)^T\boldsymbol{\varphi}(\boldsymbol{x}_l) = K(\boldsymbol{x}_k, \boldsymbol{x}_l)$ for a positive definite kernel function $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$[7]. Under the assumptions that $K(\boldsymbol{x}_k, -\boldsymbol{x}_l) = K(-\boldsymbol{x}_k, \boldsymbol{x}_l)$ and $K(-\boldsymbol{x}_k, -\boldsymbol{x}_l) = K(\boldsymbol{x}_k, \boldsymbol{x}_l)$ $\forall k, l = 1, \ldots, N$, the elimination of $\boldsymbol{w}, e_k$ and β_k gives

$$y_k = \frac{1}{2}\sum_{l=1}^N \alpha_l[K(\boldsymbol{x}_l, \boldsymbol{x}_k) + aK(-\boldsymbol{x}_l, \boldsymbol{x}_k)] + b + \frac{1}{\gamma}\alpha_k \quad (4)$$

and the final Karush-Kuhn-Tucker (KKT) system can be written as (3).

Remark 1 ***[Kernel functions]*** *For $K(\boldsymbol{x}_k, \boldsymbol{x}_l)$ there are usually the following choices: $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \boldsymbol{x}_k^T\boldsymbol{x}_l$ (linear kernel); $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = (\boldsymbol{x}_k^T\boldsymbol{x}_l + c)^d$ (polynomial of degree d, with c a tuning parameter); $K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \exp(-||\boldsymbol{x}_k - \boldsymbol{x}_l||_2^2/\sigma^2)$ (RBF kernel), where σ is a tuning parameter.*

Remark 2 ***[Equivalent Kernel]*** *The final model becomes*

$$\hat{y}(\boldsymbol{x}) = \sum_{l=1}^N \alpha_l K_{eq}(\boldsymbol{x}_l, \boldsymbol{x}) + b. \quad (5)$$

where $K_{eq}(\boldsymbol{x}_l, \boldsymbol{x}) = \frac{1}{2}[(K(\boldsymbol{x}_l, \boldsymbol{x}) + aK(-\boldsymbol{x}_l, \boldsymbol{x})]$ is the equivalent symmetric kernel that embodies the restriction about the nonlinearity. It is important to note that the final KKT system (3) has the same dimensions as the KKT obtained with standard LS-SVM. Therefore, imposing the second constraint does not increase the dimension of the system, as the new information is translated into the kernel level.

3. Application to Chaotic Time Series

In this section, the effects of imposing symmetry to the LS-SVM are presented for two cases of chaotic time series. On each example, an RBF kernel is used and the parameters σ and γ are found by 10-fold cross validation over the corresponding training sample. The results using the standard LS-SVM are compared to those obtained with the symmetry-constrained LS-SVM (S-LS-SVM) from (2). The examples are defined in such a way that there are not enough training datapoints on every region of the relevant space; thus, it is very difficult for a black-box model to "learn" the symmetry just by using the available information. The examples are compared in terms of the performance in the training sample (cross-validation mean squared error, MSE-CV) and the generalization performance (MSE out of sample, MSE-OUT). For each case, a Nonlinear AutoRegressive (NAR) black-box model is formulated:

$$y(t) = g(y(t-1), y(t-2), \ldots, y(t-p)) + e(t)$$

where g is to be identified by LS-SVM and S-LS-SVM. The order p is selected during the cross-validation process as an extra parameter. After each model is estimated, they are used in simulation mode, where the future predictions are computed with the estimated model $\hat{\boldsymbol{\varphi}}$ using past predictions:

$$\hat{y}(t) = \hat{g}(\hat{y}(t-1), \hat{y}(t-2), \ldots, \hat{y}(t-p)).$$

3.1. Lorenz attractor

This example is taken from [1]. The $x-$coordinate of the Lorenz attractor is used as an example of a time series generated by a dynamical system. A sample of 1000 datapoints is used for training, which corresponds to an unbalanced sample over the evolution of the system, shown on Figure 1 as a time-delay embedding. Figure 2 (top) shows the training sequence (thick line) and the future evolution of the series (test zone). Figure 2 (bottom) shows the simulations obtained from both models on the test zone. Results are presented on Table 1. Clearly the S-LS-SVM can simulate the system for the next 500 timesteps, far beyond the 100 points that can be simulated by the LS-SVM.

