Available online at www.sciencedirect.com
ScienceDirect
International Journal of Forecasting 27 (2011) 725–739
www.elsevier.com/locate/ijforecast

A heuristic method for parameter selection in LS-SVM: Application to time series prediction

Ginés Rubio*, Héctor Pomares, Ignacio Rojas, Luis Javier Herrera

Department of Computer Architecture and Computer Technology, University of Granada, C/ Periodista Daniel Saucedo sn, 18071 Granada, Spain

Available online 23 May 2010

Abstract

Least Squares Support Vector Machines (LS-SVM) are the state of the art in kernel methods for regression. These models have been successfully applied for time series modelling and prediction. A critical issue for the performance of these models is the choice of the kernel parameters and the hyperparameters which define the function to be minimized. In this paper a heuristic method for setting both the σ parameter of the Gaussian kernel and the regularization hyperparameter based on information extracted from the time series to be modelled is presented and evaluated.

Keywords: Least squares support vector machines; Gaussian kernel parameters; Hyperparameters optimization; Time series prediction

1. Introduction

Time series analysis and prediction refers to the branch of statistics where observations are collected sequentially in time, usually (but not necessarily) at equally-spaced time points, and the analysis relies, at least in part, on understanding or exploiting the dependence among the observations. The ultimate goal of this field is to be able to obtain some information about the series in order to predict future values. Because of the importance of time series analysis, many works can be found in the literature, especially those based on statistical models. Of these, a special mention should be given to Box and Jenkins's ARIMA (Autoregressive Integrated Moving Average) time series analysis, because it provides a comprehensive statistical modelling methodology for I/O processes. It covers a wide variety of patterns, ranging from stationary to non-stationary and seasonal (periodic) time series, and has been used extensively in the literature (Mélard & Pasteels, 2000; Valenzuela et al., 2008). However, the Box-Jenkins methodology is sometimes inadequate for situations where the relationships between the samples are not linear with time. In fact, the drawbacks of these linear methods (Box & Jenkins, 1976; Kantz & Schreider, 1997; Mélard & Pasteels, 2000;

* Corresponding author.
E-mail addresses: grubio@atc.ugr.es (G. Rubio), hector@ugr.es (H. Pomares), ignacio@atc.ugr.es (I. Rojas), jherrera@atc.ugr.es (L.J. Herrera).

0169-2070/$ - see front matter
doi:10.1016/j.ijforecast.2010.02.007

Weigend & Gershenfeld, 1994), in combination with the development of artificial intelligence, have led to the development of alternative solutions using non-linear modelling. Two of the main forecasting techniques that allow for the detection and modelling of non-linear data are rule induction and neural networks. Rule induction identifies patterns in the data and expresses them as rules. Expert systems and fuzzy systems are examples of this technique (Adya, Collopy, Kennedy, & Armstrong, 2001; Kim & Kim, 1997; Lee & Kim, 1994). On the other hand, Artificial Neural Networks (ANN) represent an important class of non-linear prediction models that has generated a considerable amount of interest in the forecasting community over the past decade (Adya & Collopy, 1998; Alves da Silva, Ferreira, & Velasquez, 2008; Balkin & Ord, 2000; Teräsvirta, Medeiros, & Rech, 2006; Zhang, Patuwo, & Hu, 1998), essentially because, under certain conditions, they are universal approximators. For instance, we can find applications in the literature to time series prediction using radial basis functions (Rojas et al., 2000a,c), multilayer perceptrons (Coulibaly, Anctil, & Bobee, 2001; Shiblee, Kalra, & Chandra, 2009), and hybrid systems that combine several of the aforementioned methodologies (Hsieh & Tang, 1998; Jang, 1993; Jursa & Rohrig, 2008; Zhang, 2003).

Over the last few years, kernel methods (Scholkopf & Smola, 2001) have proved capable of forecasting more accurately than other techniques such as neural networks, neuro-fuzzy systems or linear models (ARIMA), in terms of various different evaluation measures during both the validation and test phases (Hong & Pai, 2006; Liu, Liu, Zheng, & Liang, 2004; Wang, Chau, Cheng, & Qiu, 2009; Xu, Tian, & Jin, 2006). Kernel methods are defined by operations over the kernel function values for the data, ignoring the structure of the input data and avoiding the curse of dimensionality problem (Bellman, 1966). Their main problem is their inefficiency when the amount of data grows. The fundamental motivation for using kernel methods in the field of time series prediction is their ability to forecast time series data accurately when the underlying model could be non-linear, non-stationary and not defined a priori (Sapankevych & Sankar, 2009).

The two most promising kernel methods for time series prediction are Support Vector Regression (SVR) (Misra, Oommen, Agarwal, Mishra, & Thompson, 2009; Sapankevych & Sankar, 2009; Zhou, Bai, Zhang, & Tian, 2008) and Least Square Support Vector Machines (LS-SVM) (Van Gestel et al., 2001; Xu & Bian, 2005).

SVR has some drawbacks, mainly associated with its formulation and efficiency. Least Squares Support Vector Machines (LS-SVM) are a modification of the standard SVR formulation introduced to overcome these disadvantages (Suykens, Van Gestel, De Brabanter, De Moor, & Vandewalle, 2002), and the resulting optimization problem therefore has half the number of parameters and the model is optimized by solving a linear system of equations instead of a quadratic problem. Both SVR and LS-SVM have been applied to time series prediction with promising results, as can be seen in the work of Müller, Smola, Rätsch, Schölkopf, Kohlmorgen, and Vapnik (1999), Tay and Cao (2001) and Thiessen and Van Brakel (2003) for SVR and Van Gestel et al. (2001) and Xu and Bian (2005) for LS-SVM. However, both have some common drawbacks:

1. *The selection of the kernel function could be difficult.* Although there is wide diversity in the literature related to kernel methods, most of these works use the Gaussian kernel or kernels based on it for regression problems and time series prediction, especially because of the smooth interpolation they provide.
2. *The optimization of the parameters is computationally intensive.* This optimization process requires the evaluation of some cross-validation (CV) procedures (An, Liu, & Venkatesh, 2007; Ying & Keong, 2004) or some Bayesian criteria (Van Gestel et al., 2001) with a complexity of $O(N^3)$, where N is the number of training points. In the case of SVR, an interesting study by Cherkassky and Ma (2004) gives a guide to setting the values of the hyperparameters. In the LS-SVM case, a recent study by Liitiäinen, Lendasse, and Corona (2007) has an interesting approach to the training of a LS-SVM with a Gaussian kernel based on the use of a Non-parametric Noise Estimation (NNE) technique, which tries to estimate an upper limit to the approximation accuracy that can be reached by any model for the given data.

3. *The generated models could be huge*, because they include all training data inside. In order to alleviate this problem, a pruning method can be used on the generated models a posteriori, to reduce the number of samples retained (Cawley & Talbot, 2002; de Kruif & de Vries, 2003).

A correct procedure for the setting and optimization of the parameters of a LS-SVM model for regression is of critical importance in enabling us to obtain good performances and avoid excessive computation times. Time series prediction is an especially difficult case of regression, in that it is very sensitive to the values of the parameters in order to get models with good accuracy and generalization abilities. The results of LS-SVM models for time series prediction can be improved with a correct initialization of their parameters and the application of a good optimization method. In this paper we present a method for estimating both the Gaussian kernel parameter and the regularization hyperparameter of the optimization process of a LS-SVM model, based on the use of a non-parametric noise estimation technique and some information extracted from the time series of the model.

The rest of the paper is organized as follows: in Section 2 a brief description of LS-SVM is given, and in Section 3 the non-parametric noise estimation delta and gamma tests are presented. Section 4 presents a heuristic for obtaining an initial estimate of both the regularization factor and the Gaussian kernel parameter σ of a LS-SVM model for time series modelling from data; Section 5 presents and explains the experiments; the results obtained are shown and commented on in Section 6; and, finally, some conclusions are drawn in Section 7.

