**Noname manuscript No.**
(will be inserted by the editor)

# Nonstationary regression with Support Vector Machines

**Guillermo L. Grinblat · Lucas C. Uzal · Pablo F. Verdes · Pablo M. Granitto**

Received: date / Accepted: date

**Abstract** In this work we introduce a method for data analysis in nonstationary environments: Time Adaptive Support Vector Regression (TA-SVR). The proposed approach extends a previous development which was limited to classification problems. Focusing our study on time series applications, we show that TA-SVR can improve the accuracy of several aspects of nonstationary data analysis, namely the tasks of modelling and prediction, input relevance estimation, and reconstruction of a hidden forcing profile.

**Keywords** Regression · Support Vector Machine · Nonstationary problems

## 1 Introduction

Nonstationary problems are those in which the data generating distribution changes (or drifts) over time. Recent years have shown an increasing interest in nonstationary data analysis [1,13], which is probably related to its many challenging and technologically critical applications such as spam detection [23], user's preference modelling [20], face detection in nonstationary enviroments [31] and mechanical systems monitoring [5], amongst others.

In this work we focus on nonstationary regression problems. A concrete example of such a system, described by Bartlett et al. [3], is a steel rolling mill, where the efficiency of its operation depends on how accurately the behavior of the rolling surfaces can be predicted. As in many industrial systems, an accurate physical model of the process (relating some measured input variables to the desired quantity) exists, but several unknown parameters may change over time. The change may be slow (as the rollers wear), or occasionally fast (as in a failure). In this paper we limit our analysis to the case of slowly changing scenarios.

Guillermo L. Grinblat · Lucas C. Uzal · Pablo F. Verdes · Pablo M. Granitto
CIFASIS – French Argentine International Center for Information and Systems Sciences, UAM (France) / UNR–CONICET (Argentina), Bv. 27 de Febrero 210 bis, Rosario, S2000EZP, Argentina
Tel.: +54-341-4237248
E-mail: {grinblat, uzal, verdes, granitto}@cifasis-conicet.gov.ar

Time series applications probably represent the most studied type of problem in this area, because most real-world time series have some degree of nonstationarity. This is generally due to external perturbations of the observed system, but in some cases natural dynamics are complex enough to comprise multiple time scales, so that for short observational periods the largest scales act simply as external perturbations to the fastest modes [35]. Applications in this area range from monitoring mechanical signals [8] to ecosystem modelling [14] or financial time series prediction [22, 24]. The method described in this work is general in nature and can be applied to any kind of nonstationary regression problem —not only time series modelling. However, taking into account the prevalence of the latter kind of problem in the literature and the fact that chaotic time series represent one of the most difficult type of regression problems, in this paper we focus on several nonstationary chaotic time series cases.

Specific methods have been developed for nonstationary time series analysis [9], including the proper characterisation of nonstationarity [26], caused either by slow continuous perturbations (usually called driving forces) [35] or by abrupt discrete changes in the dynamics [9]. Several methods have been introduced for the modelling and prediction of nonstationary time series, in particular for the case of systems that change slowly with time. Stark et al. [29] explicitly incorporated the time variable $t$ into the description of the system in order to encompass time-dependent dynamics. Casdagli [9] proposed the use of an extra input parameter, $\alpha$, to account for nonstationary effects, and assumed that $\alpha(t)$ was known. In a more recent work, Verdes et al. [34] proposed an improved algorithm to estimate $\alpha(t)$, the driving force of the nonstationary system, simultaneously with the modelling of the time series using a particular neural network model, yielding a remarkable improvement in modelling performance in comparison to other strategies.

The extension of the Support Vector Machine (SVM) [11] to regression problems, usually called Support Vector Regression (SVR) [12], is a powerful modelling method with a strong theoretical basis and great potential in practical regression applications. Many introductions to this method have been published (see for example [28]). New applications appear on a daily basis, including for example travel-time prediction, which is a critical step in advanced traveller information systems [37], automatic prediction of image quality [21] and financial forecasting [6]. However, only a reduced set of works have considered the use of SVR in nonstationary scenarios. In a series of papers, Tay and Cao analysed the application of SVR to nonstationary financial time series [33, 7]. To cope with nonstationarity they employed the simple and well known strategy of assigning an increasingly lower statistical weight to distant past samples, as done for example by Koychev [19] in the context of classification. Chang et al. [10] analysed the related problem of a dynamical system switching between a discrete number of modes.

In this work we propose a new SVR-based strategy for slowly varying regression problems. We extend the recently introduced Time Adaptive Support Vector Machine (TA-SVM) [15, 16, 27] to a regression framework, here called Time Adaptive Support Vector Regression (TA-SVR). The new method recourses to a series of coupled SVRs in order to learn in slowly changing environments. It is based on individual, flexible models which are fitted on short segments of the available data and are learned simultaneously (in a global manner) using a coupling term that forces neighbouring models to be similar to each other.

We evaluate TA-SVR on several nonstationary artificial chaotic time series examples and find that the proposed method is helpful on several aspects of nonstationary regression analysis. In particular, we show that TA-SVR is useful for: i) modelling and prediction of nonstationary time series, ii) relevance estimation through time of different model input variables, and iii) profile reconstruction of a hidden driving force acting on the system. In all cases we compare the performance of the new method against competitive strategies selected from the recent literature.

The rest of the paper is organised as follows. In Section II we introduce TA-SVR. In Section III we evaluate the proposed approach on the three tasks enumerated in the preceding paragraph. Finally, in Section IV we draw some conclusions.