3.2. Multi-scroll attractors

This dataset was used for the K.U.Leuven Time Series Prediction Competition [9]. The series was gener-

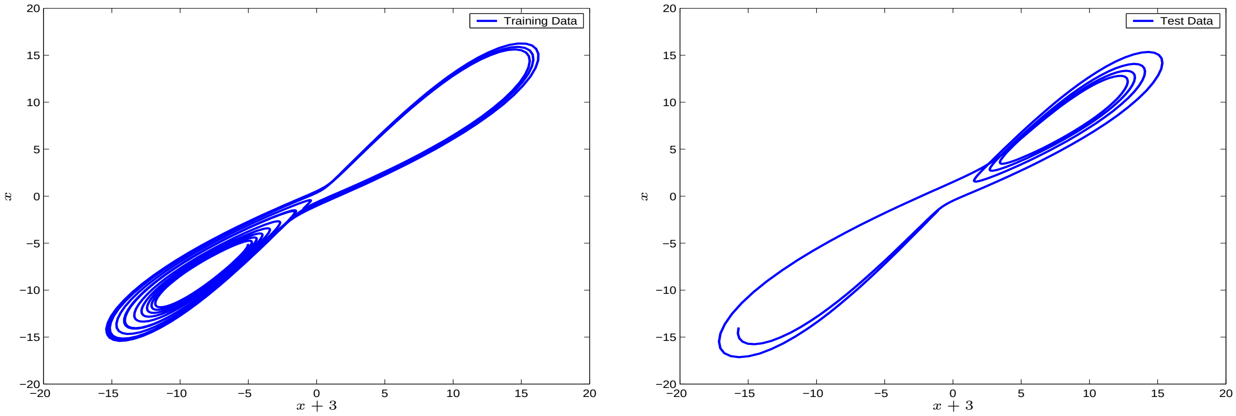

Figure 1: The training (left) and test (right) series from the $x-$coordinate of the Lorenz attractor

	LS-SVM	S-LS-SVM
MSE-CV	3.41×10^{-4}	1.62×10^{-4}
MSE-OUT	52.057	0.085

Table 1: Performance of LS-SVM and S-LS-SVM on the Lorenz data.

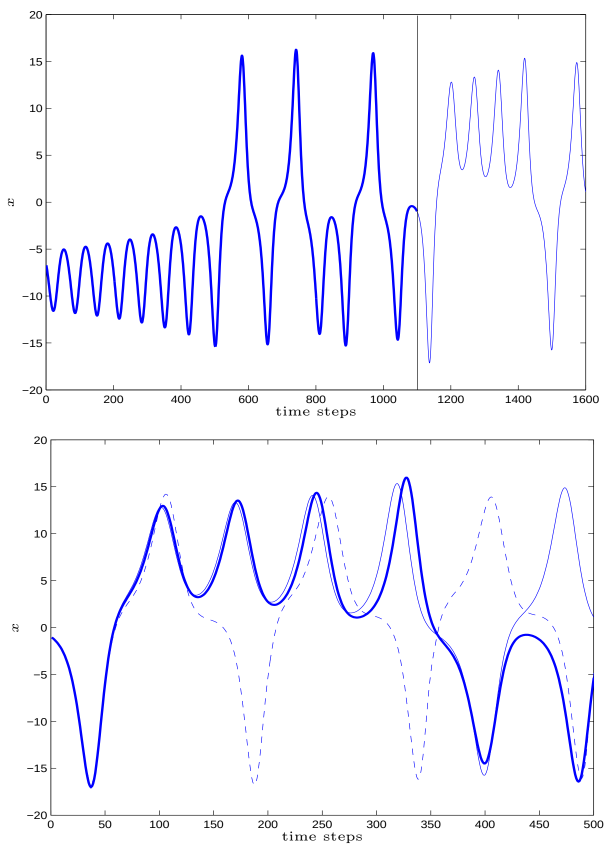

Figure 2: (Top) The series from the $x-$coordinate of the Lorenz attractor, part of which is used for training (thick line). (Bottom) Simulations with LS-SVM (dashed line), S-LS-SVM (thick line) compared to the actual values (thin line).

ated by

$$\begin{aligned} \dot{\boldsymbol{x}} &= h(\boldsymbol{x}) \\ \boldsymbol{y} &= \boldsymbol{W} \tanh\left(\boldsymbol{V}\boldsymbol{x}\right) \end{aligned} \tag{6}$$

where h is the multi-scroll equation, $\boldsymbol{x}$ is the 3-dimensional coordinate vector, and $\boldsymbol{W}, \boldsymbol{V}$ are the interconnection matrices of the nonlinear function (a 3-units multilayer perceptron, MLP). This MLP function hides the underlying structure of the attractor [5]. A training set of 2,000 points was available for model estimation, shown on Figure 3, and the goal was to predict the next 200 points out of sample. The winner of the competition followed a complete methodology involving local modelling, specialized many-steps ahead cross-validation parameters tuning, and the exploitation of the symmetry properties of the series (which he did by flipping the series around the time axis).

Following the winner approach, both LS-SVM and S-LS-SVM are trained using 10-step-ahead cross-validation for hyperparameters selection. To illustrate the difference between both models, the out of sample MSE is computed considering only the first n simulation points, where $n = 20, 50, 100, 200$. It is important to emphasize that both models are trained using exactly the same methodology for order and hyperparameter selection; the only difference is the symmetry constraint for the S-LS-SVM case. Results are reported on Table 2. The simulations from both models are shown on Figure 4.