2. Least squares support vector machines

Given a set of function samples $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^d \times \mathbb{R}$, where N is the number of samples and d is the dimensionality of the input data, the classic unbiased LS-SVM model for regression tries to approximate a zero-mean function $y = f(x)$ that relates the inputs X to the output Y by solving the following optimization problem:

$$\min_{w,e_i} \tau(w, e) = \frac{1}{2}\|w\|^2 + \gamma \frac{1}{2} \sum_{i=1}^{N} \mathrm{e}_i^2, \tag{1}$$

where $y_i - \langle w, x_i \rangle = e_i, \forall i = 1, \ldots, N$, τ is the function to be optimized which depends on the weight vector $w \in \mathbb{R}^d$, γ is a regularization factor, and e_i is the error committed when approximating the ith sample. The problem can be solved using Lagrange multipliers, and amounts to solving the following linear system:

$$[\Omega + I/\gamma]\,[\alpha] = [y]\,, \tag{2}$$

where $\Omega_{ij} = \langle x_i, x_j \rangle$ is the scalar product between a pair of input points, I is the identity matrix and α is the vector of Lagrange multipliers. If the scalar product operation in the *input space* is substituted by a scalar product in a *feature space* given by a kernel function $k(x, x'; \Theta) = \langle \phi(x; \Theta), \phi(x'; \Theta) \rangle$, where ϕ is the function that maps points from the input space to the feature space, then $\Omega_{ij} = k(x_i, x_j; \Theta)$, and the modelled function is given by:

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i; \Theta), \tag{3}$$

and therefore the initial optimization problem would be transformed into:

$$\min_{w,e_i} \tau(w, e) = \frac{1}{2}\|w\|^2 + \gamma \frac{1}{2} \sum_{i=1}^{N} \mathrm{e}_i^2, \tag{4}$$

where $y_i - \langle w, \phi(x_i; \Theta) \rangle = e_i, \forall i = 1, \ldots, N$. This formulation supposes that both the kernel parameters Θ and the regularization factor γ, which is referred to as a hyperparameter in the LS-SVM terminology in order to distinguish it from the kernel parameters, are fixed.

3. Non-parametric noise estimation: delta and gamma tests

The estimation of the noise existing in a set of input/output data pairs has been shown to be especially useful for regression problems, since it can provide an error bound that can be used to determine how accurate a model can be that is trained using these data. In other words, when a model is trained in such a way that the model accuracy is better than the noise present in the data, we should conclude that the model is overfitting the data and should be discarded. This estimation of the noise can also be used to

select the most relevant inputs to the model, as was demonstrated by Lendasse, Corona, Hao, Reyhani, and Verleysen (2006). This section briefly describes the two most popular methods of non-parametric noise estimation.

The delta test for a given set of I/O function samples $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^d \times \mathbb{R}$, where N is the number of samples and d is the dimensionality of the input data, can be defined as:

$$\delta_{N,k} = \frac{1}{2N} \sum_{i=1}^{N} (y_i - y_{nn[i,k]})^2, \tag{5}$$

where $nn[i, k]$ is the index of the kth nearest neighbour to x_i using a given distance measure, usually the Euclidean distance. Since $\delta_{N,1} \approx \sigma_e^2$, where σ_e^2 is the variance of the noise in the output, it is possible to use $\delta_{N,1}$ as an estimation of the best mean quadratic error that could be obtained by a model on the training data without overfitting.

Liitiäinen et al. (2007) describe the limitations of the delta test method for high dimensional data (input vectors of dimension above 4) as far as robustness and reliability are concerned. The authors recommend the use of the gamma test instead, which is a more complex method. Using the same notation as before, let Γ be the empirical ρ-moment of the data set to the k-nearest neighbour, defined as:

$$\Gamma_{N,\rho,k} = \frac{1}{N} \sum_{i=1}^{N} \|x_i - x_{nn[i,k]}\|^{\rho}. \tag{6}$$

Then, the noise estimator Gamma test of the variance of the noise in the output σ_e^2 for a given $k \geq 2$ can be computed as:

$$\sigma_e^2 \approx E_k[\delta_{N,l}] - \frac{E_k[\Gamma_{N,2,l}] \sum_{l=1}^{k} (\Gamma_{N,2,l} - E_k[\Gamma_{N,2,l}])(\delta_{N,l} - E_k[\delta_{N,l}])}{\sum_{l=1}^{k} (\Gamma_{N,2,l} - E_k[\Gamma_{N,2,l}])^2}, \tag{7}$$

where $E_k[\delta_{N,l}] = \frac{1}{k} \sum_{l=1}^{k} \delta_{N,l}$, $E_k[\Gamma_{N,2,l}] = \frac{1}{k} \sum_{l=1}^{k} \Gamma_{N,2,l}$ and $\delta_{N,l}$ is defined in Eq. (5). Essentially, these mathematical expressions tell us that, whereas in the delta test the noise is estimated using basically the nearest neighbour as the approximator model, the gamma test goes beyond that and considers several approximator models which take into account up to the kth nearest neighbour.

It can be demonstrated that both the delta and gamma noise estimator tests converge to the value of the variance of the noise in outputs as the amount of training data increases. Thus, a sufficiently large number of data samples is needed in order to obtain a reliable final value for the tests.

4. Heuristic method for the direct optimization of the LS-SVM parameters

The objective when trying to optimize the parameters of a LS-SVM model for a given set of I/O samples is to optimize the generalization ability of the model, and one way of achieving this is by minimizing the l-fold cross-validation error. In order to do so, we will make use in this paper of a local search method known as the conjugate gradient method. This method is capable of locating the nearest minimum to a given initial guess of the parameters to be optimized, using information from the partial derivatives of the model. In our case, these parameters are the regularization factor, also known as the γ hyperparameter, and the parameters defining the kernel function. The goodness of the local minimum found can depend strongly on these initial values. In this section we give some experimental clues about how to chose these values and provide the reader with the equations needed to do a local search for a LS-SVM model.

4.1. NNE-based guess of the γ hyperparameter

The regularization factor γ in Eq. (4), also known as the hyperparameter γ, is in charge of the trade-off between the smoothness of the model and its accuracy. The bigger the regularization factor the more importance is given to the error of the model in the minimization process. An excessively large value of γ suggests that the model over-fits the data, thus losing its generalization capabilities. Therefore, it is most important to give a proper value to the regularization factor.

In order to give a general expression for the regularization factor, it is apparent that the variance of the data must be taken into account. If we multiply the data by a certain scale factor, the weights in Eq. (4) will also be scaled accordingly. On the other hand, it

would be desirable to have an estimate of the optimal value for the factor containing the error of the model. It is evident that, for a given output range (i.e. scale factor), a model from noise-free data could achieve a much better accuracy than that derived from very noisy data, and, precisely as was explained in the last section, both the delta test and the gamma test can help us in making such estimation.

We therefore propose a heuristic initial value for the regularization factor γ_h as

$$\gamma_h = \sigma_y^2/\hat{\sigma}_e^2, \tag{8}$$

where $\hat{\sigma}_e^2$ is evaluated from data using NNE:

$$\hat{\sigma}_e^2 = \begin{cases} \text{use Eq. (5)} & \text{if the } \textit{number of dimensions} \\ & \textit{of } X \leq 4 \\ \text{use Eq. (7)} & \text{otherwise.} \end{cases} \tag{9}$$

Then the optimization problem of the LS-SVM model given by Eq. (4) can be rewritten as:

$$\min_{w,e_i} \tau(w, e) = \frac{1}{2}\frac{1}{\sigma_y^2}\|w\|^2 + \frac{1}{2}\frac{1}{\hat{\sigma}_e^2}\sum_{i=1}^{N} e_i^2, \tag{10}$$

where $y_i - \langle w, \phi(x_i; \Theta)\rangle = e_i,\ \forall i = 1, \ldots, N$. In the worst case, all data are noise, and therefore $\hat{\sigma}_e^2 = \sigma_y^2$ and we can set a heuristic minimum γ value as $\gamma_{\min} = 1$. A range over which to search for γ could be $[\gamma_{\min}, \gamma_h]$, but we can use values near γ_h if we rely on NNE. Jones, Evans, and Kemp (2007) demonstrated that the presence of noise in the input vectors means that, instead of approximating the variance of the noise, NNE gives a measure of the variance of the effective noise in the output. The variance of the effective noise is larger than the variance of the noise itself, but a better approximation cannot be performed.

