RLS Adaptation of One-Class SVM for Time Series Novelty Detection*

Jerónimo Arenas-García, Vanessa Gómez-Verdejo and Ángel Navia-Vázquez
Department of Signal Theory and Communications
Universidad Carlos III de Madrid
Avda. Universidad 30, 28911 Leganés (Madrid), Spain
{jarenas,vanessa,navia}@tsc.uc3m.es

ABSTRACT

Automatic novelty detection techniques have an essential role in important time series processing applications, such as failure detection and audio or speech segmentation. One-class Support Vector Machines (SVMs) have recently been proposed obtaining systematically good results in stationary novelty detection problems. However, their original batch formulation makes them impractical for on-line time series processing. In this paper we use a Weighted Least Squares formulation, alternative to the usual Quadratic Programming approach, from which adaptive one-class SVMs can be explicitly derived. Some advantages of our proposal over previous approaches are that the size of the resulting machine and the forgetting capabilities (exponential window) of the algorithm are under the designer control and can be modified, if necessary, at each iteration.

1 INTRODUCTION

In the last years, there has been an increasing interest in applying machine learning techniques for the extraction of relevant information from time series. The detection of rare or unseen patterns plays an essential role in a variety of time series processing applications, such as failure detection systems (Dasgupta and Forrest, 1999) and audio or speech segmentation (Davy and Godsill, 2002; Gretton and Desobry, 2003). Monitoring systems (for instance, in biomedical applications) can also greatly benefit from these novelty detection techniques, which can be used to find important events that need further human analysis.

When designing an automatic system for time series novelty detection, it is necessary to take two different decisions concerning the selection of a domain for representing the input data and the technology used for the automatic detection system. As discussed in (Keogh et al., 2001), different possibilities exist regarding the representation space (among others: Fourier Transforms, Wavelets, Phase Space and Piecewise Linear Representation), and every kind of changes may be more adequately represented in one space than in the others. For the novelty detection system, one-class Support Vector Machines (SVM) have recently been applied obtaining systematically good results (Davy and Godsill, 2002; Gretton and Desobry, 2003; Ma and Perkins, 2003).

*This work has been partly supported by grant CICYT TIC2002-03713.

SVMs are state-of-the-art tools for input-output knowledge discovery. Originally proposed to tackle binary classification problems, their good performance motivated their extension to multiple-class classification, function regression and novelty detection (Schölkopf et al., 2000), that we consider here. All these problems are usually treated in a batch manner (i.e., a functional is minimized for an a priori given set of training data). However, in on-line time series novelty detection all the data is not known beforehand. For these cases, it is necessary to derive adaptive schemes that modify the proposed solution as more data is available, forgetting also the oldest patterns, that do not longer reflect the current behavior of the system.

In (Gretton and Desobry, 2003) an on-line one-class SVM is proposed, that considers just the last patterns presented to the machine: at each iteration, the exact SVM solution is updated by learning the effects of incorporating to the training dataset the most recent pattern and removing the oldest one. However, the authors of the method assume that it would be interesting to develop systems that use an exponential window to weight the training data, since the estimation and adaptation of the window length is not trivial. The NORMA algorithm (Kivinen et al, 2004) relies in a stochastic gradient minimization of the SVM functional. In addition to gradient noise, this algorithm has the disadvantage that the number of Support Vectors (SVs) increases with time, and so does the complexity of the novelty detector. When ad-hoc heuristics are used to avoid this growing complexity problem, NORMA no longer provides the exact solution to a SVM problem.

In this paper we propose an alternative method for building on-line SVM novelty detectors which is derived from the Iterative Weighted Recursive Least Squares (IW-RLS) algorithm for solving SVMs of (Navia-Vázquez et al., 2001). When IW-RLS is used, the importance of each pattern can be directly modified in the SVM functional, leading to adaptive versions in a straightforward manner. In addition to this, the use of a semiparametric form for the machine provides a direct control mechanism for the maximum permisible complexity of the machine. We will illustrate the performance of the new algorithm through different experiments on four different time series segmentation problems.

2 Novelty Detection with One-Class SVM

Based on Statistical Learning Theory, SVMs were originally proposed to solve the binary classification problem and, due to their excellent performance in most applications, have then been extended for regression and novelty detection problems (one-class SVM, (Schölkopf et al., 2000)). SVMs work by projecting the input data to a very high dimensional space (the so called feature space, $\mathcal{F}$), where a linear solution (non-linear in input space) obtains good performance. Two important reasons for the good performance of SVMs are:

- They do not need to explicitly compute projections on $\mathcal{F}$, but just inner products between vectors (by means of the kernel of $\mathcal{F}$). This allows us to use a very high (or even infinite) dimensional feature spaces.
- SVMs regularize their solution directly in the projection space, what provides them with excellent regularization properties.