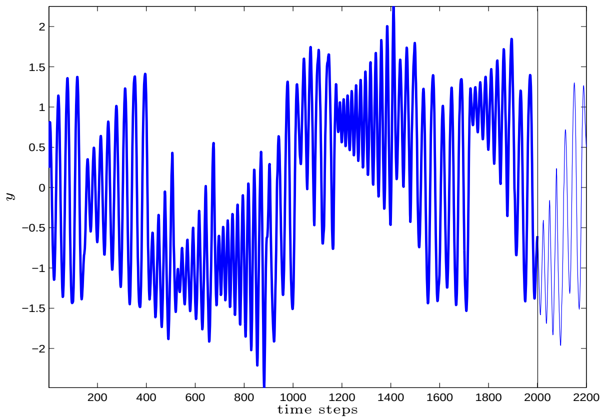

Figure 3: The training sample (thick line) and future evolution (thin line) of the series from the K.U.Leuven Time Series Competition

	LS-SVM	S-LS-SVM
MSE-CV	0.15	0.11
MSE-OUT (1-20)	0.03	0.03
MSE-OUT (1-50)	0.05	0.03
MSE-OUT (1-100)	0.05	0.03
MSE-OUT (1-200)	0.64	0.24

Table 2: Performance of LS-SVM and S-LS-SVM on the K.U.Leuven data.

4. Conclusions

For the task of chaotic time series prediction, we have illustrated how to use LS-SVM regression with symmetry constraints to improve the simulation performance for the cases of series generated by Lorenz attractor and multi-scroll attractors. By adding symmetry constraints to the LS-SVM formulation, it is possible to embed the information about symmetry into the kernel level. This translates not only in better predictions for a given time horizon, but also on a larger forecast horizon in which the model can track the time series into the future.

Acknowledgments. This work is supported by grants and projects for the Research Council K.U.Leuven (GOA-Mefisto 666, GOA- Ambiorics, several PhD/ Postdocs & fellow grants), the Flemish Government (FWO: PhD/ Postdocs grants, projects G.0211.05, G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, G.0120.03, G.0452.04, G.0499.04, ICCoS, ANMMM; AWI; IWT: PhD grants, GBOU (McKnow) Soft4s), the Belgian Federal Government (Belgian Federal Science Policy Office: IUAP V-22; PODO-II (CP/ 01/40), the EU (FP5- Quprodis; ERNSI, Eureka 2063- Impact; Eureka 2419- FLiTE) and Contracts Research / Agreements (ISMC /IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard). J. Suykens and B. De Moor are an associate professor and a full professor with K.U.Leuven, Belgium, respectively. The scientific responsibility is assumed by its authors.

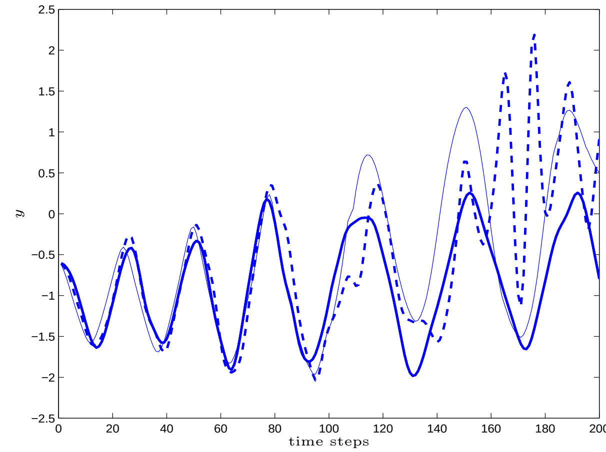

Figure 4: Simulations with LS-SVM (dashed line), S-LS-SVM (thick line) compared to the actual values (thin line) for the next 200 points of the K.U.Leuven data.

References

[1] L.A. Aguirre, R. Lopes, G. Amaral, and C. Letellier. Constraining the topology of neural networks to ensure dynamics with symmetry properties. *Physical Review E*, 69, 2004.

[2] M. Espinoza, J.A.K. Suykens, and B. De Moor. Imposing symmetry in least squares support vector machines regression. In *Proc. of the 44nd IEEE Conference on Decision and Control (CDC)*, 2005.

[3] T. Johansen. Identification of non-linear systems using empirical data and prior knowledge-an optimization approach. *Automatica*, 32(3):337–356, 1996.

[4] D.J.C. MacKay. Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11:1035–1068, 1999.

[5] J. McNames, J.A.K. Suykens, and J. Vandewalle. Winning entry of the k.u.leuven time series prediction competition. *International Journal of Bifurcation and Chaos*, 9(8):1485–1500, 1999.

[6] J.A.K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, 48(1-4):85–105, 2002.

[7] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[8] J.A.K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 1999.

[9] J.A.K. Suykens and J Vandewalle. The K.U.Leuven competition data : a challenge for advanced neural network techniques. In *Proc. of the European Symposium on Artificial Neural Networks (ESANN'2000)*, pages 299–304, Brugges,Belgium, 2000.

[10] A.S. Weigend and N.A. Gershenfeld, editors. *Time Series Prediction. Forecasting the Future and Understanding the past*. Addison-Wesley, Reading, MA, 1993.