Since in the case of time series prediction we have to generate input/output vectors from the very time series, the input vectors will always have noise. It is therefore advisable to search for the γ value in a range centered on γ_h.

4.2. Training data-based guess of the σ parameter of the Gaussian kernel

We consider the most common case of the LS-SVM model for regression, namely the LS-SVM model with a *Gaussian* kernel, the equation of which is:

$$k(x, x'; \sigma) = \exp\left(-\frac{1}{\sigma^2}\|x - x'\|^2\right). \tag{11}$$

The σ value is related to the distance between training points and the smoothness of the interpolation of the model (Rojas et al., 2000b). As a general rule, the higher the σ, the smoother the interpolation between two consecutive points. A reasonable range search for σ is $[\sigma_{\min}, \sigma_{\max}]$, where $\sigma_{\min}$ is the minimum distance (non-zero) between 2 training points and $\sigma_{\max}$ is the maximum distance between 2 training points.

4.3. Parameter optimization of the LS-SVM model

In the previous sub-sections some heuristic ranges were proposed for selecting initial guesses for both γ and the Gaussian kernel parameter σ. These values are just that, initial guesses, and we cannot expect them to be the optimal values with respect to our chosen criteria, the cross-validation error. Thus, a further optimization process is necessary. To compute the l-fold cross-validation error of a model, the available I/O data describing the system to be approximated are split into l equally-sized subsets. Then each of the sets in turn is used as a validation set while the rest of the subsets are used as the training set for generating a model. The average validation error is known as the l-fold cross-validation error of the model.

The main drawback of cross-validation is the computational complexity it entails, since l models must be trained. A reduced cost method for the evaluation of the l-fold cross-validation error for LS-SVMs is presented by An et al. (2007).

In order to perform a better optimization, the partial derivatives of the l-fold cross-validation error of the model with respect to a given parameter p, which can be either the hyperparameter γ or a kernel parameter, have to be provided. These were obtained as is shown in the Appendix. In the case of the kernel parameters, this evaluation will also depend on the partial derivatives of the kernel function with respect to its parameters, expressions which must be provided for each individual kernel.

In any case, a conjugate gradient (CG) or Levenberg-Marquardt (LM) scheme can be used with the above equations to optimize both γ and the kernel parameters in a LS-SVM model. Since this is normally

not a linear problem, the convergence to the global optimum is not guaranteed; hence the importance of good initial guesses for the parameters.

5. Data sets and experimental setup

In order to make a fair and thorough comparative study, a number of time series were selected to cover a range of possible practical situations:

- non-linear deterministic discrete and continuous time series;
- stochastic linear and non-linear time series;
- real time series; and
- time series from competitions.

A non-linear deterministic (i.e. non-stochastic) series must present some form of chaotic behaviour in order to be a non-trivial approximation problem (e.g., it must neither diverge nor converge to a value or cycle). There are several examples of deterministic time series available in the literature, and four were selected for the comparison:

- *Hénon*: the Hénon map is one of the most studied dynamic systems. The canonical Hénon map takes points in the plane following Hénon (1976):

$$\begin{aligned} x_{n+1} &= y_n + 1 - 1.4x_n^2 \\ y_{n+1} &= 0.3x_n. \end{aligned} \tag{12}$$

- *Logistic*: the Logistic map is a demographic model that was popularized by May (1976) as an example of a simple non-linear system that exhibits complex, chaotic behaviour. It is drawn from:

$$y(t) = 4y_{t-1}(1 - y_{t-1}). \tag{13}$$

- *Mackey-Glass*: the Mackey-Glass time series is approximated from the differential equation (14) (see Mackey & Glass, 1977). It is a widely used benchmark for the generalization abilities of time series prediction methods. The series is continuous and is obtained by integrating Eq. (14) with a numerical integration method such as the fourth order Runge-Kutta method. The data for this time series were obtained from the `mgdata.dat` file included in the *Fuzzy Logic Toolbox*[1] from the Matlab software.

[1] See http://www.mathworks.com/products/fuzzylogic/?BB=1.

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-17)} - 0.1x(t) \tag{14}$$

The stochastic linear and non-linear time series for the study consist of deterministic series with some associated noise level:

- *AR(4)*: a 4th order autoregressive series[2]; and
- *STAR*: A smooth transition autoregressive model, consisting of two AR model parts linked by a transition function. The non-linear time series was generated from Eq. (15), as was shown by Berardi and Zhang (2003), with a noise variance of $\sigma^2 = 5.0\mathrm{e}{-2}$.

$$\begin{aligned} y(t) = {} & 0.3y(t-1) + 0.6y(t-2) \\ & + \frac{0.1 - 0.9y(t-1) + 0.8y(t-2)}{1 + \mathrm{e}^{-10y(t-1)}} \\ & + N(0, \sigma^2) \end{aligned} \tag{15}$$

Real time series are difficult to model, as they normally present a lot of noise and non-linear properties. The series considered here are:

1. *Sunspots*: number of sunspots by month from 1700 to 2005[3]; this chaotic time series is highly studied, but has various local behaviours, noise and even unpredictable zones;
2. *London*: monthly mean temperatures in London from January 1659 to October 2007[4]; and
3. *Electric*: daily electricity load data in California[5] (the original load data are sampled every hour).

Finally, time series from competitions are often used as benchmarks in the literature. A pair of series from the Santa Fe Competition (Weigend & Gershenfeld, 1994) were added to the experiments: *Laser* generated data and a *Computer* generated series.

As a pre-processing stage, the trend was eliminated when necessary and the data were normalized to have mean 0 and variance 1. In order to use LS-SVM for time series modelling, the input/output (I/O)

[2] Provided at http://www.robjhyndman.com/TSDL/misc/simar4.dat.

[3] Available at ftp://ftp.ngdc.noaa.gov/STP/SOLAR_DATA/SUNSPOT_NUMBERS/MONTHLY.PLT.

[4] Available at http://hadobs.metoffice.com/hadcet/cetml1659on.dat.

[5] Available at http://www.ucei.berkeley.edu/CSEM/datamine/ISOdata/.

Table 1
Details of the time series.

Data	Model	F.S.	Trend	Size
Hénon	$y_{t+1} = F(y_t, y_{t-1})$	No	No	500
Logistic	$y_{t+1} = F(y_t)$	No	No	500
Mackey-Glass	$y_{t+1} = F(y_{t-5}, y_{t-11}, y_{t-17}, y_{t-23})$	No	No	500
AR(4)	$y_{t+1} = F(y_t, y_{t-1}, y_{t-2}, y_{t-3})$	No	No	500
STAR	$y_{t+1} = F(y_t, y_{t-1})$	No	No	500
Sunspots	$y_{t+1} = F(y_t, \ldots, y_{t-12})$	Yes	No	306
London	$y_{t+1} = F(y_t, \ldots, y_{t-13})$	Yes	No	500
Electric	$y_{t+1} = F(y_t, \ldots, y_{t-16})$	Yes	Yes	500
Computer	$y_{t+1} = F(y_t, \ldots, y_{t-20})$	Yes	No	500
Laser	$y_{t+1} = F(y_t, \ldots, y_{t-6})$	Yes	No	500

Notes. F.S.: feature selection: *No* means that the model is obtained from literature and *Yes* that a feature selection based on the gamma test is performed; Trend: trend elimination by differentiation: *Yes* means that the series had to be differentiated to de-trend it and *No* that this was not necessary; Size: number of data points used in the experiments.

vectors required by the methods were created, so that a number of regressors have been specified for the prediction horizon ($t + 1$) of interest for each dataset. For time series with known generating processes, the optimal regressors have been given directly to create the I/O vectors (*Hénon, Logistic, AR(4), STAR*). In the case of the Mackey-Glass time series, we have chosen the regressors which are most commonly found in the literature (Herrera, Pomares, Rojas, Valenzuela, & Prieto, 2005; Rojas et al., 2002). For the remaining series, a feature selection procedure was needed. The method used, based on the work of Sorjamaa, Hao, Reyhani, Ji, and Lendasse (2007), consists of the evaluation of the gamma test for all I/O vectors created with regressors from $t - 1$ to $t - 20$, and the selection of the one with the best value. A more careful study of each series could reveal other useful properties which could improve the model, but feature selection is not our objective in these experiments. In Table 1 we summarize the actions performed on every data set collected from each time series. In each case, 500 values of the series were used as the training data, except for the Sunspots series, which has fewer values. In the case of the Hénon, Logistic and Mackey-Glass series, a Gaussian additive error of variance 0.01 was added to the data to avoid a value of 0 on the NNE.