Formulation of the one-class SVM problem

In novelty detection we are given a set of l patterns, $D = \{\mathbf{x}_i,\ i = 1 \ldots, l\}$, which are i.i.d. generated from a probability distribution $P(\mathbf{x})$ characterizing the normal behavior of the system. Then, the objective is to build a function $f(\mathbf{x})$ that takes the value $+1$ when $\mathbf{x}$ is a typical pattern, and -1 in any other case. One-class SVMs use the parametric form in $\mathcal{F}$:

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \phi(\mathbf{x}) - 1) \tag{1}$$

where $\phi(\mathbf{x})$ is the projection of pattern $\mathbf{x}$ to feature space and $\mathbf{w}$ are the parameters to be learned during the training phase. The optimization problem proposed by the one-class SVM (Schölkopf et al., 2000) consists on minimizing functional

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i \tag{2}$$

subject to restrictions $\mathbf{w}^T \phi(\mathbf{x}_i) - 1 + \xi_i \geq 0$ and $\xi_i \geq 0$, where the minimization is to be carried out with respect to $\mathbf{w}$ and ξ_i, where the slack variables ξ_i have been introduced to allow that the most unusual patterns in D can be regarded as novel patterns by the automatic system, and C controls the tradeoff between structural (regularization) and empirical risk. The usual procedure for solving the above problem relies on Quadratic Programming (QP). Alternatively, in the following section we offer a solution based on IW-RLS, showing how it can be applied to an adaptive version of (2).

3 RLS Adaptive One-Class SVM

When working with a time series $u(n)$, data is only available as time goes on, what results in a time dependent training dataset for the novelty detection system: $D(n) = \{\mathbf{x}_i,\ i = 1 \ldots, n\}$, where $\mathbf{x}_i$ can correspond to any parameterization of the samples of the time series received up to time i. Correspondingly, we will modify (2) to get an exponentially weighted time-varying functional that reflects the on-line characteristics of the addressed problem

$$L_p(n) = \frac{1}{2}\|\mathbf{w}(n)\|^2 + C \sum_{i=1}^{n} \lambda^{n-i} \xi_i(n) \tag{3}$$

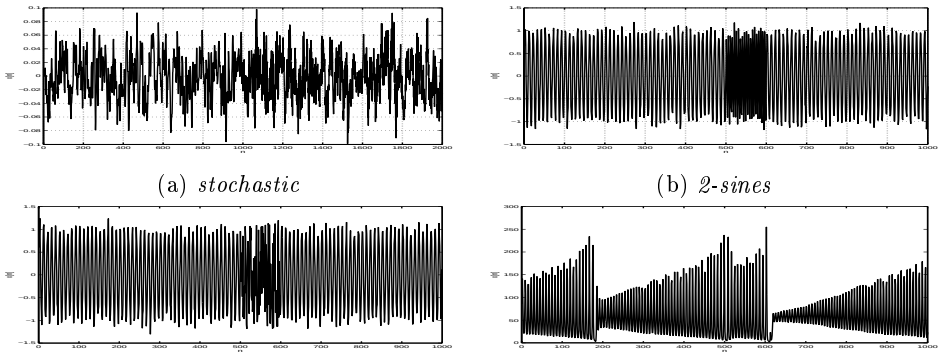
which needs to be minimized with respecto to $\mathbf{w}(n)$ and $\xi_i(n)$ subject to the same restrictions imposed over the minimization of (2), and where the forgetting factor λ is a positive constant, usually very close to 1, that gives less importance to the oldest patterns.

The Representer Theorem (Schölkopf and Smola, 2001) states that the SVM optimal weights can be expressed as a linear combination of some of the traning vectors in feature space (the so-called Support Vectors, SV), $\mathbf{w}(n) = \sum_{i=1}^{n} \alpha_i(n)\phi(\mathbf{x}_i)$, and correspondingly the one-class SVM solution at time $n-1$ becomes

$$f_{n-1}(\mathbf{x}(n)) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i(n-1)\kappa(\mathbf{x}_i, \mathbf{x}(n))\right) \tag{4}$$

where $\kappa(\mathbf{x}_i, \mathbf{x}(n)) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}(n))$, and the problem of increasing size appears. However, it has been shown in (Navia-Vázquez et al., 2001) that on-line algorithms can be constructed when using the the semiparametric form

$$\mathbf{w}_R(n) = \sum_{m=1}^{R} \beta_m(n)\phi(\mathbf{c}_m) \tag{5}$$



(a) *stochastic* (b) *2-sines*

(c) *sine-noise* (d) *laser*

Figure 1: Selected time series for the experimental benchmarking.