## 2 TA-SVR

In this section we extend the TA-SVM method to the regression domain by combining it with the original $\epsilon$-SVR strategy, which casts a regression problem as a classification one by means of an $\epsilon$-insensitive tube. To this end we will closely follow the procedure presented in Grinblat et al. [15].

We begin by assuming that we are given a time-ordered data set $\{(\mathbf{x}_i, y_i), i = 1, \dots n\}$, where $\mathbf{x}_i$ is a multivariate input, $y_i \in \mathbb{R}$, and the relationship between $\mathbf{x}$ and $y$ slowly changes in time, which is here parameterised by $i$. We divide the dataset into $m$ consecutive, disjoint time-windows $tw_\nu$ ($\nu = 1, \dots m, m \leq n$) and fit a sequence of $m$ (static) regression models, one for each time-window. If the $(\mathbf{x}, y)$ mapping changes slowly over time, the sequence of individual regressions should inherit this property. We therefore seek for a succesion of models with first-neighbour similarity. The optimal solution to this problem will be given by a trade-off between individual model optimality and neighbouring models similarity. Assuming that d is a distance measure in model space, the core idea of our method is to minimise a two-term cost function:

$$\frac{1}{m}\sum_{\mu=1}^{m} Err_\mu + \frac{\gamma}{m-1}\sum_{\mu=1}^{m-1} \mathrm{d}(f_\mu, f_{\mu+1}), \tag{1}$$

where the first term represents the average prediction error of the fitted regressions while the second one measures the mean distance d between neighbouring models. The free hyperparameter $\gamma$ controls the compromise between both terms, as is customarily done in regularised model fitting.

The proposed approach can be implemented with any model family over which an appropriate distance measure can be defined. In this work we use SVRs[1]. Therefore, we look for a sequence of $m$ pairs $(\mathbf{w}, b)$, each one defining a linear regression function $f_\mu$ such that $f_\mu(\mathbf{x}) = \mathbf{w}_\mu \mathbf{x} + b_\mu$.

Following the same strategy as in TA-SVM, we use a simple quadratic distance to measure similarity between these models:

$$\mathrm{d}(f_\mu, f_\nu) = ||\mathbf{w}_\mu - \mathbf{w}_\nu||^2 + (b_\mu - b_\nu)^2.$$

[1] Here we employ linear SVRs but, as usual, kernels can be used to produce non-linear predictors if needed. In fact, in all of our examples we use a Gaussian kernel.

Applying this measure to (1), we can rewrite the cost function for the full sequence of SVRs as:

$$\frac{1}{m}\sum_{\mu=1}^{m}||\mathbf{w}_\mu||^2 + C\sum_{i=1}^{n}(\xi_i+\xi_i^*) + \frac{\gamma}{m-1}\sum_{\mu=1}^{m-1}\mathrm{d}(f_\mu, f_{\mu+1}), \tag{2}$$

which is to be minimised subject to

$$\begin{aligned}\xi_i, \xi_i^* &\geq 0,\\ y_i - \mathbf{w}_{\mu(i)}\mathbf{x}_i - b_{\mu(i)} + \varepsilon + \xi_i &\geq 0,\\ \mathbf{w}_{\mu(i)}\mathbf{x}_i + b_{\mu(i)} - y_i + \varepsilon + \xi_i^* &\geq 0,\end{aligned}$$

where $i = 1, \ldots n$, and $\mu(i)$ indicates the data window that includes point $\mathbf{x}_i$. The first term in (2) corresponds to the well-known margin term in SVM [11]. The second term is also typical, corresponding to the particular error penalisation term for SVR [28]. Note that these terms evaluate a complete set of models, each one trained on a different time window. So far, the solution of this two-term problem is the same set of SVRs that can be obtained by fitting each model independently, if we used the same $C$ for all SVRs. The last term in (2) adds the new diversity penalisation, which couples the sequence by relating each model to its first neighbours. Small $\gamma$ values will tend to decouple the sequence of regressions, allowing for increased flexibility. Large $\gamma$ values, on the other hand, will yield a chain of similar SVRs.

Along the same lines of the TA-SVM derivation [15, Appendix A], it is easy to see that the problem in (2) can be reformulated in terms of its corresponding dual as:

$$\max_{\alpha,\alpha^*}[-\frac{1}{2}(\alpha-\alpha^*)^T R(\alpha-\alpha^*) - \varepsilon\sum_i(\alpha_i+\alpha_i^*) + \sum_i y_i(\alpha_i+\alpha_i^*)], \tag{3}$$

subject to

$$0 \leq \alpha_i, \alpha_i^* \leq C \;\text{ and }\; \sum(\alpha_i - \alpha_i^*) = 0,$$

where $\alpha_i, \alpha_i^*$ are Lagrange multipliers (with $\alpha_i\alpha_i^* = 0$) and $R$ is a matrix with Kernel properties. The solution to this maximisation problem is a coupled set of SVRs that vary in time, which we call time-adaptive support vector regression machine (TA-SVR).

Most of the discussion and properties of TA-SVM also hold for TA-SVR. The computational burden of TA-SVR is of the same order as plain SVM. Problem (3) is a conventional SVM optimisation problem, which can be solved with typical methods, e.g. squential minimum optimisation (SMO) [25]. In the present formulation we only considered the case of data items arriving at regular time intervals. The more general case of irregularly sampled data can be addressed with simple extensions, as discussed in Grinblat et al. [15]. Finally, note that the method is valid even for degenerate time-windows of only one point ($m = n$), because the coupling introduced by the penalisation term breaks the degeneracy of trying to fit a hyperplane to a single data point. However, for regularisation purposes it is advisable to use $m < n$.