6. Results

Using the data sets described in the previous section, some experiments are performed in order to test our proposed heuristics for setting the γ hyperparameter and the Gaussian kernel σ parameter in the unbiased LS-SVM for regression.

For each dataset, and for a fixed value of σ in the mid-point of the range recommended, the 10-fold cross-validation error of the LS-SVM model was computed for a very wide range of γ values. The resulting graphs of the logarithm of the cross-validation error to the logarithm of γ are shown in Figs. 1–5.

In the case of the Henon, Logistic and STAR data sets (Figs. 1 and 3), there is a large range of nearly optimal values of γ, and thus the value provided by our heuristic, marked by a dashed line, is not as relevant as in the other cases for being the starting point for an optimization process or even a good final solution for the regularization factor. In the case of the Mackey-Glass, Sunspots, London Temperature, Electric Load, Computer and Laser data sets (Figs. 2–5), there is a clear optimal value, and the proposed heuristic value leads to a point very close to it. The results for AR(4) (Fig. 2) are the worst of the experiments; nevertheless, the value reached is adequate as the starting point for an optimization process. As can be shown, γ_h is generally a reasonable starting point from which to search for the optimum γ value.

In order to make a deeper statistical analysis, a set of experiments was carried out to evaluate the heuristic initial values proposed in Section 4. This is done by drawing 500 pairs of values (γ, σ) from the following increasing ranges of bi-dimensional distributions that will be called *ranges* $R_0, \ldots, R_5$:

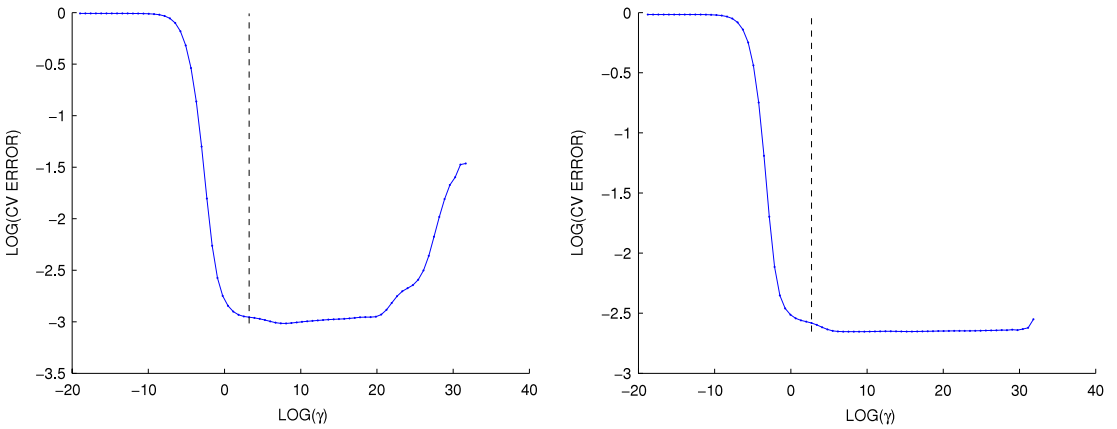

Fig. 1. LS-SVM 10-fold cross validation error versus γ (γ_h is marked with a dashed line) for a fixed σ for Henon (left) and Logistic (right).

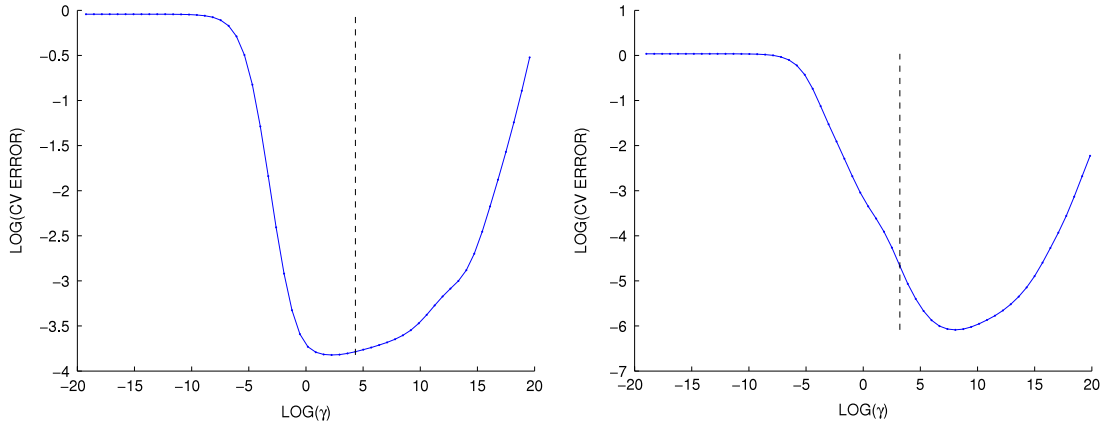

Fig. 2. LS-SVM 10-fold cross validation error versus γ (γ_h is marked with a dashed line) for a fixed σ for Mackey-Glass (left) and AR(4) (right).

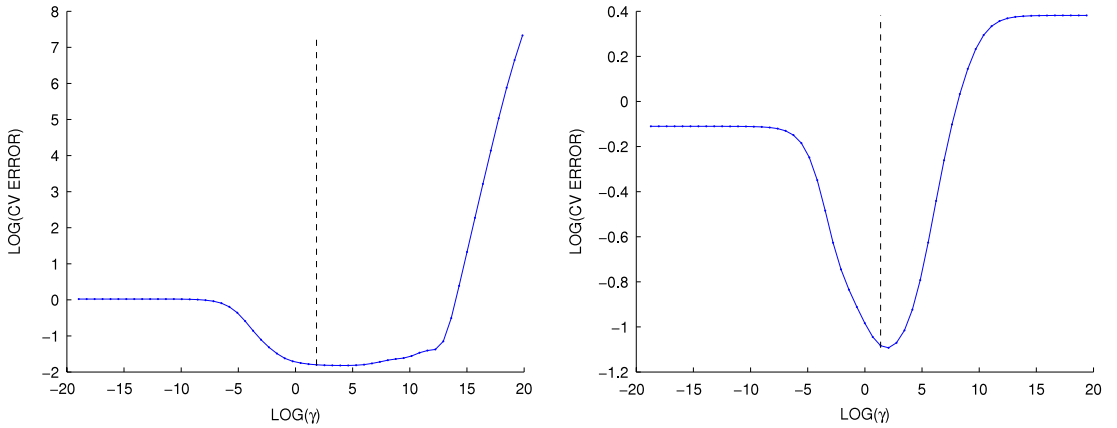

Fig. 3. LS-SVM 10-fold cross validation error versus γ (γ_h is marked with a dashed line) for a fixed σ for STAR (left) and Sunspots (right).

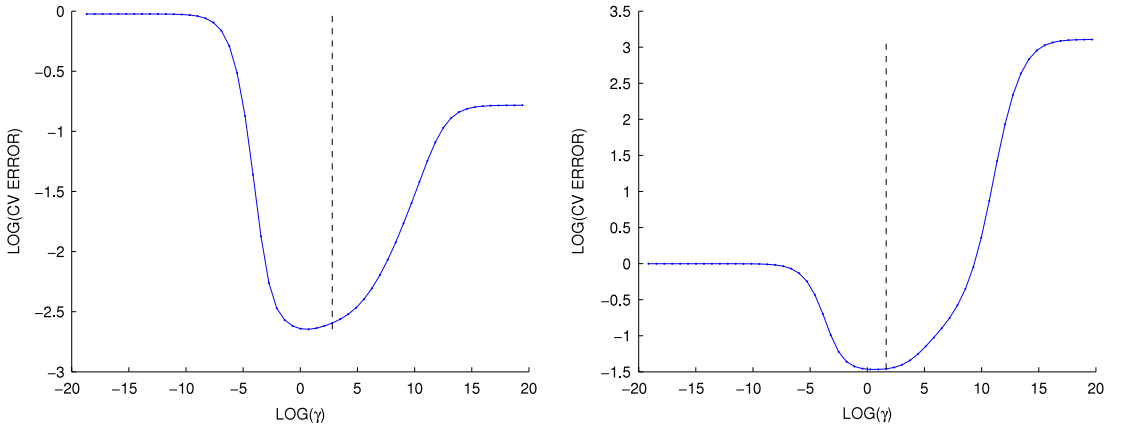

Fig. 4. LS-SVM 10-fold cross validation error versus γ (γ_h is marked with a dashed line) for a fixed σ for London (left) and Electric Load (right).