for the weights, with $\{\mathbf{c}_m, m = 1, \cdots, R\}$ being a set of centroids that are used to represent (to an arbitrary accuracy) the training data in feature space. Obviously, another important advantage attained by using (5) is that the number of nodes of the resulting machine is fixed to R (a parameter under the designer's control), thus providing a direct mechanism to limit the complexity of the solution and memory requirements of the algorithm. Regarding the selection of these centroids, different procedures can be used, although for on-line algorithms we prefer the heuristic approach in (Parrado-Hernández et al., 2003), which can be applied both to add new centroids as these are needed, or to remove those which become useless. Then, we should come back to the SVM functional (3) and introduce in it the restrictions to get

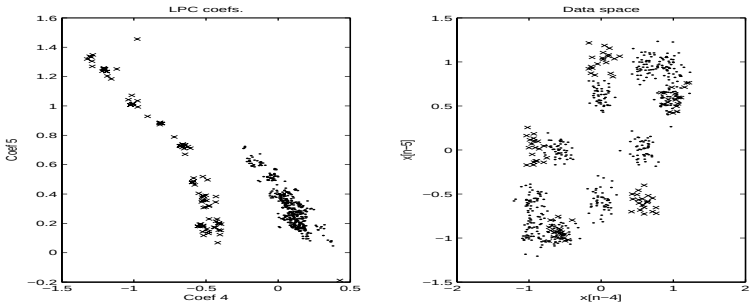
$$L_p(n) = \frac{1}{2}\|\mathbf{w}(n)\|^2 + \sum_{i=1}^{n} \lambda^{n-i} a_i(n-1) e_i^2(n) + \sum_{i=1}^{n} \lambda^{n-i} \xi_i(n) \left(C - \eta_i(n-1) - \alpha_i(n-1)\right) \quad (6)$$

where $\alpha_i(n-1)$ and $\eta_i(n-1)$ are Lagrange Multipliers, and where we have also defined $a_i(n-1) = 2\alpha_i(n-1)/e_i(n)$ and $e_i(n) = 1 - \mathbf{w}^T(n)\phi(\mathbf{x}_i)$. It is easy to show that the last term in the right hand side of (6) vanishes at the solution, and it results that the solution of the functional at time n can be obtained from that at time $n-1$ with a Weighted Least Squares Algorithm, where it is also necessary to assume the semiparametric form (5). Details of such a procedure are well-described in (Navia-Vázquez et al., 2001) and have been omitted here for reasons of space.

4 EXPERIMENTS

We have selected several time series for conducting the experiments (see Fig. 1): *stochastic* represents a stochastic process with an abrupt transition at $n = 1000$; *2-sines* corresponds to a sinusoidal signal changing its frequency at $n = 500$ and $n = 600$; *sine-noise* is the same kind of signal, but with uniform noise replacing sinusoid samples between $n = 500$ and $n = 600$; and *laser* (Ma and Perkins, 2003) is the output of a laser device, with transitions approximately at $n = 185$, $n = 500$ and $n = 620$.

Before applying the one-class adaptive SVM algorithm, we have selected the type of preprocessing more appropriate for each time series. It is very common to directly operate in the sample space, i.e., using a finite window containing the last samples. However, in



(a) LPC coefficients (b) Sample space

Figure 2: Two different representational spaces for the *2-sines* time series.

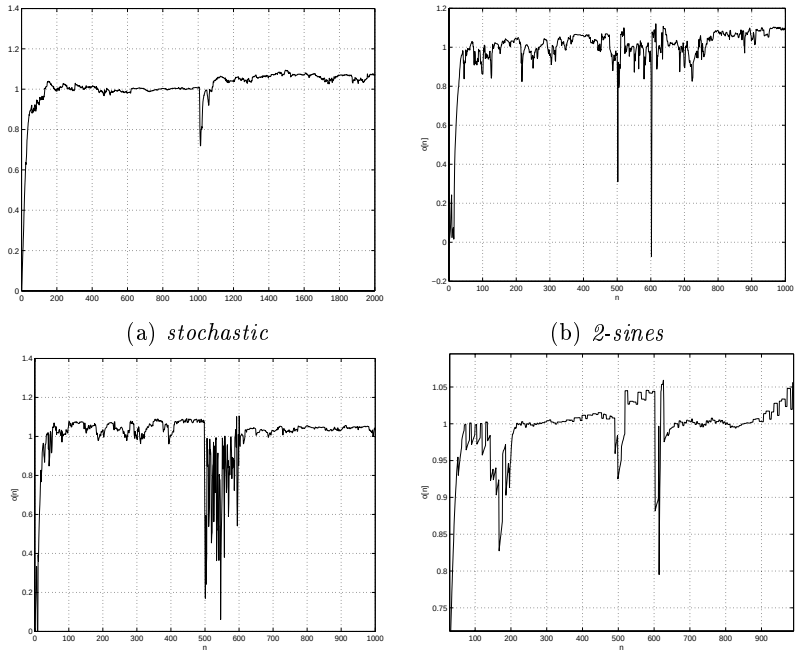
certain cases, the most valuable information is not easily accesible in such space, as can be seen in Fig. 2 (where we have represented with different symbols the two portions of the signal to be segmented) for *2-sines*, with representations in sample space (b) and using Linear Prediction Coefficients (LPC) (a).

