An Adaptive Support Vector Regression Filter: A Signal Detection Application

Roman Rosipal and Mark Girolami
Computational Intelligence Research Unit
Department of Computing and Information Systems
University of Paisley
Paisley PA1 2BE, Scotland
e-mail: {rosi-ci0, giro-ci0}@paisley.ac.uk

Abstract

A new method for the construction of nonlinear adaptive filters called adaptive support vector regression is introduced for signal detection in noisy environments. A modification of support vector regression for online learning is motivated by the *chunking* approach and is based on repeated re-training of the filter parameters without the loss of former estimates. Performance of the proposed method was found superior to the method using a Resource-Allocating RBF network with Givens QR decomposition and pruning [8].

1 Introduction

Support Vector Regression (SVR) is a new technique which has been applied successfully to regression and function estimation. In [5] the superior performance of SVR is compared to Radial Basis Function (RBF) networks for noisy chaotic time-series prediction. In [4] a comparison of SVR with several existing approximation techniques have been carried out.

An adaptive version of the SVR technique based on block-by-block re-training of the SVR approximation of the chaotic time-series was introduced in [2].

In this contribution we implement a modified adaptive SVR (ASVR) technique for the construction of nonlinear adaptive filters and demonstrate its performance in the detection of a chaotic Mackey-Glass time-series in noisy conditions. We compare ASVR with the modified Resource-Allocating Networks method introduced in [8]. We report encouraging results showing the superior performance of the ASVR technique on a noisy chaotic time-series.

2 Support Vector Regression

The idea of the SVR technique is based on computation of the linear regression function in a high dimensional feature space Ψ where the input data $\mathbf{x}$ are mapped via some nonlinear function. Thus, the SVR problem can be defined as the determination of function $f(\mathbf{x}, \mathbf{w})$ which approximates an unknown desired function and has the following form:

$$f(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{M} w_i \psi_i(\mathbf{x}) + b = (\mathbf{w}.\psi(\mathbf{x})) + b,$$

where b and $\{w_i\}_{i=1}^{M}$ are unknown coefficients, $(\mathbf{w}.\psi(\mathbf{x}))$ is a dot product in M-dimensional feature space Ψ ($M \leq \infty$). The function $f(\mathbf{x}, \mathbf{w})$ represents a hyperplane in feature space Ψ defined by the functions $\{\psi_i\}_{i=1}^{M}$. In [13] the following regularized risk functional is stated to compute the unknown coefficients b and $\{w_i\}_{i=1}^{M}$:

$$R(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} |Err|_\epsilon + \frac{\beta}{2} \|\mathbf{w}\|^2, \qquad (1)$$

where $Err = y_i - f(\mathbf{x}_i, \mathbf{w})$, $\beta \geq 0$ is a regularization constant to control the trade-off between complexity and accuracy of the regression model and

$$|Err|_\epsilon = \left\{ \begin{array}{rcl} 0 & : & |Err| < \epsilon \\ |Err| - \epsilon & : & otherwise \end{array} \right.$$

is Vapnik's ϵ-insensitive loss function [13]. In [13] Vapnik also showed that the regression estimate that minimizes the risk functional (1) has the form:

$$\begin{aligned} f(\mathbf{x}, \boldsymbol{\alpha}) &= \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)(\psi(\mathbf{x}_i).\psi(\mathbf{x})) + b \\ &= \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b. \end{aligned}$$

Lagrange multipliers $\{\alpha_i, \alpha_i^*\}_{i=1}^N$ satisfy conditions $\alpha_i, \alpha_i^* \geq 0$, $\alpha_i \alpha_i^* = 0$ and $K(\mathbf{x}, \mathbf{y})$ is a *kernel* function, satisfying Mercer's condition[1], which corresponds to a dot product in feature space Ψ:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{M} c_i \psi_i(\mathbf{x}) \psi_i(\mathbf{y}),$$

where $c_i > 0$ are positive coefficients. In our study we used a translation invariant Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2})$. The coefficients $\{\alpha_i, \alpha_i^*\}_{i=1}^N$ are obtained by maximizing the quadratic form defined as

$$\begin{aligned} R(\boldsymbol{\alpha}) = -\epsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) y_i \\ -\frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}_j)(\alpha_j - \alpha_j^*) \end{aligned}$$

$$\text{subject to} \quad \begin{aligned} &\textstyle\sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0 \\ &\alpha_i, \alpha_i^* \in \langle 0, \tfrac{1}{\beta} \rangle. \end{aligned} \qquad (2)$$

To solve this optimization problem we used a primal-dual interior point method described in [11] and tailored to SVR problems in [10]. The data points $\mathbf{x}_i$ associated with the coefficients α_i, α_i^* satisfying optimizing problem (2) are called *support vectors* (SV).