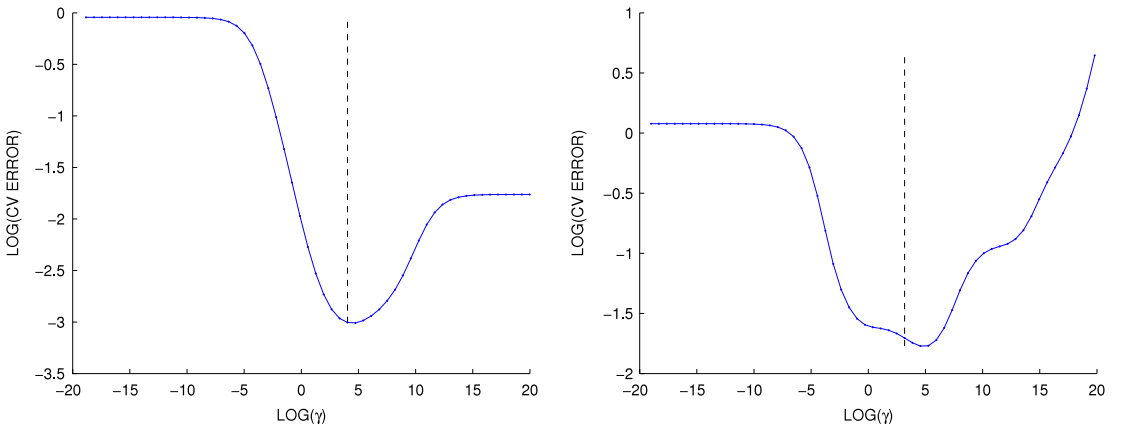

Fig. 5. LS-SVM 10-fold cross validation error versus γ (γ_h is marked with a dashed line) for a fixed σ for Computer (left) and Laser (right).

R_0: $\gamma \sim \gamma_h + N(0, (\gamma_h - \gamma_{\min})/100)$ and $\sigma \sim U(\sigma_{\min}, \sigma_{\max})$
R_1: $\gamma \sim U(\gamma_{\min}, \gamma_h)$ and $\sigma \sim U(1/2 \cdot \sigma_{\min}, 2 \cdot \sigma_{\max})$
R_2: $\gamma \sim U(\gamma_{\min}, \gamma_h)$ and $\sigma \sim U(1/4 \cdot \sigma_{\min}, 4 \cdot \sigma_{\max})$
R_3: $\gamma \sim U(\gamma_{\min}, \gamma_h)$ and $\sigma \sim U(1/6 \cdot \sigma_{\min}, 6 \cdot \sigma_{\max})$
R_4: $\gamma \sim U(\gamma_{\min}, \gamma_h)$ and $\sigma \sim U(1/8 \cdot \sigma_{\min}, 8 \cdot \sigma_{\max})$
R_5: $\gamma \sim U(\gamma_{\min}, \gamma_h)$ and $\sigma \sim U(1/10 \cdot \sigma_{\min}, 10 \cdot \sigma_{\max})$,

where N stands for the normal distribution and U for the uniform distribution. The 10-fold cross-validation error of the LS-SVM model that uses the parameter values picked out from each range was computed using Eq. (16). The first *range* is that recommended in our heuristic for getting good initial parameter values; while the next 5 are successively larger search ranges for each parameter.

To evaluate the performance of the heuristics for each dataset, first the pseudo-optimal values of γ and σ with respect to the 10-fold cross-validation error of the LS-SVM model were computed by multiple execution of a CG procedure (using Eqs. (16)–(23)) using all of the initial values picked out as starting points.

For each dataset, the distances from these computed optimum values of the parameters to those obtained from every range are summarized in Table 2. In addition, in Figs. 6–15 results for the 10-fold cross-validation error are represented in boxplots, ordered by ranges. This representation draws a box delimited

Table 2
Average distance to the optimum found and standard deviation by range.

DATASET		
Henon	R_0: 1.68e+00 ± 3.83e−01	R_1: 2.00e+00 ± 6.70e−01
	R_2: 2.66e+00 ± 6.98e−01	R_3: 2.99e+00 ± 7.67e−01
	R_4: 3.23e+00 ± 8.95e−01	R_5: 3.50e+00 ± 8.55e−01
Logistic	R_0: 5.28e+00 ± 2.40e−01	R_1: 6.05e+00 ± 6.40e−01
	R_2: 6.13e+00 ± 6.45e−01	R_3: 6.27e+00 ± 6.76e−01
	R_4: 6.29e+00 ± 6.34e−01	R_5: 6.31e+00 ± 6.18e−01
Mackey-Glass	R_0: 1.68e+00 ± 1.88e−01	R_1: 1.43e+00 ± 4.73e−01
	R_2: 1.89e+00 ± 5.48e−01	R_3: 2.18e+00 ± 6.58e−01
	R_4: 2.42e+00 ± 7.08e−01	R_5: 2.62e+00 ± 7.39e−01
AR(4)	R_0: 2.85e+01 ± 2.51e−01	R_1: 2.91e+01 ± 7.01e−01
	R_2: 2.89e+01 ± 7.74e−01	R_3: 2.88e+01 ± 7.33e−01
	R_4: 2.87e+01 ± 7.68e−01	R_5: 2.86e+01 ± 7.31e−01
STAR	R_0: 4.03e+00 ± 2.90e−01	R_1: 4.65e+00 ± 5.84e−01
	R_2: 4.67e+00 ± 5.10e−01	R_3: 4.75e+00 ± 5.16e−01
	R_4: 4.78e+00 ± 4.99e−01	R_5: 4.86e+00 ± 5.25e−01
Sunspots	R_0: 1.95e+00 ± 2.28e−01	R_1: 2.35e+00 ± 3.52e−01
	R_2: 2.46e+00 ± 3.29e−01	R_3: 2.56e+00 ± 3.49e−01
	R_4: 2.71e+00 ± 3.80e−01	R_5: 2.78e+00 ± 4.29e−01
London	R_0: 2.06e+00 ± 2.43e−01	R_1: 1.46e+00 ± 5.47e−01
	R_2: 1.64e+00 ± 5.00e−01	R_3: 1.87e+00 ± 5.64e−01
	R_4: 2.11e+00 ± 5.51e−01	R_5: 2.22e+00 ± 5.89e−01
Electric	R_0: 8.83e−01± 5.14e−01	R_1: 8.65e−01± 5.28e−01
	R_2: 1.11e+00 ± 5.25e−01	R_3: 1.39e+00 ± 5.61e−01
	R_4: 1.58e+00 ± 6.02e−01	R_5: 1.78e+00 ± 6.21e−01
Computer	R_0: 8.01e+00 ± 2.19e−01	R_1: 8.54e+00 ± 5.63e−01
	R_2: 8.50e+00 ± 5.88e−01	R_3: 8.43e+00 ± 5.94e−01
	R_4: 8.43e+00 ± 5.72e−01	R_5: 8.43e+00 ± 5.68e−01
Laser	R_0: 1.71e+01 ± 1.56e−01	R_1: 1.79e+01 ± 7.21e−01
	R_2: 1.79e+01 ± 7.43e−01	R_3: 1.79e+01 ± 7.40e−01
	R_4: 1.78e+01 ± 7.43e−01	R_5: 1.78e+01 ± 7.95e−01

by the lower, median and upper quartile whiskers whose extremes are 1.5 times the interquartile range from the ends of the box. Outliers are denoted by + signs.

