Predicting Time Series with a Local Support Vector Regression Machine

Rodrigo Fernández
LIPN, Institut Galilée-Université Paris 13
Avenue J.B. Clément 93430 Villetaneuse France
rf@lipn.univ-paris13.fr

Abstract

Recently, a new Support Vector Regression (SVR) algorithm has been proposed, this approach, called ν-SV regression allows the SVR to adjust its accuracy parameter ϵ automatically. In this paper, we combine ν-SV regression with a local approach in order to obtain accurate estimations of both the function and the noise distribution. This approach seems to be extremely useful when the noise distribution does depend on the input values. We illustrate the properties of the algorithm by a toy example and we benchmark our approach on a 100000 points time series (Santa Fe Competition data set D) obtaining the state of the art performance on this data set.

1 Introduction

Support Vector (SV) Machines are becoming a powerful and useful tool for classification and regression tasks. SV regressors are able to approximate a real valued function $y(x)$ in terms of a small subset (named *support vectors*) of the training examples. In [7], together with the SVR algorithm itself, Vapnik proposes a loss function with nice properties: $|\eta|_\epsilon = max\{0, |\eta| - \epsilon\}$, where $\eta = y - f(x)$ is the estimation error. This loss function, called ϵ-insensitive, does not penalize errors smaller than $\epsilon > 0$, which has to be chosen a *priori*.

In [3], Schölkopf et al. propose a new SVR algorithm, called ν-SV regression (ν-SVR), which adjusts automatically the parameter ϵ. In this paper, we combine ν-SVR and a local approach to obtain accurate estimations of both the function $y(x)$ and the noise model. The paper is organized as follows: in the Section 2 we briefly sketch our algorithm, the Section 3 illustrates the properties of the algorithm by a toy example, in the Section 4 we benchmark our approach on a 100000 points time series (Santa Fe Competition data set D), a brief discussion of the results an of some possible further issues concludes the paper.

2 Local ν-SV regression

Support Vector Regression seeks to estimate functions

$$f(x) = w \cdot x + b, \quad w, x \in I\!R^n, b \in I\!R \tag{1}$$

starting from l independent identically distributed (iid) data $\{(x_i, y_i)\} \subset I\!R^n \times I\!R$. We show here how SVR can be used when data are not iid. For example, for non-stationary time series forecast.

The ν-SVR algorithm minimizes the risk

$$R_\nu[f] = \frac{1}{2}\|w\|^2 + C \cdot \frac{1}{l}\sum_{i=1}^{l} |y_i - f(x_i)|_\epsilon + C\nu\epsilon. \tag{2}$$

The solution (w, b, ϵ) of (2) can be obtained by solving the following QP problem:

$$max_{\alpha,\alpha^*}\Big[-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i \cdot x_j + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*)\Big] \tag{3}$$

subject to

$$\begin{aligned}
\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) &= 0 \\
\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) &\leq C\nu l \\
\alpha_i, \alpha_{i*} &\in [0, C] \quad i = 1, \cdots, l.
\end{aligned}$$

The quantities ϵ and b are the dual variables of (3), as we use an interior point primal-dual solver LOQO [4], they can be recovered easily. The computation of w follows the standard SV techniques: $w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)x_i$. The extension of the algorithm to non linear regression via the Hilbert-Schmidt kernels follows also standard SV techniques. See [6] for the details.

Among other interesting properties, Schölkopf et al. prove that the positive quantity ν is an upper bound on the fraction of errors and a lower bound on the fraction of support vectors [3].

Our local approach for support vector regression is inspired by the *Local Linear* classifier, proposed by Bottou and Vapnik [1]. The algorithm is simple, let be $Z = \{(x_1, y_1) \cdots (x_l, y_l)\}$ the training set and let be x^* a test input, in order to estimate the output $y(x^*)$, Local SVR proceed as follows:

1. Compute the M nearest neighbors, in the input space, of x^* among the training inputs. these M neighbors define a subset of the training set. Let's call $Z_M(x^*) \subset Z$ this subset.

2. Using $Z_M(x^*)$ as learning set, compute a SVRM. This SVRM approximates $y(x)$ in a neighborhood of x^* by $f_M^*(x)$.

