Expert Systems with Applications 40 (2013) 7242–7249

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Abrupt change detection with One-Class Time-Adaptive Support Vector Machines

Guillermo L. Grinblat, Lucas C. Uzal, Pablo M. Granitto *

CIFASIS – French Argentine International Center for Information and Systems Sciences, AMU (France)/ UNR–CONICET (Argentina), Bv. 27 de Febrero 210 bis, Rosario S2000EZP, Argentina

ARTICLE INFO

Keywords:
Abrupt change detection
One class classification
Support vector machine

ABSTRACT

We recently introduced an algorithm for training a sequence of coupled Support Vector Machines which shows promising results in the field of non-stationary classification problems Grinblat, Uzal, Ceccatto, and Granitto (2011). In this paper we analyze its application to the abrupt change detection problem. With this goal, we first introduce and analyze an extension of it to deal with the One-Class Support Vector Machine (OC-SVM) problem, and then discuss its use as an improved abrupt change detection method. Finally, we apply the proposed procedure to artificial and real-world examples, and demonstrate that it is competitive by comparison against other abrupt change detection methods.

1. Introduction

In data modeling, an abrupt change is usually defined as a noticeable variation in the distribution that generates the data, produced on a very short period of time (Basseville & Nikiforov, 1993). Abrupt changes can be easily spotted in some cases, for example when there is a big increase in the variance of the noise of a time-series, or hard to find in others, as in a subtle change in the vibration pattern of a complex machine. The problem has many real world applications, including time series analysis (Blythe, von Bunau, Meinecke, & Muller, 2012), spam filtering (Hsiao & Chang, 2008), modeling of data streams (Tsai, Lee, & Yang, 2009), or some industrial processes (Petrovic, Jakovljevic, & Milacic, 2010).

Among the methods aimed at abrupt change detection, a well-known strategy is to measure an appropriate property of the data over short periods of time (time windows) and to associate a change in the value of this property with an abrupt change in the data. Simple methods use a given statistic of the data (Basseville & Nikiforov, 1993), while more complex ones involve the analysis of the parameters of statistical or machine learning models fitted to the windowed data set (Desobry, Davy, & Doncarli, 2005). One of the drawbacks of this approach is that the fitting of complex models over small datasets usually produces poorly regularized solutions that lead to a high false positive detection rate.

In particular, One-Class classifiers are appropriate models for this analysis. In One-Class problems the goal is to describe a single class of objects and distinguish it from all other possible objects (usually outliers). One of the most successful and effective methods in this area is the One-Class SVM (OC-SVM) (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001), which has been widely applied, including for example document classification (Manevitz & Yousef, 2001), intrusion detection (Giacinto, Perdisci, Rio, & Roli, 2008), Bioinformatics (Wu, Gan, & Jiang, 2011) and fault detection (Camci & Chinnam, 2008).

It is also well known that in the context of non-stationary problems it is useful to consider methods that do not assume that the data samples are identically distributed, thereby able to adapt themselves to the non-stationary situation. In this work we first introduce an extension of the Time-Adaptive Support Vector Machines (TA-SVM) (Grinblat et al., 2011) to One-Class problems. Our new method uses a series of coupled OC-SVMs in order to learn efficiently in slowly changing environments. It is based on individual, adaptive models which are fitted on short segments of the full time interval, all learned simultaneously (in a global way) using a coupling term that forces neighbor models to be similar, introducing a regularization effect on the sequence of models. A non-stationary version of OC-SVM has been proposed by Camci and Chinnam (2008). It assigns different (lower) weights to data items as a function of their "age", obtaining a model that is fitted to the current time. While this can be used to generate a sequence of models capturing the evolution of the data, we would need to train each model separately. The method introduced in this paper can fit the whole sequence with a similar cost to one OC-SVM.

The second novelty of this work is that we show how this coupled sequence of OC-SVMs can be used to detect abrupt changes in the data (using a single user-selected threshold as other methods, such as the Kernel Change Detection (KCD) algorithm (Desobry et al., 2005)). The resulting method has several benefits. In the first

* Corresponding author. Tel.: +54 3414237248.
E-mail addresses: grinblat@cifasis-conicet.gob.ar (G.L. Grinblat), uzal@cifasis-conicet.gob.ar (L.C. Uzal), granitto@cifasis-conicet.gob.ar (P.M. Granitto).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.074

place, the user does not need a priori to assume a particular data distribution and the method can be applied to problems with high dimensionality, given that it is based on OC-SVM. It shares this characteristic with KCD. In the second place, it is more robust to noise than KCD, since it is based on TA-SVM while KCD uses sliding windows. This characteristic was already observed for TA-SVM (Grinblat et al. (2011)). In 4 and the following sections we show the potential of our new method with artificial and real world applications, comparing the detection capabilities against two previous methods. We close this paper drawing some conclusions in Section 6.