Therefore, we have opted for a different preprocessing for each time series, namely, an LPC estimation, for *stochastic*, *2-sines*, and *sine-noise*, and an envelope detector (which retains maximum and minimum values of the last k samples of the time series) for *laser*. We have trained an one-class SVM (using Gaussian kernels) for each of the problems using these preprocessed data. In Fig. 3 we represent the outputs of each model, where 'normal' values around 1 represent 'steady model' and therefore non-novel, while minima in the curves represent points of detected novelty.

It can be observed how the transition between stochastic processes is correctly detected (a), as well as the transition between sinusoids with different frequencies (b), and the adaptive one-class SVM rapidly adapts to the new situation until a new change in the series is observed. The SVM model for *sine-noise* 'marks' the whole noisy interval $500 < n < 600$ as 'novel', since there is not an appropriate model for that part of the series and the one-class machine is always adapting itself during that period. Finally, the *laser* time series has also been correctly segmented using our approach.

5 Conclusions and Future research

We have presented an on-line version of one-class SVMs, relying on robust RLS optimization techniques of a time-varying SVM functional. The proposed algorithm has the advantage that the memory of the algorithm and the size of the machine are under the designer's control, and can be changed during operation if necessary. Furthermore, the RLS optimization process is less computationally demanding than QP. The performance of the algorithm has been tested in a segmentation task using four different time series, correctly detecting all transitions. However, in these preliminary experiments we selected the tunable parameters (forgetting factors, kernel parameters, etc) upon a trial and error basis. As a further research, we aim at developing semi-automatic mechanisms for adjusting such parameters. We would also like to obtain some criteria for optimal preprocessing selection based on the time series properties.



(a) *stochastic* (b) *2-sines*

(c) *sine-noise* (d) *laser*

Figure 3: Output of the adaptive one-class SVM model: minima in the output value represent a novelty detection.

References

Dasgupta, D. and Forrest, S. (1999). Novelty detection in time series data using ideas from immunology. In *Proc. of the Intl. Conf. on Intelligent Systems*, Reno, Nevada.

Davy, M. and Godsill, S. (2002). Detection of abrupt spectral changes using support vector machines: an application to audio signal segmentation. In *Proc. of ICASSP02*, Orlando, FL, vol. 2, 1313-1316.

Gretton, A. and Desobry, F. (2003). On-line one-class support vector machines. An application to signal segmentation. In *Proc. of ICASSP03*, Hong Kong, vol. 2, 709-712.

Keogh, E., Chu, S., Hart, D., and Pazzani, M. (2001). An online algorithm for segmenting time series. In *Proc. of the 2001 IEEE Intl. Conf. on Data Mining*, San Jose, CA, 289-296.

Ma, J. and Perkins, S. (2003). Time-series novelty detection using one-class support vector machines. In *Proc. of IJCNN 2003*, Portland, OR, 1741-1745.

Schölkopf, B., Williamson, R., Smola, A. J., Shawe-Taylor, J., and Platt, J. (2000). Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, vol. 12, Cambridge, MA, MIT Press.

Kivinen, J., Smola, A. J., and Williamson, R. C. (2004). Online learning with kernels. *IEEE Trans. on Signal Proc.*, 52(8):2165-2176.

Navia-Vázquez, A., Pérez-Cruz, F., Artés-Rodríguez, A., and Figueiras-Vidal, A. R. (2001). Weighted least squares training of support vector classifiers leading to compact and adaptive schemes. *IEEE Trans. on Neural Networks*, 12(5):1047-1059.

Schölkopf, B. and Smola, A. J.(2001). *Learning with kernels.* MIT Press.

Parrado-Hernández, E., Mora-Jiménez, I., Arenas-García, J., Figueiras-Vidal, A. R., and Navia-Vázquez, A. (2003). Growing support vector classifiers with controlled complexity. *Pattern Recognition*, 36(7):1479-1488.