3 Nonlinear Adaptive Filters

Nonlinear adaptive filters represent a class of filters, which try to overcome the shortcomings of linear adaptive filters dealing with data generated by physical processes with nonlinear behavior. They incorporate some form of nonlinearity into their structure to account for such physical phenomena which are quite common in practice[2]. Moreover, in the case of nonstationarity or quasi-stationarity of the input data optimum filters have to assume a time-varying form to adaptively model an unknown system. There are several types of algorithms for estimating parameters of nonlinear adaptive filters. Normally they use batch mode computation, that means they compute the filter parameters from predefined blocks of observations. In the next sections we will describe two types of nonlinear adaptive filters based on algorithms with a different, on-line or quasi on-line, strategy for adaptation of the structure and parameters of the filters. The parameters and structure of the filters are computed on new data or blocks of data dynamically added during the learning process without needing to recompute filters on all previously available data subsets and thus discarding previous estimates.

3.1 Adaptive Support Vector Regression Filter

Vapnik in [12] proposed a *chunking* method for SV estimation which tries to overcome a well-known difficulty associated with batch training of SVR on large data sets. The *chunking* method is based on the fact that having only SV (i.e. data points corresponding to non-zero α_i, α_i^* coefficients) we will achieve the same solution of the regression problem as with the full data set. The problem is that we do not know the SV set before solving the regression problem. The idea is to compute SV on a smaller subset and add to this subset the training data on which the current regression model makes errors (i.e. data with error larger than ϵ). Then we have to re-train data on the new subset and iterate until all training data satisfies the regression model represented by the selected SV. In [6] a formal proof of the convergence of the proposed *chunking* algorithm is given.

In our ASVR method we have been motivated by the described *chunking* approach. We initialize our ASVR filter by computing SV on first the N_0 input data. In the next step we dynamically introduce new in-

[1] $\int\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} > 0$ for all $g(.) \neq 0$ and $\int_R g^2(\mathbf{x}) d\mathbf{x} < \infty$ [3]

[2] Fundamentally, there are two types of nonlinear adaptive filters, Volterra-based filters and filters based on the artificial neural network approach [1].

put samples and compute outputs of the filter until the error of the L outputs does not exceed the ϵ-insensitive region of Vapnik's loss function (i.e. outputs lie outside the ϵ-tube). Then we simply re-train ASVR filter on a data set which consists of the current SV and input data points corresponding to L "error" outputs. Smaller value of L allows quicker adaptation of the filter but leads to higher computational demands.

A similar ASVR method was introduced in [2] however, the current SV and all data segments on which the average approximation error exceeds a predefined value were used for re-training.

3.2 Resource-Allocating RBF Network

The idea of Resource-Allocating networks (RAN) introduced by Platt [7] is based on iterative center (neuron) allocation and parameters (positions and widths of the centers) and output-layer weights adaptation of Radial Basis Function (RBF) networks during the on-line learning process. The criteria for adding a new center are satisfied if the predefined accuracy[3] of the RAN outputs is exceeded or if the Euclidean distance of input from the nearest center is greater than the predefined critical scale resolution. In [8] the performance of RAN was improved by introducing a recursive least-squares technique based on a Givens QR decomposition (RAN-GQRD) for adaptation of the output weights and by using a pruning strategy for removing existing centers which have an insignificant contribution to decreasing of the overall output error. As was shown in [8] RAN with GQRD and pruning (RAN-P-GQRD) performed with the same prediction accuracy as RAN-GQRD but with no increase in the complexity of the network.

4 Experiments

In the case where only a data set with additive noise is available one possible way to detect a signal of interest is based on prediction of the signal for time-lags beyond the correlation length of the noise. For white noise, a one-step-ahead adaptive linear predictor trained on noisy data will not be able to predict the noise and can only learn to predict the signal of interest [1]. The use of neural predictors with on-line learning will lead to improved signal prediction accuracy, however, without limiting their flexibility the undesirable effect of noise tracking on finite data segments can arise. One possible way to reduce this problem is based on the ability of ASVR filter to tune the trade-off between model complexity and accuracy of approximation.

4.1 Noisy Mackey-Glass data

To compare performance of the proposed filters to "pick-up" clean time-series from noisy data sets we added different types of noise to a generated Mackey-Glass time-series. "Clean" chaotic Mackey-Glass time-series is defined by the differential equation

$$\frac{ds(t)}{dt} = -bs(t) + a\frac{s(t-\tau)}{1+s(t-\tau)^{10}}$$

with $a = 0.2$, $b = 0.1$. We used the first 3103 data points (training part) of the data set available from CMU Learning Benchmark Archive [4].