2. Previous work

Given a data set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ sampled independently from an unknown distribution $\mathbf{P}$, the objective of the OC-SVM is to find a region of the input space where the sampling probability is high. Schölkopf et al. (2001) achieve this by using a kernel — usually a Gaussian kernel — to map the data points to a feature space where they are separated from the origin with a plane (defined by a vector $\mathbf{w}$ and a scalar ρ) with maximum margin.

This can be done by solving the following problem:

$$\min_{\mathbf{w},\rho,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho \tag{1}$$
$$\text{subject to} \quad \mathbf{w}\cdot\Phi(\mathbf{x}_i) \geqslant \rho - \xi_i; \quad \xi_i \geqslant 0,$$

where $\|\mathbf{x}\|$ is the norm of vector $\mathbf{x}$, ξ_i are the slack variables, ν is a parameter selected by the user and Φ a mapping from the input to the feature space (Schölkopf et al., 2001). It is worth noting that Tax and Duin (2004) presented a similar problem, reaching an equivalent solution.

In order to learn classification tasks in slowly changing environments, a new method based in SVMs was recently proposed Grinblat et al. (2011), the TA-SVM. It consists of fitting SVMs on short segments of the full time interval, which are all learned simultaneously (in a global way) using a coupling term that forces neighboring models to be similar. It assumes that each sample $(\mathbf{x}_i,\ y_i)$ of the data set $\{(\mathbf{x}_1,\ y_1), \ldots, (\mathbf{x}_n,\ y_n)\}$ was obtained at time i, i.e., they are time-ordered, and that the relationship between $\mathbf{x}$ and y changes slowly over time. It then divides the data into m consecutive, non-overlapping time windows and creates a coupled sequence of m classifiers, each one optimal in its corresponding time window. As the problem follows a slow evolution, each classifier should be similar to its temporal neighbors. To this end, TA-SVM searches a solution which is a trade-off between (individual) optimality and (neighbor) similarity. Given a distance measure $\mathrm{d}(c_\mu, c_\nu)$ that quantifies the diversity between two neighboring models f_μ and f_ν, this is obtained as the solution of the problem

$$\min \quad \frac{1}{m}\sum_{\mu=1}^{m} Err_\mu + \frac{\gamma}{m-1}\sum_{\mu=1}^{m-1}\mathrm{d}(f_\mu, f_{\mu+1}), \tag{2}$$

where the first term is the average of the usual cost function for each of the m classifiers and the second evaluates the total difference among the sequence of modeling functions. The free parameter γ regulates the compromise between both terms, as in any usual regularized fitting.

The parameter m indicates the detail desired in the modeling of the sequence. With a small value for m, only a few hyperplanes will be actually used in the model, each one describing the sequence in a long period of time. With a big value for m (including $m = n$) each hyperplane will describe only a short time period of the sequence. As shown in Grinblat et al. (2011), it is advisable to use $m = n$ when the data under analysis has a low noise level, while with a higher noise level it may be useful to use lower values. For the abrupt change detection method proposed in Section 4 it is convenient to maintain $m = n$, since this allows the detection of a change in any point of the sequence. A lower value would make the method detect changes only in a fraction of the points.

To quantify the diversity between the classifiers, a quadratic distance is used:

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

In a recent work, Camci and Chinnam (2008) introduced a non-stationary version of the OC-SVM, the General Support Vector Representation Machine (GSVRM). The method proposes to solve the same problem as in the standard OC-SVM, but assigning different (lower) weights to data points as a function of their "age".

Regarding the abrupt change detection problem, the solutions proposed can be divided in parametric and nonparametric. Amongst the first type, one of the most important and widely used is the CUSUM algorithm (Montgomery, 2009). Unfortunately, as with all parametric methods, it needs information about the data that is generally unavailable in real problems.

One of the state-of-the-art tools to tackle this problem is the Product Partition Model (PPM) originally introduced by Hartigan (1990), Loschi, Cruz, Iglesias, and Arellano-Valle (2003), Quintana (2006) and Müller and Quintana (2010). This method casts the number of changes and their positions as random variables, and for this reason does not need the number of changes to be specified by the user. It also gives not just the most probable partition, but several partitions with their probabilities. The main drawback is that the type of the data distribution must be known a priori D'Angelo, Palhares, Takahashi, and Loschi (2011).