3. Estimate $y(x^*)$ by $f_M^*(x^*)$.

The combined choice of ν and M bounds tightly the number of training examples playing a role in the expansion of f_M^*, in fact, M defines the number of examples used to compute the mini SVRM and ν bounds the proportion of these examples supporting f_M^*. For example, if $M = 10$ and $\nu = 0.3$, the number of training examples that support the local estimation is between 3 and 10.

Remark that every Local ν-SVR adjusts automatically the (local) accuracy parameter ϵ. It is clear that ϵ is directly related to the noise in the following sense: the more important is the noise rate the bigger is the ϵ computed by ν-SVR. That means that Local ν-SVR estimates simultaneously the target function and the noise. Since the parameter M is much smaller than l, a single-point regression using local SVR is as fast (or as slow) as the KNN algorithm.

3 A toy example

The task is to estimate the *sinc* function defined by $sinc(x) = sin(\pi x)/(\pi x) + \xi(x)$ where the noise $\xi(x)$ is input dependent[1]. The learning set is composed by 41 samples drawn uniformly from the interval [-2,2]. In order to test the ability of the ν-SVR to match the noise, two different noise models were used. Two kind of regressors were trained: a global (or standard) ν-SVR and a local ν-SVR. The risk (or test error) was computed with respect to the *sinc* function without noise over the interval [-1.5,1.5], the results are averaged over 50 trials.

The Figure 1 shows two examples of noise model estimation via the *automatic ϵ-tube* one gets from the regression. For each test point x, local ν-SVR provides, besides the estimation of the target function, a value of ϵ which can be interpreted as the estimated accuracy of the model in a neighborhood of x. After processing the whole train set, one gets a curve $\epsilon(x)$ estimating the accuracy of the model at each point. The results are given in Table 1, for each noise model, the best 3 performances are reported. In both cases the local approach performs better than the global one.

4 Time series prediction

We implemented the local ν-SVR algorithm using LOQO as solver for the subproblems. The algorithm was tested on a time series known to be long (100000 points) and difficult (the series is non-stationary). Data set D from Santa Fe Competition [8] is artificial data generated by numerically integrating the equations of motion for a dumped, driven particle

$$\frac{d^2x}{dt^2} + \gamma\frac{dx}{dt} + \nabla V(x) = F(t).$$

In the experiences, we used an embedding size of 25, 20 and 15 consecutive points; the kernel for the SVMs is a RBF kernel with $\sigma^2 = 0.03$ and the regularization parameter C was fixed to 10, no cross-validation over these parameters was made, but some preliminary experiments ensure that these values of σ and C works fairly well. The last 100 points of the series was used to determine, by cross validation over 1-step forecast, the parameters of the local ν-SVR algorithm M and ν). For the final prediction the full time series was used. The Tables 2,3,4 show the results of the cross-validation procedure for

[1]That is, $\xi(x) = u(x)a(x)$, where $u(x)$ follows a uniform distribution over [-0.5,0.5] and $a(x)$ modules its amplitude. Different choices of $a(x)$ allows us to different noise models.