The filters were then trained to predict the values at time $t+6$ or $t+85$ from inputs at time t, $t-6$, $t-12$, and $t-18$. We added noise with normal (n) or uniform (u) distribution and with levels corresponding to ratios of the standard deviation of the noise and the "clean" Mackey-Glass time-series. We tuned T parameter for RAN-P-GQRD in the range $\langle 30, 40\rangle$, set $\lambda(0) = 0.99$ and used the same parameters as those reported in [8]. For ASVR filter we used the following parameters: $\beta = 0.1$, $\sigma^2 = 0.75$, $N_0 = 50$ and $L = 1$. The ϵ parameter was chosen 0.05 or 0.01. We observed, that pruning SV "older" than 400 time steps did not significantly influence our results for detection of Mackey-Glass time-series, however, computational time was reduced.

4.2 Results

The quality of detecting the signal of interest from learning on a noisy data set was evaluated in terms of the normalized root

[3] Which in our experiments was chosen same as ϵ-insensitive parameter for ASVR filter.

[4] http://www.boltz.cs.cmu.edu/bench.html. This data was generated with $\tau = 17$ and using a second-order Runge-Kutta method with a step size 0.1.

mean squared error (NRMSE) defined as

$$\mathrm{NRMSE}(j) = \sqrt{\frac{\sum_{i=1}^{j}(y_c(i) - \hat{y}_n(i))^2}{\sum_{i=1}^{j}(y_c(i) - \bar{y}_c(i))^2}}$$

$$\bar{y}_c(j) = \frac{1}{j}\sum_{i=1}^{j} y_c(i)$$

where $\hat{y}_n$ represents an estimation of the signal of interest on noisy time-series and y_c the "clean" Mackey-Glass time-series, respectively.

From Figures 1-5 we can see that on the noisy time-series the performance of the ASVR filter in detecting the signal of interest was in terms of the NRMSE approximately 35% better. By using RAN-P-GQRD method we are unable to track signal of interest in noisy environment even with noise of low level (Figure 2). ASVR filter performs well in the case of noise with low level (n=11%, u=10.3%) (Figure 2) but with increasing level of the noise (n=17.7%, u=18.1%) detection abilities are decreased (Figure 1). Using prediction step $t + 6$ (Figures 3 and 4) we achieved similar results but the ASVR filter was able to track the signal of interest also in the case of a higher noise level (n=17.7%, u=18.1%). With $\epsilon = 0.01$ the performance of ASVR filter was increased but performance of RAN-P-GQRD stayed approximately on the same level (Figure 5). In all the cases the convergence of the ASVR filter was significantly faster than convergence of RAN-P-GQRD approach. We did not observed a instability of the proposed ASVR method.

5 Conclusions

In this paper the adaptive support vector regression filter was used for signal detection problems. The filter showed superior performance on the detection of chaotic time-series corrupted by noise when compared to the modified Resource-Allocating RBF network technique.

We introduced a strategy, based on the *chunking* approach, for on-line selection of the data points for re-training of the filter. Although, no proof of the convergence of the method is provided in this paper, we observed that on longer stationary data segments the method can be applicable in practice.

One of the open questions which remains is the appropriate prior choice of the ϵ and β parameters. Recently, in [9] a new ν-SVR algorithm and parametric insensitivity models were proposed for automatic accuracy control. However, in noisy environments *a priori* knowledge of the general signal and noise statistics is still required.

Promising results encourage us to use the ASVR filter for the other problems of nonlinear adaptive filtering which will be studied in our future work.

Acknowledgements The first author is funded by a research grant for the project "Objective Measures of Depth of Anaesthesia"; University of Paisley and Glasgow Western Infirmary NHS trust, and is partially supported by Slovak Grant Agency for Science (grants No. 2/5088/98 and No. 98/5305/468).

References

[1] S. Haykin. *Adaptive Filter Theory.* Prentice-Hall, New Jersey, III. edition, 1996.

[2] D. Mattera, F. Palmieri, and S. Haykin. Adaptive nonlinear filtering with the support vector method. In *Euspico'98 - IX European Signal Processing Conference*, volume II, page 773, 1998.

[3] J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A209:415–446, 1909.

[4] S. Mukherjee, E. Osuna, and F. Girosi. Nonlinear Predicting of Chaotic Time Series Using Support Vector Machines. In *Proc. of IEEE NNSP'97.* Amelia Island, 1997.