The results obtained for the Hénon, Logistic, Mackey-Glass, STAR, Sunspots, and Electric Load time series in Figs. 6–8, 10, 11 and 13, respectively, confirm the validity of the heuristics. They show that the average value of the cross validation error grows with the size of the range from which the parameters are taken, despite the relatively large number of outliers. The best results are obtained for the first range, the one with the proposed heuristic values.

The results for AR(4) and London Temperature, in Figs. 9 and 12 respectively, show that the heuristic

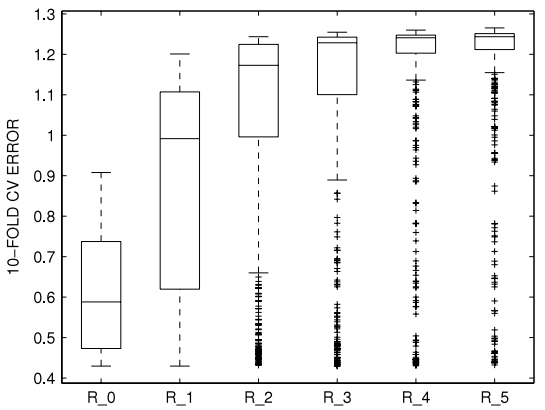

Fig. 6. Boxplot 10-fold cross validation error by range for Hénon.

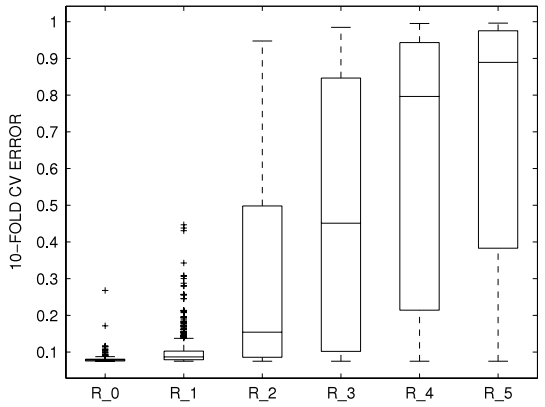

Fig. 7. Boxplot 10-fold cross validation error by range for Logistic.

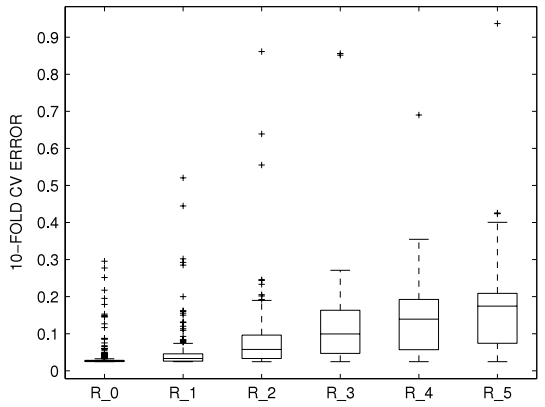

Fig. 8. Boxplot 10-fold cross validation error by range for Mackey-Glass.

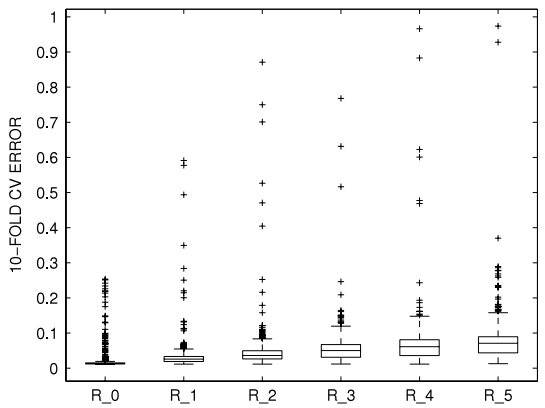

Fig. 9. Boxplot 10-fold cross validation error by range for AR(4).

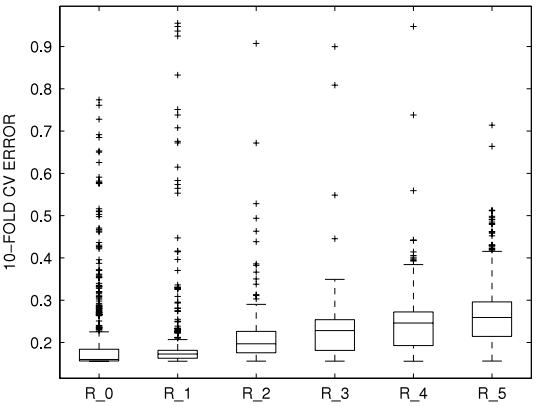

Fig. 10. Boxplot 10-fold cross validation error by range for STAR.

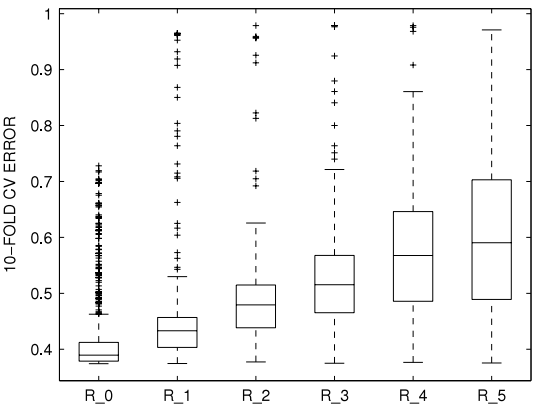

Fig. 11. Boxplot 10-fold cross validation error by range for Sunspots.

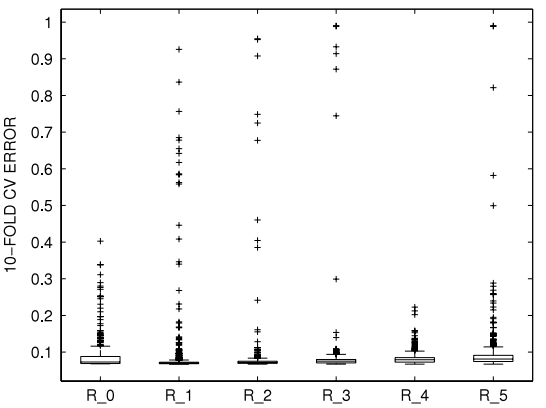

Fig. 12. Boxplot 10-fold cross validation error by range for London.

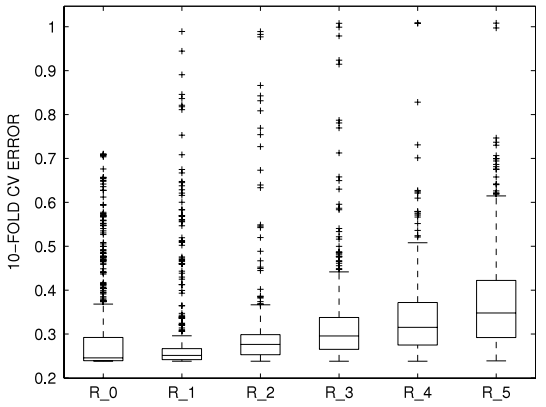

Fig. 13. Boxplot 10-fold cross validation error by range for Electric Load.

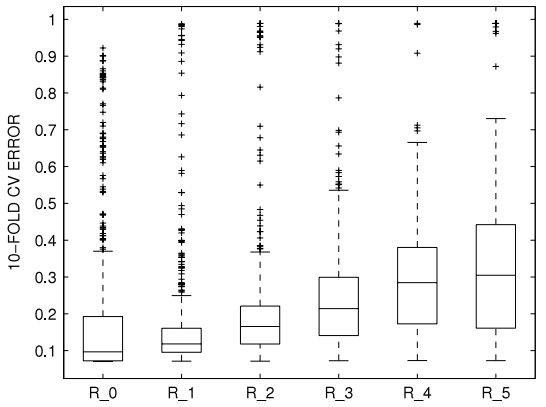

Fig. 14. Boxplot 10-fold cross validation error by range for Computer.