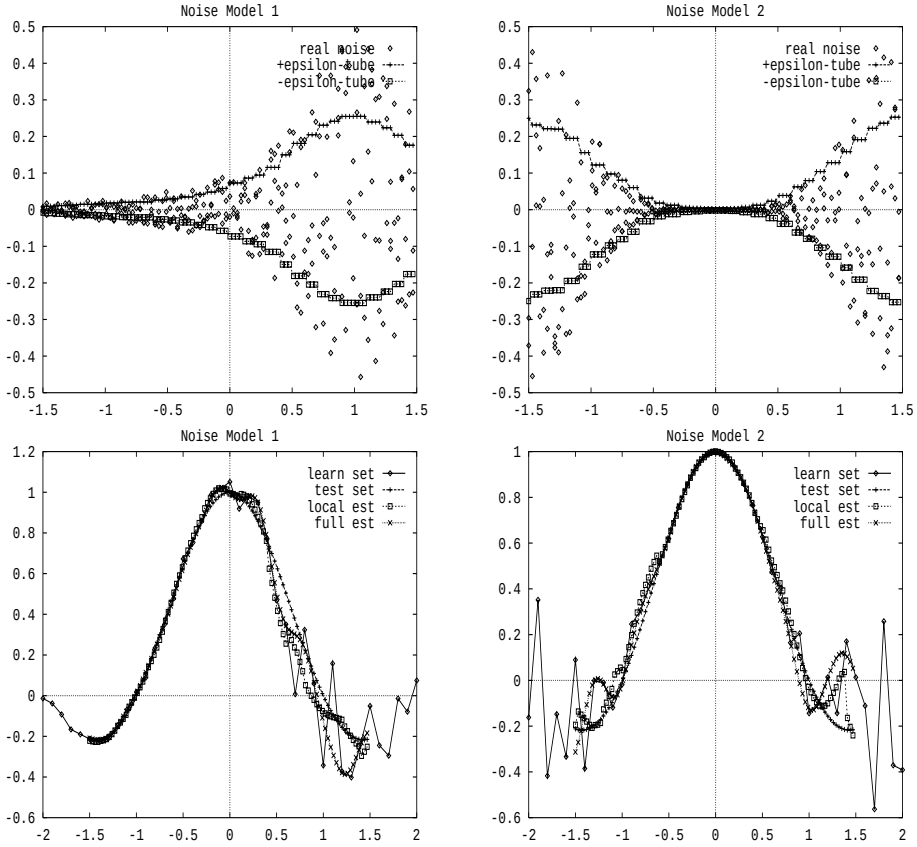

Figure 1: Top: The automatic ϵ estimates the noise model. In this figure we compare the smoothed $\epsilon(x)$-tube ($\nu = 0.03$, averaged over 50 trials) with a sample of each noise model. Down: Regression examples (only one trial). According to the results shown in 1. Left: Noise Model 1, right: Noise Model 2.

Noise Model 1			
Full		Local	
ν	error	ν	error
0.5	0.077	0.1	0.057
0.4	0.078	0.05	0.051
0.3	0.081	0.03	**0.048**

Noise Model 2			
Full		Local	
ν	error	ν	error
0.5	0.070	0.1	0.056
0.4	0.069	0.05	**0.052**
0.3	0.077	0.03	0.056

Table 1: Local ν-SVR consistently outperforms the full regression. Remark that full ν-SVR performs well for big values of ν while local ν-SVR works well with little values of ν.

different lengths of the embedding dimension, only the best results are reported. The couples (ν, M) achieving the minimum validation error are selected for the final forecast.

The Figure 2 shows the best 25 step iterated prediction, in the Table 5 we compare the Root Mean Squared error for the first 25 predicted values to the error achieved by other algorithms. We include the results reported by Müller et al. [2] and Pawelzik et al. [5] using a segmentation of the learning set into regimes of approximately stationary dynamics. The best local ν-SVR algorithm is 39% better than the best result reported using the full set and it is slightly better than the best result reported using the segmentation preprocessing, and, in general, local ν-SVR performs consistently better than global approaches. This result is remarkable because in our experimental setup any *a priori* assumption about the stationarity of the data was made.

5 Discussion

This paper presents a local regression algorithm that profits from the properties of the ν-SV regression. Local ν-SVR outperforms global (or "standard") SV regression, one can explain that by the fact that the local approach is free to change the radius of the ϵ-tube for each point to be estimated (see Figure 1), hence it is more robust to the variations of the amplitude of the noise. In addition, it provides an estimation of the noise that matches quite well the real noise model. The results on the Santa Fe data set D are excellent. Local ν-SVR algorithm outperforms by nearly 40% the best result reported using the full set and it is comparable (slightly better in fact) to the best result reported using a segmentation algorithm as preprocessing. Since we used the full 100000 points series to predict the next 25 values, very little information about the stationarity of the data was used in our prediction[2]. It will be interesting to test the local approach over the segmented set and to compare the estimation of the noise model over the training set provided by the local approach with the segmentation proposed in [5]

[2]The parameters M and ν were determined using the last 100 values of the series. One could think that the dynamics of the last 100 points of the series and of the 25 predicted points are the same, in this sense, our predictor incorporates a little amount of information about the stationarity of the series.