If there is no previous information available about the sequence to be analyzed, another nonparametric method can be used. A relevant example, based in OC-SVMs, is the KCD algorithm (Desobry et al., 2005). It detects abrupt changes by fitting two One-Class SVMs for each point t in the sequence, one in the immediate past subset (a time window that ends at point t) and one in the immediate future subset (a time window that starts at point $t+1$). It then calculates a dissimilarity measure between the two results, and if it is greater than a user specified threshold, it decides that there is an abrupt change between points t and $t+1$. KCD is mainly used in audio problems. Successful applications include music segmentation (Desobry et al., 2005; Gillet, 2007), classification of impulsive sounds (Rabaoui, 2007), and speaker diarization (Fergani, Davy, & Houacine, 2008).

Another successful application field of nonparametric methods is fault detection in power systems. In Ukil and Zivanovic (2006) and Ukil and Zivanovic (2007) the authors present a method based on wavelets which shows a good performance for this type of problems.

3. One-Class TA-SVM

In this section we extend the TA-SVM method to deal with a non-stationary version of the One-Class problem, that is to find a region of the input space where most of the data points can be found, taking into account that the input distribution, and thus this small region, may vary slowly with time. We will call this extension One-Class Time-Adaptive Support Vector Machine (OC-TA-SVM).

To this end, we combine Eq. 2 with the problem introduced by Schölkopf et al. (2001) Eq. 1. As a result we obtain the primal version of the problem to be minimized:

$$\min_{\mathbf{w}_\mu,\rho_\mu,\xi} \quad \frac{1}{m}\left(\frac{1}{2}\sum_{\mu=1}^{m}\|\mathbf{w}_\mu\|^2 - \rho_\mu\right) + C\sum_{i=1}^{n}\xi_i + \frac{\gamma}{m-1}\sum_{\mu=1}^{m-1}\mathrm{d}(f_\mu, f_{\mu+1}) \tag{3}$$
$$\text{subject to} \quad \xi_i \geqslant 0;\ \mathbf{w}_{\mu_i}\cdot\mathbf{x}_i \geqslant \rho_\mu - \xi_i,$$

where the distance function d is defined over hyperplanes characterized by $\mathbf{w}$ and ρ:

$$d(f_v, f_\mu) = \|\mathbf{w}_v - \mathbf{w}_\mu\|^2 + (\rho_v - \rho_\mu)^2. \quad (4)$$

It is worth noting that the cost function in Eq. 1 depends on the norm of $\mathbf{w}$ and the scalar ρ. Problem 3, applying a factor $\frac{1}{2}$ to the sum of the norms of $\mathbf{w}_\mu$, maintains the relationship found in the original problem between $\mathbf{w}$ y ρ.

This problem shares some characteristics with the original TA-SVM method. The parameter γ regulates how strong the coupling will be along the sequence of models. A low value will almost decouple the sequence, while large ones will produce a sequence of almost identical models. The formulation is also valid when considering time windows including one point ($m = n$), as in the original case (Grinblat et al., 2011).

Following the same derivation for TA-SVM (Grinblat et al., 2011, Appendix A), it can be seen that the problem in (3) can be rephrased in terms of its corresponding dual as:

$$\max_{\boldsymbol{\alpha}} \quad -\frac{1}{2}\boldsymbol{\alpha}^T \mathbf{R} \boldsymbol{\alpha},$$
$$\text{subject to} \quad 0 \leqslant \alpha_i \leqslant \frac{1}{vn}; \quad \sum \alpha_i = 1.$$

for an appropriate matrix $\mathbf{R}$.

The time complexity of OC-TA-SVM can be analyzed in two stages: the computation of matrix $\mathbf{R}$, and the solution of the optimization problem. The matrix can be computed in $O(n^2)$ (Grinblat et al., 2011). Once this matrix is calculated, the optimization problem is a conventional SVM problem, which is between $O(n^2)$ and $O(n^3)$. From this we can conclude that the time complexity of an OC-TA-SVM problem is roughly the same as an SVM one.

3.1. Artificial example

We first study the behavior of the new method using an artificial data set sampled from a distribution in a two-dimensional space that changes slowly with time. Each data point (in polar coordinates) has a radius taken from a normal distribution with mean 1 and standard deviation 0.05. Its angle is taken from a uniform distribution in the interval $\left[2\pi\frac{t}{500} - \frac{\pi}{2} + 0.1; 2\pi\frac{t}{500} + \frac{\pi}{2} - 0.1\right]$, where t is the timestamp of each point. The data set has 500 points (that means $t \in 1 \ldots 500$), expressed in Cartesian coordinates.