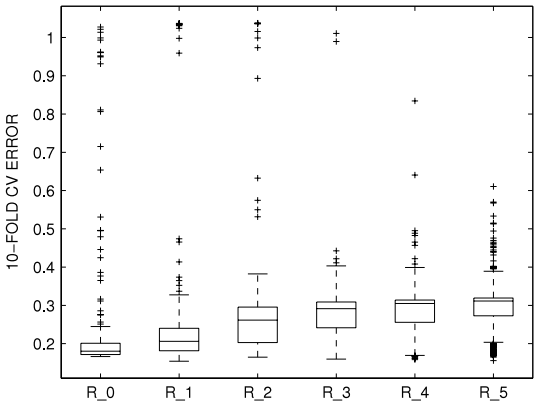

Fig. 15. Boxplot 10-fold cross validation error by range for Laser.

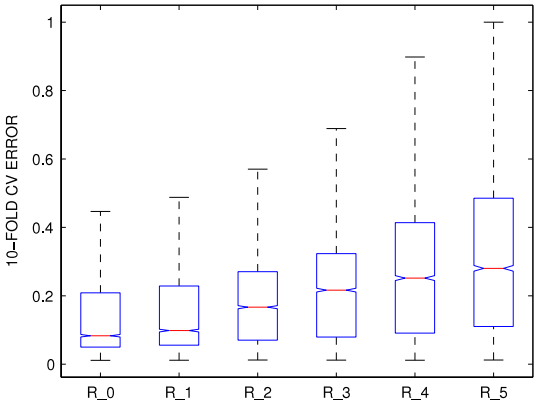

Fig. 16. Boxplot 10-fold cross validation error by range for all series.

values are not so critical in these cases. It is worth pointing out here that the influence of the range on the cross validation error is less in these cases than with other series, but still present. The explanation for these results is related to the fact that both time series are highly linear, and it is possible that the σ parameter may not be very relevant in this case.

For the two highly non-linear series from the Santa Fe time series competition (Computer in Fig. 14 and Laser in Fig. 15), the results again show that the first range is the best. However, the rest of the ranges have little influence. The results for the average cross validation error have a large number of outliers, but still confirm the validity of the heuristics.

From the aforementioned figures and Table 2, we have seen that the parameter values generated in the first range are closer to the computed optimum on average, and the 10-fold cross-validation (CV) errors of the model are also smaller. In order to make a statistical comparison of all of the results (distance and CV errors) for all series, they were normalized by dataset by the mean of the worst case (the last range) and analyzed using the Kruskal-Wallis test. We found that, with a very high probability ($p \ll 0.001$), the ranges influence the results, and the proposed range R_0 is the best solution to the problem (the corresponding boxplots can be seen in Fig. 16).

7. Conclusions

In this paper, we have presented a heuristic initialization value for the regularization factor γ in LS-SVM based on the non-parametric noise estimation of the data, as well as a reasonable range on

which to search for the value of the kernel parameter σ for the particular case of LS-SVM with a Gaussian kernel. We have also provided the expressions needed to perform a local search from the proposed initial values in order to minimize the cross-validation error of the model. The results showed that, for a very wide range of time series data sets, the values obtained starting from our given initial values are statistically better than those obtained from other larger ranges of search, both in terms of the distance to the *global* optimum and in terms of the cross-validation error. These results are significant because they provide a guide to the initialisation and orientation of the search for the parameter values for this kind of model, which is one of the most used in practice for time series modelling and prediction.

Acknowledgements

This work has been supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768.

Appendix. Evaluation of the l-fold cross-validation error of LS-SVM

Given an unbiased LS-SVM model built upon a set of function samples $\{(x_1, y_1), \ldots, (x_N, y_N)\} \subset \mathbb{R}^d \times \mathbb{R}$, where N is the number of samples and d is the dimensionality of the input data for a given kernel function with fixed values of the kernel parameters and γ; a reduced cost method for the evaluation of the l-fold cross-validation error for LS-SVMs consists of computing the following expression (An et al., 2007):

$$MSE_{l\text{-fold}} = \frac{1}{N}\sum_{m=1}^{l}\sum_{j=1}^{|\beta^{(m)}|}\beta_j^{(m)2}, \tag{16}$$

where

$$\beta^{(m)} = C_{mm}^{-1}\alpha^{(m)}$$

$$C = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1l} \\ C_{12}^T & C_{22} & \cdots & C_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ C_{l2}^T & C_{2l}^T & \cdots & C_{ll} \end{bmatrix} = K_\gamma^{-1}$$

$$K_\gamma = \Omega + I/\gamma$$

$$\alpha = \left[\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(l)}\right]^T,$$

and Ω, I, γ and α were defined in Eq. (2). It should be noted that, for N data, C is a matrix of dimension $N \times N$, each of its sub-matrices C_{mm} is a matrix of dimension $N/l \times N/l$, and each $\beta^{(m)}$ and $\alpha^{(m)}$ is a vector of dimension N/l.

In order to find a local minimum of $MSE_{l\text{-fold}}$, we must first compute the partial derivatives of Eq. (16) with respect to a given parameter p, which can be either the hyperparameter γ or a kernel parameter:

$$\frac{\partial MSE_{l\text{-fold}}}{\partial p} = \frac{2}{N}\sum_{m=1}^{l}\sum_{j=1}^{|\beta^{(m)}|}\beta_j^{(m)}\left[\frac{\partial \beta^{(m)}}{\partial p}\right]_j \tag{17}$$

$$\frac{\partial \beta^{(m)}}{\partial p} = \frac{\partial C_{mm}^{-1}}{\partial p}\alpha^{(m)} + C_{mm}^{-1}\frac{\partial \alpha^{(m)}}{\partial p} \tag{18}$$

$$\frac{\partial C_{mm}^{-1}}{\partial p} = -C_{mm}^{-1}\frac{\partial C_{mm}}{\partial p}C_{mm}^{-1} \tag{19}$$

$$\frac{\partial C}{\partial p} = \begin{bmatrix} \frac{\partial C_{11}}{\partial p} & \frac{\partial C_{12}}{\partial p} & \cdots & \frac{\partial C_{1l}}{\partial p} \\ \frac{\partial C_{12}^T}{\partial p} & \frac{\partial C_{22}}{\partial p} & \cdots & \frac{\partial C_{2l}}{\partial p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_{l2}^T}{\partial p} & \frac{\partial C_{2l}^T}{\partial p} & \cdots & \frac{\partial C_{ll}}{\partial p} \end{bmatrix} \tag{20}$$

$$\frac{\partial C}{\partial p} = \frac{\partial K_\gamma^{-1}}{\partial p} \tag{21}$$

$$\frac{\partial \alpha}{\partial p} = \frac{\partial K_\gamma^{-1}}{\partial p}y \tag{22}$$

$$\frac{\partial K_\gamma^{-1}}{\partial p} = -K_\gamma^{-1}\frac{\partial K_\gamma}{\partial p}K_\gamma^{-1}. \tag{23}$$

The evaluation of the partial derivatives therefore implies the inversion of the matrix K_γ (an operation which has a computational complexity of $O(N^3)$ with exact methods) and the evaluation of the partial derivatives of K_γ with respect to p. In the case of $p = \gamma$, $\partial K_\gamma/\partial p = -I/\gamma^2$. In the case of the kernel parameters, this evaluation will depend on the partial derivatives of the kernel function with respect to its parameters, expressions which must be provided for each particular kernel. In the particular case of the

Gaussian kernel:

$$\frac{\partial k(x, x'; \sigma)}{\partial \sigma} = 2\frac{\|x - x'\|^2}{\sigma^3} \times \exp\left(-\frac{1}{\sigma^2}\|x - x'\|^2\right), \quad (24)$$

where $\sigma > 0$ is the only kernel parameter. It is common to perform the optimization with respect to $\log(\sigma)$ instead of σ:

$$\frac{\partial k(x, x'; \log(\sigma))}{\partial \log(\sigma)} = 2\frac{\|x - x'\|^2}{\sigma^2} \times \exp\left(-\frac{1}{\sigma^2}\|x - x'\|^2\right). \quad (25)$$

References

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? a review and evaluation. *Journal of Forecasting*, *17*, 481–495.