## 3 Applications

### 3.1 Nonstationary modelling of chaotic time series

As a first application of TA-SVR we analyse the problem of modelling nonstationary time series. We say that a signal measured from a dynamical system is stationary if all transition probabilities from one state of the system to another are independent of time within the observation period, i.e. when estimated from the data. This requires the constancy of the system's internal parameters but also that events belonging to the dynamics are contained in the time series sufficiently frequently, so that transition probabilities can be inferred properly. In this work we will focus on the first case, formalising nonstationarity as time-varying system parameters. We do not consider the notion of weak stationarity, which can be found in the literature on linear time series analysis and only requires statistical quantities up to second order to be constant, because it is inadequate in a nonlinear setting.

In order to assess the performance of TA-SVR, we follow the discussion in Verdes et al. [34] and benchmark against three other nonstationary modelling approaches. As a base method we use the simple strategy of fitting SVRs to local subsets of the original record, which are assumed to be stationary. This method is usually known as the Sliding Window (SW) approach. In the second method, following Stark et al. [29], we explicitly incorporate $t$ (the current time) as an extra input variable to the model, thereby allowing it to learn directly the time-dependent dynamics. We call this method "SVR + t". The last method we implement is, to our knowledge, the best strategy in the literature, and consists of estimating the driving force acting on the system while using it as an input variable to the regression [32]. Here we don't estimate $\alpha$ simultaneously with the modelling as in Verdes et al. [34]. Instead, we begin by estimating $\alpha$ with a different method [35] described in Section 3.2 —more precisely, we used TA-SVR as reported in the same section— and then use it as an extra input variable to a global SVR. We call this third method "SVR + $\alpha$".

In the following we describe the experimental settings. For benchmarking purposes, we consider nonstationary chaotic time series because they constitute one of the most challenging types of forecasting problems. Chaotic systems exhibit a sensitive dependence on initial conditions, meaning that nearby trajectories separate exponentially over time, thereby making medium to long term prediction difficult [2,4,36]. The sensitivity of a system to initial conditions can be measured with the Lyapunov exponent, which we now define. Two close starting trajectories in phase space, with initial separation $\delta Z_0$, will diverge at a rate given by $e^{\lambda t}|\delta Z_0|$, where $t$ is time and $\lambda$ the Lyapunov exponent. Since the separation rate depends on the orientation of the initial separation vector $\delta Z_0$, there is actually a spectrum of Lyapunov exponents. The number of Lyapunov exponents is equal to the number of dimensions of the phase space. However, it is common to only refer to the largest one, the maximum Lyapunov exponent (MLE), because it determines the overall predictability of the system. A positive MLE is usually taken as an indication that the system is chaotic. The systems considered in this work are not only chaotic but also nonstationary, as we describe below.

To compare the four modelling methods we worked on the same time-series employed by Verdes et al. [35]. They are all well-known, single-species discrete

chaotic ecosystem models, whose dynamics under external forcing has been already discussed by Summers et al. [30]. The models are the logistic map $x_{t+1} = \mu x_t(1 - x_t)$, the Moran-Ricker map, $x_{t+1} = x_t exp[r(1 - x_t/K)]$, and the Hassell map $x_{t+1} = \lambda x_t/(1 + x_t)^{\beta}$. To make the maps nonstationary we slowly changed one of the parameters in the previous definitions. In particular, we considered four cases: we drove the parameter $\mu$ for the logistic map, $K$ for the Moran-Ricker map, and $\lambda$ and $\beta$ for the Hassell map (one at a time). For the remaining parameter values we used the same base settings as in Verdes et al. [35]. We forced the dynamics using a piecewise constant profile, splitting the full time record into $s = 10$ equally-sized segments, and inside each one used a constant value $\alpha_t$ given by

$$\alpha_t = C_\alpha cos(2\pi t/T) exp(-t/T) + B_\alpha. \quad (4)$$

for a time $t$ corresponding to the middle point of each segment. We took $T = n/2$ so that the driving force profile is the same independently of the record length considered. In Figure 1 we depict this profile.

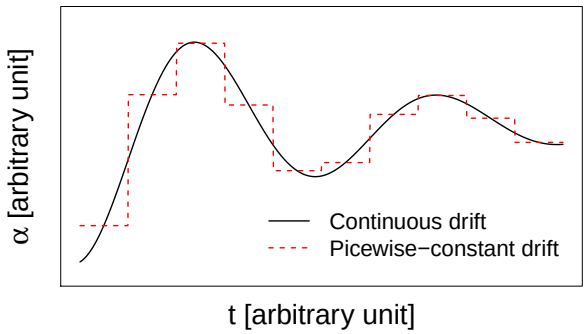


**Fig. 1** Profile of the parameter drift applied to the different maps, both in continuous and piecewise constant versions.

For each of the four problems, we produced three different versions by adding (observational) Gaussian noise with diverse signal-to-noise ratios, namely: i) no noise, ii) 0.1% noise, and iii) 1% noise.

For the four nonstationary modelling strategies compared in this Section, we used SVRs with a Gaussian kernel (defined as $\langle x, y\rangle = \exp(-\|x-y\|^2/\sigma)$) as a base model. The general procedure was the following. After generating $n = 1000$ points for each map, we added Gaussian noise in the required proportions. We separated a test set with 20% of randomly chosen datapoints, uniformly distributed over the different segments of the dataset, and used the remaining 80% for model fitting and selection. Using cross-validation in the training set, we optimised the different model parameters (C, $\epsilon$ and $\sigma$ for each SVR, $\gamma$ and $m$ for TA-SVR, and the optimal window length for SW) over a grid of values in a two-step procedure, starting with a coarse grid followed by a finer one centered at the optimal value obtained from the first step. Once the optimal models were determined for each interval, we predicted the test set. The full procedure was repeated 30 times in order to collect statistics.