Validation error, 15 points				
M	$\nu = 0.012$	$\nu = 0.015$	$\nu = 0.018$	$\nu = 0.021$
6	0.0304	0.0303	0.0303	0.0306
7	0.0304	**0.0300**	0.0300	0.0302
8	0.0303	0.0302	0.0304	0.0308
10	0.0326	0.0332	0.0338	0.0345

Table 2: One-step forecast over the last 100 points of the series. The Root Mean Squared (RMS) Error as a function of M et ν is reportetd. The embedding size is 15, $\sigma = 0.03$ and $C = 10$.

Validation error, 20 points				
M	$\nu = 0.020$	$\nu = 0.030$	$\nu = 0.040$	$\nu = 0.050$
5	0.0332	0.0326	0.0320	0.0320
6	0.0288	0.0280	**0.0278**	0.0281
7	0.0299	0.0297	0.0301	0.0306
8	0.0321	0.0327	0.0341	0.0352

Table 3: One-step forecast over the last 100 points of the series. The RMS error as a function of M et ν is reportetd. The embedding size is 20, $\sigma = 0.03$ and $C = 10$.

Validation error, 25 points				
M	$\nu = 0.010$	$\nu = 0.014$	$\nu = 0.020$	$\nu = 0.024$
6	0.0355	0.0354	0355	0357
8	0.0327	0.0323	**0.0319**	0.0319
10	0.0343	0.0336	0.0332	0.0332
12	0.0329	0.0324	0.0323	0.0326

Table 4: One-step forecast over the last 100 points of the series. The RMS error as a function of M et ν is reportetd. The embedding size is 25, $\sigma = 0.03$ and $C = 10$.

	p=15	p=20	p=25	Müller [2]	ZH [9]	PKM [5]
full set	0.0456	0.0539	**0.0391**	0.0639	0.0665	-
segmented set	-	-	-	0.0418	-	0.0596

Table 5: Root Mean Squared error for the first 25 predicted values achieved by several algorithms. Full set predictions assume stationary data, segmented set predictions use a segmentation of the learning set into regimes of approximately stationary dynamics. '-' indicates no prediction available.

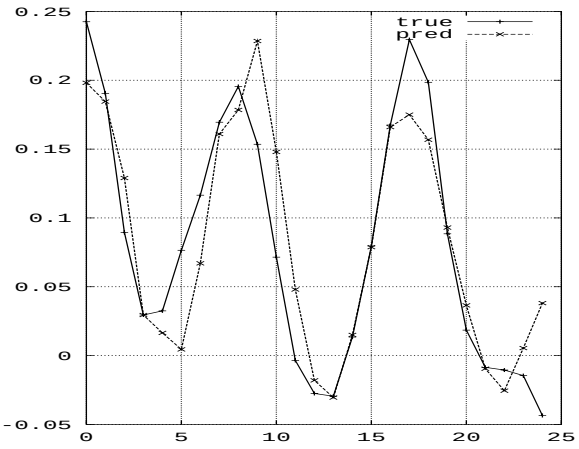

Figure 2: Santa Fe data set D. 25 step iterated prediction using an embedding dimension of 25 points. The mean value of the full training set (0.5705) was substracted of both curves.

References

[1] L. Bottou, V. Vapnik. Local Learning Algorithms. Neural Computation, 4:888-900. 1992.

[2] K-R. Müller, A. Smola, G. Rätsh, B. Schölkopf, J. Kohlmorgen and V. Vapnik. Predicting time series with support vector machines. In Proceedengs ICANN'97. 1997.

[3] B. Schölkopf, P. Bartlett, A. Smola, R. Williamson. Support Vector Regression with Automatic Accuracy Control. Proceedings ICANN 98, 1998.

[4] R.J. Vanderbei. LOQO: User's manual. Program in Statistics and Operational Research, Princeton University. 1997.

[5] K. Pawelzik, J. Kohlmorgen and K-R. Müller. Annealed competition of experts for segmentation and classification of switching dynamics. Neural Computation, 8(2):342-358. 1996.

[6] A.J. Smola, B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report NeuroCOLT2 TR-1998-030. 1998.

[7] V. Vapnik. The Nature of Statistical Learning Theory. Springer. 1995.

[8] A. Weigend and N. Gershenfeld (Eds.). Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley. 1994.

[9] X. Zhang and J. Hutchinson. Simple architectures on fast machines: practical issues in nonlinear time series prediction. In [8]. 1994.