Adya, M., Collopy, F., Kennedy, M., & Armstrong, J. S. (2001). Identifying features of time series for rule-based forecasting. *International Journal of Forecasting*, *17*(4), 143–157.

Alves da Silva, A. P., Ferreira, V. H., & Velasquez, R. M. (2008). Input space to neural network based load forecasters. *International Journal of Forecasting*, *24*(4), 616–629.

An, S., Liu, W., & Venkatesh, S. (2007). Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. *Pattern Recognition*, *40*(8), 2154–2162.

Balkin, S. D., & Ord, J. K. (2000). Automatic neural network modeling for univariate time series. *International Journal of Forecasting*, *16*(4), 509–515.

Bellman, R. (1966). Dynamic programming, system identification, and suboptimization. *SIAM Journal on Control and Optimization*, *4*, 1–5.

Berardi, V., & Zhang, G. (2003). An empirical investigation of bias and variance in time series forecasting: modeling considerations and error evaluation. *IEEE Transactions on Neural Networks*, *14*(3), 668–679.

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis, forecasting and control*. San Francisco, CA: Holden Day.

Cawley, G. C., & Talbot, N. L. C. (2002). Efficient formation of a basis in a kernel induced feature space. In *Proceedings of ESANN* (pp. 1–6).

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126.

Coulibaly, P., Anctil, F., & Bobee, B. (2001). Multivariate reservoir inflow forecasting using temporal neural networks. *Journal of Hydrologic Engineering*, *6*, 367–376.

de Kruif, B., & de Vries, T. (2003). Pruning error minimization in least squares support vector machines. *IEEE Transactions on Neural Networks*, *14*(3), 696–702.

Hénon, M. (1976). A two-dimensional mapping with a strange attractor. *Communications in Mathematical Physics*, *50*, 69–77.

Herrera, L., Pomares, H., Rojas, I., Valenzuela, O., & Prieto, A. (2005). TaSe, a Taylor series-based fuzzy system model that combines interpretability and accuracy. *Fuzzy Sets and Systems*, *153*, 403–427.

Hong, W.-C., & Pai, P.-F. (2006). Predicting engine reliability by support vector machines. *The International Journal of Advanced Manufacturing Technology*, *28*, 154–161.

Hsieh, W. W., & Tang, B. (1998). Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*, *79*(9), 1855–1870.

Jang, J. S. R. (1993). ANFIS: adaptive network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, *23*, 665–685.

Jones, A. J., Evans, D., & Kemp, S. E. (2007). A note on the Gamma test analysis of noisy input/output data and noisy time series. *Physica D — Nonlinear Phenomena*, *229*, 1–8.

Jursa, R., & Rohrig, K. (2008). Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. *International Journal of Forecasting*, *24*(4), 694–709.

Kantz, H., & Schreider, T. (1997). *Nonlinear time series analysis*. Cambridge University Press.

Kim, D., & Kim, C. (1997). Forecasting time series with genetic fuzzy predictor ensemble. *IEEE Transactions on Fuzzy Systems*, *5*(4), 523–535.

Lee, S. H., & Kim, I. (1994). Time series analysis using fuzzy learning. In W. Kim, & S. Y. Lee (Eds.), *Proceedings of the international conference on neural information processing*, vol. 6 (pp. 1577–1582).

Lendasse, A., Corona, F., Hao, J., Reyhani, N., & Verleysen, M. (2006). Determination of the Mahalanobis matrix using nonparametric noise estimations. In *Proceedings of ESANN* (pp. 227–232).

Liitiäinen, E., Lendasse, A., & Corona, F. (2007). Non-parametric residual variance estimation in supervised learning. In *Proceedings of IWANN* (pp. 63–71).

Liu, H., Liu, D., Zheng, G., & Liang, Y. (2004). Research on natural gas load forecasting based on support vector regression. In *Proceedings of the 5th world congress on intelligent control and automation* (pp. 3591–3595).

Mackey, M. C., & Glass, L. (1977). Oscillation and chaos in physiological control systems. *Science*, *197*(4300), 287–289.

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, *261*, 459–467.

Mélard, G., & Pasteels, J. M. (2000). Automatic ARIMA modeling including interventions, using time series expert software. *International Journal of Forecasting*, *16*(4), 497–508.

Misra, D., Oommen, T., Agarwal, A., Mishra, S. K., & Thompson, A. M. (2009). Application and analysis of support vector machine based simulation for runoff and sediment yield. *Biosystems Engineering*, *103*, 527–535.

Müller, K.-R., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., & Vapnik, V. (1999). it Using support vector machines for time series prediction. In B. Scholkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—support vector learning* (pp. 243–254). Cambridge, MA: MIT Press.

Rojas, I., González, J., Cañas, A., Díaz, A. F., Rojas, F. J., & Rodriguez, M. (2000a). Short-term prediction of chaotic time series by using RBF network with regression weights. *International Journal of Neural Systems*, *10*(5), 353–364.

Rojas, I., Pomares, H., Bernier, J., Ortega, J., Pino, B., Pelayo, F., et al. (2002). Time series analysis using normalized PG-RBF network with regression weights. *Neurocomputing*, *42*, 267–285.

Rojas, I., Pomares, H., González, J., Bernier, J., Ros, E., Pelayo, F., et al. (2000b). Analysis of the functional block involved in the design of radial basis function networks. *Neural Processing Letters*, *12*, 1–17.

Rojas, I., Pomares, H., González, J., Ros, E., Salmerón, M., Ortega, J., et al. (2000c). A new radial basis function networks structure: application to time series prediction. In S. I. Amari, C. L. Giles, M. Gori, & V. Piuri (Eds.), *Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks*: *Vol. IV* (pp. 449–454). Como, Italy: IEEE Computer Society.

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: a survey. *Computational Intelligence Magazine, IEEE*, *4*(2), 24–38.

Scholkopf, B., & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press.

Shiblee, M., Kalra, P. K., & Chandra, B. (2009). Time series prediction with multilayer perceptron (MLP): a new generalized error based approach. *Advances in Neuro-Information Processing*, 37–44.

Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (2007). Methodology for long-term prediction of time series. *Neurocomputing*, *70*, 2861–2869.

Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific Publishing.

Tay, F., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega: The International Journal of Management Science*, *29*(4), 309–317.

Teräsvirta, T., Medeiros, M. C., & Rech, G. (2006). Building neural network models for time series: a statistical approach. *Journal of Forecasting*, *25*(1), 49–75.

Thiessen, U., & Van Brakel, R. (2003). Using support vector machines for time series prediction. *Chemometrics and Intelligent Laboratory Systems*, *69*, 35–49.

Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., Herrera, L. J., Guillen, A., et al. (2008). Hybridization of intelligent techniques and ARIMA models for time series prediction. *Fuzzy Sets and Systems*, *159*(7), 821–845.

Van Gestel, T., Suykens, J., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., et al. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks*, *12*(4), 809–821.

Wang, W., Chau, K., Cheng, C., & Qiu, L. (2009). A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. *Journal of Hydrology*, *374*, 294–306.

Weigend, A. S., & Gershenfeld, N. A. (Eds.) (1994). *Time series prediction: forecasting the future and understanding the past*. Reading, MA: Addison-Wesley.

Xu, R., & Bian, G. (2005). Discussion about nonlinear time series prediction using least squares support vector machine. *Communications in Theoretical Physics*, *43*, 1056–1060.

Xu, G., Tian, W., & Jin, Z. (2006). An AGO-SVM drift modelling method for a dynamically tuned gyroscope. *Measurement Science and Technology*, *17*(1), 161–167.

Ying, Z., & Keong, K. C. (2004). Fast leave-one-out evaluation and improvement on inference for LS-SVMs. In *ICPR 2004: Proceedings of the 17th international conference on pattern recognition*, *3* (pp. 494–497).

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 159–175.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting*, *14*(1), 35–62.

Zhou, J., Bai, T., Zhang, A., & Tian, J. (2008). Forecasting share price using wavelet transform and LS-SVM based on chaos theory. In *IEEE international conference on cybernetic intelligent systems* (pp. 300–304).