**Table 1** Mean prediction error for the studied datasets, on randomly chosen test sets, for all methods tested in this work in the noise-free case. The MSE results for SW (base method) are expressed in $10^{-5}$ units. Performance for the remaining methods is expressed as a ratio to the corresponding SW result. Between brackets we report the standard error of MSE. Statistically significant underperformance with respect to TA-SVR is indicated with a symbol † ($p < 0.05$). For each row, the minimum number is highlighted in bold.

| Dataset | SW [MSE x $10^{-5}$] | TA-SVR | SVR + t | SVR + $\alpha$ |
|---|---|---|---|---|
| | | [Units relative to SW] | | |
| Logistic | 13.47 (0.75)† | 0.87 (0.05) | **0.83 (0.03)** | 0.86 (0.03) |
| Hassel ($\lambda$) | 58.15 (5.60)† | **0.22 (0.01)** | 0.90 (0.14)† | 0.67 (0.12)† |
| Hassel ($\beta$) | 67.37 (6.93)† | **0.28 (0.05)** | 0.90 (0.13)† | 0.71 (0.12)† |
| Mrn-Rckr | 42.28 (5.41) | **0.81 (0.08)** | 1.18 (0.15)† | 0.97 (0.13) |

**Table 2** Same as Table 1 with 0.1% added noise.

| Dataset | SW [MSE x $10^{-4}$] | TA-SVR | SVR + t | SVR + $\alpha$ |
|---|---|---|---|---|
| | | [Units relative to SW] | | |
| Logistic | 1.31 (0.07) | 1.08 (0.03) | 0.80 (0.04) | **0.78 (0.03)** |
| Hassel ($\lambda$) | 13.37 (0.81)† | **0.70 (0.02)** | 0.94 (0.06)† | 0.82 (0.04)† |
| Hassel ($\beta$) | 14.77 (0.96)† | **0.73 (0.02)** | 0.99 (0.08)† | 0.93 (0.06)† |
| Mrn-Rckr | 6.64 (0.66)† | 0.82 (0.11) | 0.95 (0.08)† | **0.74 (0.10)** |

**Table 3** Same as Table 1 with 1.0% added noise.

| Dataset | SW [MSE x $10^{-4}$] | TA-SVR | SVR + t | SVR + $\alpha$ |
|---|---|---|---|---|
| | | [Units relative to SW] | | |
| Logistic | **3.38 (0.10)** | **1.00 (0.03)** | 1.02 (0.04) | 1.03 (0.02) |
| Hassel ($\lambda$) | 324.0 (10.6)† | **0.88 (0.03)** | 1.05 (0.04)† | 1.02 (0.01)† |
| Hassel ($\beta$) | 353.4 (11.0) | 1.00 (0.04) | 1.03 (0.03)† | **0.99 (0.01)** |
| Mrn-Rckr | 95.51 (3.42)† | **0.72 (0.03)** | 0.94 (0.03)† | 0.86 (0.01)† |

In order to evaluate the performance of the considered modelling strategies, we computed the test set mean squared error $\text{MSE} = (1/n_T)\sum_{i=1}^{n_T}(y_i - \hat{y}_i)^2$, where $y$ is the target value, $\hat{y}$ the predicted one, and $n_T$ the test set size. In Tables 1, 2 and 3 we show the obtained results for the three different noise levels considered. We investigated whether the obtained differences are statistically significant by performing a set of paired $t$-tests whereby each methodology is in turn compared against TA-SVR. We use a symbol † in Tables 1, 2 and 3 to indicate that a given modelling approach is found to underperform TA-SVR in a statistically significant manner ($p < 0.05$). From Tables 1, 2 and 3 we conclude that TA-SVR is superior to the other methods included in this comparison, giving the best result in 8 out of the 12 cases under analysis.

As a final simulation experiment, we consider the case of a test set which is not randomly chosen but a block located at the end of the available data. For the sliding window approach (SW) the procedure in this setting is clear: the test set is predicted with the most recent available model. However, in order to apply the other considered forecasting methods to predict the continuation of a time series, some specific choices need to be made, namely:

**Table 4** Mean prediction error for the studied datasets, on block test sets at the end of the databases, for three methods tested in this work in the noise-free case. The MSE results for SW (base method) are expressed in $10^{-4}$ units. Performance for the remaining methods is expressed as a ratio to the corresponding SW result. Between brackets we report the standard error of MSE. Statistically significant underperformance with respect to TA-SVR is indicated with a symbol † ($p < 0.05$). For each row, the minimum number is highlighted in bold.

| Dataset | SW [MSE x $10^{-4}$] | TA-SVR [Units relative to SW] | SVR + t [Units relative to SW] |
|---|---|---|---|
| Logistic | 1.52 (0.09)† | **0.48 (0.03)** | 1.02 (0.09)† |
| Hassel ($\lambda$) | 23.07 (0.98)† | **0.23 (0.01)** | 3.57 (0.40)† |
| Hassel ($\beta$) | 19.18 (0.89)† | **0.26 (0.01)** | 6.62 (0.81)† |
| Mrn-Rckr | 212.5 (37.5)† | **0.15 (0.01)** | 0.16 (0.04) |

- TA-SVR: Which SVR model (training set window) should be employed to predict the test set?
- SVR+$t$: Should the time variable $t$ be extrapolated linearly into the test set, pushing it outside of its modelling domain?
- SVR+$\alpha$: How should the driving force profile $\alpha$ be extrapolated into the future?