[5] K.R. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In W. Gerstner, A.Germond, M.Hasler, and J.-D.Nicoud, editor, *Artificial Neural Networks - ICANN'97*, pages 999–1004. Springer Lecture Notes in Computer Science, Berlin, 1997.

[6] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In J. Principe,

L. Gile, N. Morgan, and E. Wilson, editor, *Neural Networks for Signal Processing VII - Proceedings of the 1997 IEEE Workshop*, pages 276–285, NY, 1997. IEEE.

[7] C.J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[8] R.Rosipal, M. Koska, and I. Farkaš. Prediction of Chaotic Time-Series with a Resource-Allocating RBF Networks. *Neural Processing Letters*, 7:185–197, 1998.

[9] B. Schölkopf, P. Barlett, A. Smola, and R. Williamson. Shrinking the tube: A new support vector regression algorithm. In To appear in: M.S. Kearns, S.A. Soll, and D.A. Cohn, editor, *Advances in NIPS 11*, Cambridge, 1999. MIT Press.

[10] A.J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report NC2-TR-1998-030, NeuroColt2 Technical Report Series, 1998.

[11] R.J. Vanderbei. LOQO: An interior point code for quadratic programming. Technical Report Statistics and Operations Research,TR SOR-94-15, Princeton University, NJ, 1994.

[12] V. Vapnik. *Estimation of Dependencies Based on Empirical Data.* Springer-Verlag, Berlin, 1982.

[13] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

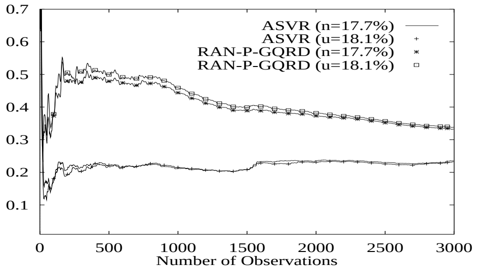

Figure 1: Detection accuracy of RAN-P-GQRD and ASVR in terms of NRMSE ($\epsilon = 0.05$, prediction $t + 85$, n-normal and u-uniform noise, respectively.)

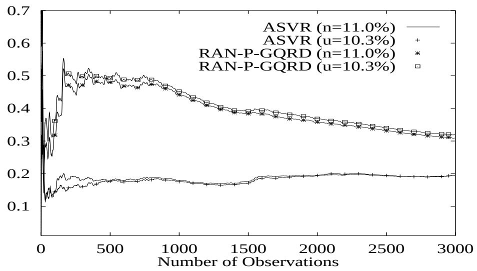

Figure 2: Detection accuracy of RAN-P-GQRD and ASVR in terms of NRMSE ($\epsilon = 0.05$, prediction $t + 85$, n-normal and u-uniform noise, respectively.)

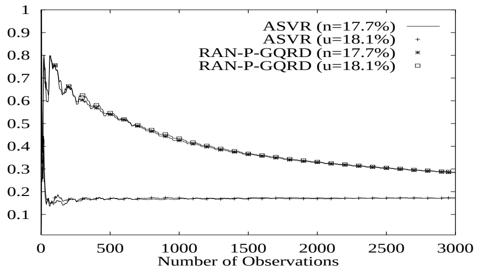

Figure 3: Detection accuracy of RAN-P-GQRD and ASVR in terms of NRMSE ($\epsilon = 0.05$, prediction $t + 6$, n-normal and u-uniform noise, respectively).

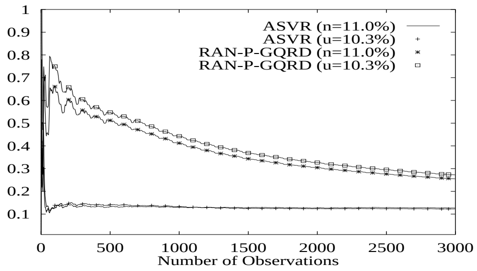

Figure 4: Detection accuracy of RAN-P-GQRD and ASVR in terms of NRMSE ($\epsilon = 0.05$, prediction $t + 6$, n-normal and u-uniform noise, respectively).

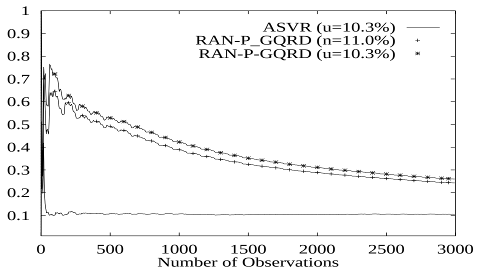

Figure 5: Detection accuracy of RAN-P-GQRD and ASVR in terms of NRMSE ($\epsilon = 0.01$, prediction $t + 6$, n-normal and u-uniform noise, respectively).