In the case of TA-SVR, we decided to use the most recent SVR model to predict the test set. For SVR+$t$, we initially chose to linearly extrapolate time $t$, but this produced very poor results. Close inspection revealed that this was due to poor performance of SVR when test input data lies outside of the training set domain or support. We therefore adopted the view of fixing the value of $t$ for the complete test set to the last time value seen on the training set. The extrapolation of the driving force profile for the SVR+$\alpha$ method would involve a study of optimal methodological approaches, the discussion of which is beyond the scope of this work. We therefore chose to leave SVR+$\alpha$ out of this forecasting exercise. Finally, for this study we reverted to the continuous (smooth) driving force profile, shown in Figure 1, because a jump in $\alpha$ from the last training to the test intervals would not only represent an unlikely (and unlucky) coincidence for the practitioner but would also dominate the prediction error figures thereby hindering the comparison of their intrinsic performance.

The prediction protocol followed similar lines to the previous one, namely: 1) we generated 1000 points for each map and added Gaussian noise in the required proportions (0, 0.1, and 1%, respectively); 2) we separated a test set with the last 100 data points, and used the remaining 90% for model fitting and selection; 3) we determined the different model parameters as above, but this time using a (block) validation set consisting of the last 100 data points of the training set; 4) once the optimal model parameters were determined for each interval, we built models using the full training set and predicted the test set. The complete procedure was repeated 30 times in order to collect statistics. The obtained results are reported in Tables 4, 5 and 6. As we can see, TA-SVR performs very well, doing better than SW and SVR+$t$ in almost all considered instances. For the Moran-Ricker map we find that the performance of TA-SVR and SVR+$t$ is equally good.

**Table 5** Same as Table 4 with 0.1% added noise.

| Dataset | SW [MSE x $10^{-4}$] | TA-SVR [Units relative to SW] | SVR + t |
|---|---|---|---|
| Logistic | 1.49 (0.07)$^{\dagger}$ | **0.80 (0.05)** | 0.86 (0.08) |
| Hassel ($\lambda$) | 29.04 (1.26)$^{\dagger}$ | **0.49 (0.03)** | 3.48 (0.30)$^{\dagger}$ |
| Hassel ($\beta$) | 28.18 (1.06)$^{\dagger}$ | **0.53 (0.03)** | 4.73 (0.62)$^{\dagger}$ |
| Mrn-Rckr | 152.1 (15.8)$^{\dagger}$ | 0.23 (0.02) | **0.20 (0.02)** |

**Table 6** Same as Table 4 with 1.0% added noise.

| Dataset | SW [MSE x $10^{-4}$] | TA-SVR [Units relative to SW] | SVR + t |
|---|---|---|---|
| Logistic | 3.51 (0.14) | **0.91 (0.04)** | 1.22 (0.06)$^{\dagger}$ |
| Hassel ($\lambda$) | 274.9 (17.0)$^{\dagger}$ | **0.94 (0.05)** | 2.52 (0.42)$^{\dagger}$ |
| Hassel ($\beta$) | 291.6 (16.7) | **0.96 (0.06)** | 2.34 (0.33)$^{\dagger}$ |
| Mrn-Rckr | 269.0 (22.1)$^{\dagger}$ | **0.53 (0.05)** | 0.88 (0.09)$^{\dagger}$ |

### 3.2 Driving force reconstruction

In this second application of TA-SVR we show how it can be used to improve the driving force profile reconstruction. We selected the reconstruction approach introduced in Verdes et al. [35] because it can be used with any kind of model family, as opposed to the slightly improved method by Széliga et al. [32], which is limited to neural network models. The selected method is based on the fact that, for two consecutive data segments generated by a driven system, the change in prediction error from the first to the second segment, for a model trained on the first segment, is proportional (to first order approximation) to the change in the driving parameter. The accuracy of the reconstruction is related to the model goodness of fit, which, in view of the results discussed in the previous subsection, suggests the use of TA-SVR in this problem.

To evaluate this hypothesis we used the same experimental settings as in the previous subsection. In this case we applied two different methods (SW as in Verdes et al. [35] and TA-SVR) to model the diverse systems and then reconstruct the changing parameter profile.

To compare both methods we computed the MSE between the original and reconstructed profiles, $\mathrm{MSE}_{\alpha} = \sum_{t=1}^{n}(r_t - \alpha_t)^2$, where $r$ denotes the imposed parameter variation (scaled to zero mean and unit variance) and $\alpha$ the reconstructed profile (with the same scaling). The corresponding results for $\mathrm{MSE}_{\alpha}$ are given in Table 7. It is clear from this table that the improved nonstationary modelling of TA-SVR leads to a better reconstruction of the driving force in all situations. As an illustrative example, in Figure 2 we show the mean reconstructed profiles together with the actual one.

We also explored the use of a continuous driving force profile in Eq. 4 instead of a piecewise constant one (see Fig. 1), using the same settings as before, for the noise-free scenario. The results in Table 8 indicate that the coupled modelling of TA-SVM also outperforms the independent modelling of SW in this (more difficult) case in which the individual models in the sequence are unable to accurately approximate the original maps due to the continuous drift of the forcing.

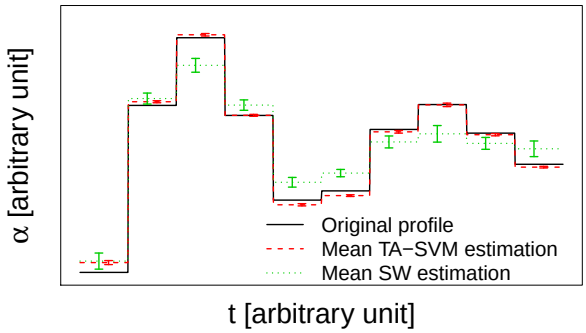


**Fig. 2** An example of reconstructed drift profiles using TA-SVM and SW. The figure shows the mean value and standard error, over the 30 experiments, for the noise-free Hassel map case (varying $\lambda$).

**Table 7** Driving force reconstruction error for the method introduced in Verdes et al. [35] using the SW and TA-SVR methods, with 10 modelling functions. Between brackets we report the standard error of the reconstruction error.

| Noise | Dataset | SW [MSE] | TA-SVR [relative units] |
|---|---|---|---|
| 0% | Logistic | 0.07 (0.02) | **0.42 (0.33)** |
| | Hassel ($\lambda$) | 0.35 (0.09) | **0.04 (0.01)** |
| | Hassel ($\beta$) | 0.56 (0.10) | **0.04 (0.01)** |
| | Moran Ricker | 0.17 (0.02) | **0.44 (0.08)** |
| 0.1% | Logistic | 0.50 (0.09) | **0.65 (0.17)** |
| | Hassel ($\lambda$) | 0.47 (0.08) | **0.28 (0.11)** |
| | Hassel ($\beta$) | 0.59 (0.10) | **0.20 (0.07)** |
| | Moran Ricker | 0.20 (0.06) | **0.41 (0.07)** |
| 1.0% | Logistic | 1.50 (0.11) | **0.91 (0.07)** |
| | Hassel ($\lambda$) | 0.89 (0.09) | **0.33 (0.07)** |
| | Hassel ($\beta$) | 0.61 (0.08) | **0.56 (0.12)** |
| | Moran Ricker | 0.18 (0.07) | **0.97 (0.39)** |

Finally, we used noise-free data to briefly evaluate the reconstruction error dependence on $n$ and $m$. First, we doubled the number of modelling functions in the sequence, i.e. $m = 20$, and also doubled the number of segments in the piecewise constant driving force $s = 20$. This is a more challenging setting for all approaches, as each model is fed with less information than before, which is confirmed by the larger $\text{MSE}_\alpha$ values reported in Table 9. Again, TA-SVM clearly outperforms SW in this task. As a last experiment we used the original configuration, i.e. 10 modelling functions ($m = 10$) and 10 segments in the driving force ($s = 10$), but halved the total length of the sequence, which also increased the difficulty of the modelling problem. However, as we can see from Table 10, TA-SVM still exhibits a significant outperformance with respect to SW.

**Table 8** Reconstruction error for a continuous driving force.

| Dataset | SW [MSE] | TA-SVR [relative units] |
|---|---|---|
| Logistic | 0.20 (0.06) | **0.45 (0.17)** |
| Hassel ($\lambda$) | 0.27 (0.05) | **0.19 (0.08)** |
| Hassel ($\beta$) | 0.59 (0.11) | **0.11 (0.04)** |
| Moran Ricker | 0.37 (0.08) | **0.27 (0.09)** |

**Table 9** Driving force reconstruction error when doubling the number of prediction functions in the sequence.

| Dataset | SW [MSE] | TA-SVR [relative units] |
|---|---|---|
| Logistic | 0.99 (0.14) | **0.13 (0.06)** |
| Hassel ($\lambda$) | 1.71 (0.07) | **0.22 (0.06)** |
| Hassel ($\beta$) | 1.76 (0.10) | **0.15 (0.03)** |
| Moran Ricker | 1.20 (0.16) | **0.51 (0.15)** |

**Table 10** Driving force reconstruction error when shortening the dataset to half of the original length.

| Dataset | SW [MSE] | TA-SVR [relative units] |
|---|---|---|
| Logistic | 0.24 (0.06) | **0.05 (0.02)** |
| Hassel ($\lambda$) | 1.10 (0.12) | **0.07 (0.03)** |
| Hassel ($\beta$) | 1.24 (0.10) | **0.06 (0.03)** |
| Moran Ricker | 0.43 (0.09) | **0.20 (0.05)** |

### 3.3 Input feature relevance

In this last application we analyse a different type of nonstationarity. In the previous problems some parameter changed over time, but the input-output functional dependence was fixed. Here we analyse a problem in which there is a drift in the system which is associated to a change in the relative importance of two input features, not only to a change in a hidden parameter. For example, suppose that we have a movie recommendation system in which the inputs are qualitative aspects of the movie, like genre, country of origin, director, etc., and the output is the estimated ranking that a given user would give to the movie. During a long period of time, the most relevant feature, for a user that likes war movies, is 'genre'. Then, at a given time, the user becomes a fan of French movies, and then the most relevant feature changes to 'country of origin'. This is the kind of change that we would like to detect in this application.

One of the interesting properties of SVR is that the relative importance of each input can be easily estimated, following the recursive feature elimination (RFE) method introduced by Guyon et al. [17]. The main idea behind RFE is that the importance of a given input variable is directly related to the second derivative of the SVR's cost function with respect to that input. We propose to use TA-SVR as an improved detector of changes in relative input's importance, applying to

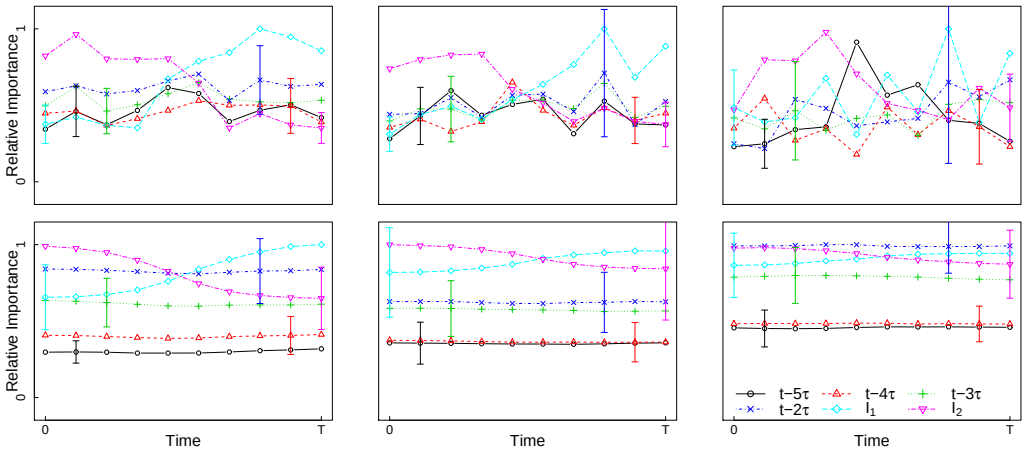


**Fig. 3** Change in input feature relevance for the Ikeda map estimated with the SW method (top row) and TA-SVR (bottom row). From left to right, columns correspond to databases of size 1000, 500 and 200. Error bars indicate the standard error of the mean.

each of the coupled SVRs the RFE method. We compare the performance of this combination with the estimation obtained by RFE with the typical SW strategy.

In this experiment we used the Ikeda map [18]. We generated a long record with 5,000 time points, which we embedded in a 6-dimensional space of lagged copies of the series according to: $x_t = f(I_1, I_2, x_{t-2\tau}, x_{t-3\tau}, x_{t-4\tau}, x_{t-5\tau})$, with $\tau = 1$ time units, $I_1 = \alpha_t x_{t-\tau} + (1-\alpha_t)\varepsilon_t$ and $I_2 = (1-\alpha_t)x_{t-\tau} + \alpha_t \zeta_t$, where $\varepsilon$ and $\zeta$ are centered Gaussian noise variables with the same variance as $x_t$, and $\alpha_t$ is a sigmoid function of $t$ centered at the dataset midpoint. These two special inputs are used to simulate a problem in which there is a slow shift from a model depending on $I_1$ to $I_2$.

From the full dataset we took random samples with 200, 500 and 1000 datapoints, 30 sets for each length. For each sample we applied the procedure previously described to select all free parameters ($C$, $\gamma$, etc.) and, using those optimal values, constructed a sequence of 10 independent SVRs (for the SW strategy) and a single TA-SVR with $m = 10$ coupled models. Finally, we applied the procedure described by Guyon et al. [17] to estimate the importance of each input.

In Figure 3 we show the obtained results. In each panel we report mean relevance values estimated over 30 runs. The top row corresponds to the SW method, while the bottom one to TA-SVR. In the left column, corresponding to the largest dataset size of 1000 points, we see that both methods clearly detect the relevance shift from $I_1$ to $I_2$. In the central column, corresponding to 500 points, the SW estimation becomes noisier than TA-SVR but the drift can still be detected. For small datasets, in the right column, we find that the SW method can no longer be used to detect the dependence drift, in contrast with the good results obtained with TA-SVR. Overall, it is clear that the regularisation provided by the coupling term in TA-SVR helps produce a better estimation of the relative importance of each model input.

## 4 Conclusions

In this work we introduced the TA-SVR strategy, an extension to the regression case of the previously developed TA-SVM. Here we illustrated its application to nonstationary chaotic time series only, but it should be noted that the methodology can be applied to nonstationary modelling problems of any kind.

We first analysed the modelling task on four different nonstationary regression problems. Upon comparison with three other efficient modelling methods from the literature, TA-SVR proved to be superior to its competitors on this task.

We also compared TA-SVR against the sliding window strategy on two other aspects of nonstationary modelling: hidden parameter variation estimation (or driving force reconstruction) and input feature relevance determination under a dependency drift. On both tasks we found that the proposed TA-SVR is more efficient than the sliding window approach.

The three nonstationary data analysis exercises considered in this work are different in nature but share a common property: their solutions follow from a regression model fitted to some dataset. As such, the good results of TA-SVR on these tasks can be probably related to its better performance in nonstationary modelling, produced, as in its classification version, by its more comprehensive use of information along the full sequence of models through the coupling term.

Future work includes a theoretical analysis of the properties of TA-SVM and TA-SVR.

## Acknowledgements

The authors acknowledge partial support from ANPCyT.

## References

1. Alippi, C., Roveri, M.: Just-in-time adaptive classifiers–part i: Detecting nonstationary changes. IEEE Transactions on Neural Networks **19**(7), 1145–1153 (2008).
2. Andrawis, R.R., Atiya, A.F., El-Shishiny, H.: Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. International Journal of Forecasting **27**(3), 672–688 (2011).
3. Bartlett, P.L., Ben-Dabid, S., Kulkarni, S.R.: Learning changing concepts by exploiting the structure of change. Machine Learning **41**(2), 153–174 (2000).
4. Ben Taieb, S., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Systems with Applications **39**(8), 7067–7083 (2012).
5. Camci, F., Chinnam, R.: General support vector representation machine for one-class classification of non-stationary classes. Pattern Recognition **41**(10), 3021–3034 (2008).
6. Cao, L., Tay, F.: Financial forecasting using support vector machines. Neural Computing & Applications **10**(2), 184–192 (2001).
7. Cao, L., Tay, F.: Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on Neural Networks **14**(6), 1506–1518 (2004).
8. Cao, Y., Tung, W., Gao, J., Protopopescu, V., Hively, L.: Detecting dynamical changes in time series using the permutation entropy. Physical Review E **70**(4), 046217 (2004).
9. Casdagli, M.: Nonlinear prediction of chaotic time series. Physica D: Nonlinear Phenomena **35**(3), 335–356 (1989).
10. Chang, M.W., Lin, C.J., Weng, R.C.: Analysis of nonstationary time series using support vector machines. In: Pattern Recognition with Support Vector Machines, SVM 2002, LNCS 2388, pp. 239–255. Springer (2002).

11. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000).
12. Drucker, H., Burges, C., Kaufman, L., Smola, A., Vapnik, V.: Support vector regression machines. In: NIPS 9, Advances in neural information processing systems, pp. 155–161. Citeseer (1997).
13. Ghazikhani, A., Monsefi, R., Yazdi, H.S.: Online cost-sensitive neural network classifiers for non-stationary and imbalanced data streams. Neural Computing & Applications, **23**(5), 1283–1295 (2013).
14. Gotelli, A., Ellison, N.: Forecasting extinction risk with nonstationary matrix models. Ecological Applications **16**(1), 51–61 (2006).
15. Grinblat, G., Uzal, C., Ceccatto, H., Granitto, P.: Solving non-stationary classification problems with coupled support vector machines. IEEE Transactions on Neural Networks **22**(1), 27–51 (2011).
16. Grinblat, G., Uzal, C., Granitto, P.: Abrupt Change Detection with One-Class Time-Adaptive Support Vector Machines. Expert Systems with Applications **40**(18), 7242–7249 (2013).
17. Guyon, I., Weston, S., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning **46**(1-3), 389–422 (2002).
18. Ikeda, K.: Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system. Optics Communications **30**(2), 257–261 (1979).
19. Koychev, I.: Gradual forgetting for adaptation to concept drift. In: In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning, pp. 101–106 (2000).
20. Li, L., Yang, Z., Wang, B., Kitsuregawa, M.: Dynamic adaptation strategies for long-term and short-term user profile to personalize search. Advances in Data and Web Management pp. 228–240 (2007).
21. Narwaria, M., Lin, W.: Objective image quality assessment based on support vector regression. IEEE Transactions on Neural Networks **21**(3), 515–519 (2010).
22. Pang, S., Song, L., Kasabov, N.: Correlation-aided support vector regression for forex time series prediction. Neural computing & applications **20**(8), 1193–1203 (2011).
23. Pelletier, L., Almhana, J., Choulakian, V.: Adaptive filtering of spam. In: Communication Networks and Services Research, 2004. Proceedings. Second Annual Conference on, pp. 218–224. IEEE (2004).
24. Pereira Salazar, D.S., Leitao Adeodato, P.J., Lucena Arnaud, A.: Continuous Dynamical Combination of Short and Long-Term Forecasts for Nonstationary Time Series. IEEE Transactions on Neural Networks and Learning Systems **25**(1), 241–246 (2014).
25. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning, pp. 185–208 (2000).
26. Schreiber, T.: Detecting and analyzing nonstationarity in a time series using nonlinear cross predictions. Physical Review Letters **78**(5), 843–846 (1997).
27. Shi,Y.-Z., Wang, S.-T., Wang, J., Deng, Z.-H.: Fast classification for nonstationary large scale data sets using minimal enclosing ball. Kongzhi yu Juece/Control and Decision **28**, 1065–1072 (2013).
28. Smola, A., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**(3), 199–222 (2004).
29. Stark, J.: Delay embeddings for forced systems. i. deterministic forcing. Journal of Nonlinear Science **9**, 255–332 (1999).
30. Summers, D., Cranford, J., Healey, B.: Chaos in periodically forced discrete-time ecosystem models. Chaos, Solitons & Fractals **11**(14), 2331–2342 (2000).
31. Susnjak, T., Barczak, A.L.C., Hawick, K.A.: Adaptive cascade of boosted ensembles for face detection in concept drift. Neural Computing and Applications **21**(4), 671–682 (2012).
32. Széliga, M., Verdes, P., Granitto, P., Ceccatto, H.: Modelling nonstationary dynamics. Physica A: Statistical Mechanics and its Applications **327**(1-2), 190–194 (2003).
33. Tay, F., Cao, L.: $\varepsilon$-descending support vector machines for financial time series forecasting. Neural Processing Letters **15**(2), 179–195 (2002).
34. Verdes, P.F., Granitto, P.M., Ceccatto, H.A.: Overembedding method for modelling nonstationary systems. Phys. Rev. Lett. **96**(11), 118,701 (2006). DOI 10.1103/PhysRevLett.96.118701
35. Verdes, P.F., Granitto, P.M., Navone, H.D., Ceccatto, H.A.: Nonstationary time-series analysis: Accurate reconstruction of driving forces. Phys. Rev. Lett. **87**(12), 124,101 (2001). DOI 10.1103/PhysRevLett.87.124101

36. Wong, W.K., Xia, M., Chu, W.C.: Adaptive neural network model for time-series forecasting. European Journal of Operational Research **207**(2), 807–816 (2010).
37. Wu, C., Ho, J., Lee, D.: Travel-time prediction with support vector regression. IEEE Transactions on Intelligent Transportation Systems **5**(4), 276–281 (2004).