As it can be seen from previous works (Schölkopf et al., 2001; Tax & Duin, 2004), the solution obtained by OC-SVM is highly dependent on the values selected for the free parameters. The same is valid for the non-stationary version of the method, the GSVRM (Camci & Chinnam, 2008).

In this first experiment, we train a OC-TA-SVM with a Gaussian kernel. This kernel is particularly useful to the OC-SVM problem, showing better results than other kernels (Tax & Duin, 2004). Moreover, a relationship between OC-SVM and Density Estimation can be established when this kernel is used (Muñoz & Morguerza, 2004).

In this and the following experiments, the method proposed was implemented using LibSVM (Chang & Lin, 2011) and R (Team, 2012).

In Fig. 1 we show the solutions obtained for different values of γ while maintaining the remaining free parameters constant. Fig. 2 presents the results corresponding to different values of σ, the Gaussian kernel width.

In these figures we can see that, as expected, high values of γ (dashed and dashed-and-dotted lines) result in an overgrown region, because of the influence of points that are too far away in time. On the other hand, with a low value of γ (dashed and double point line) the distribution cannot be modeled correctly. For values between these extremes (thick line) we observe satisfactory solutions. Similar results are obtained by varying the σ parameter. The main difference is that in this case the region does not grow or shrink following the trend in the data — it just becomes smoother or bumpier.

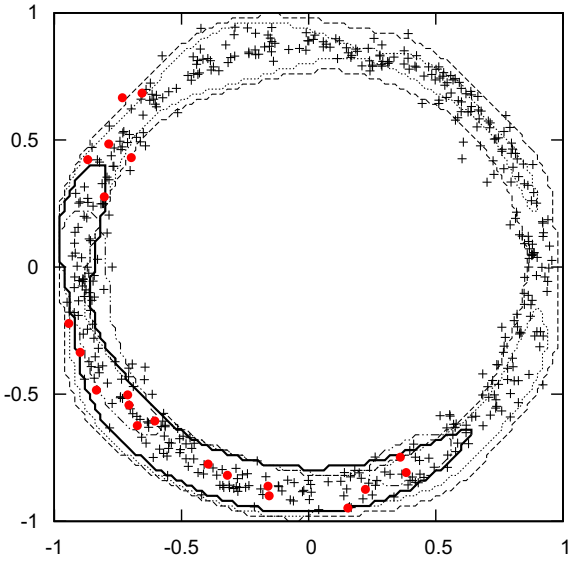

Fig. 1. Selected area by OC-TA-SVM for time $t = 200$ for various values of γ. Big values of γ are shown with dotted and dashed lines. A low value is shown with a dashed and double point line, and an approximately optimal value with a thick line. The points from $t = 190$ to $t = 210$ are shown in red. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

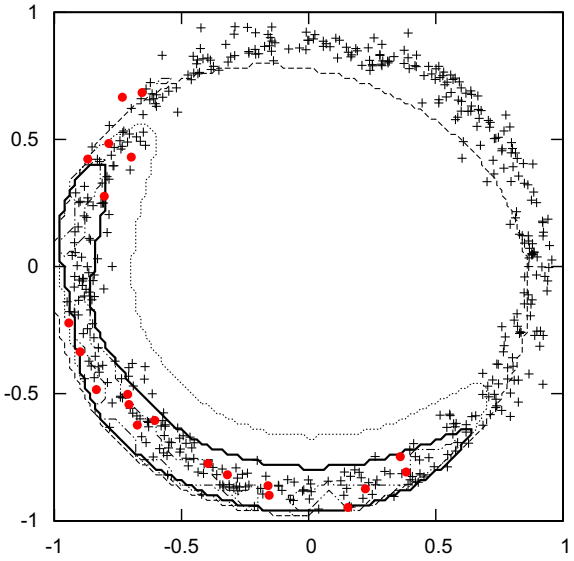

Fig. 2. Selected area by OC-TA-SVM for time $t = 200$ for various values of σ. Big values of σ are shown with dotted and dashed lines. A low value is shown with a dashed and double point line, and an approximate optimal value with a thick line. The points from $t = 190$ to $t = 210$ are indicated in red. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

In a second experiment we compared the performance of OC-TA-SVM with OC-SVMs trained in a sliding window and with the GSVRM. This last algorithm was designed to obtain a current model in the case of non-stationary data. We can apply it to the case at hand with a simple modification: we generate a sequence by

Table 1
Mean error over 30 independent trials of the three methods. In parenthesis we show the standard deviation of this mean.

Method	Error
OC-SVMs in sliding windows	0.30 (0.01)
OC-TA-SVM	0.16 (0.01)
GSVRM	0.22 (0.01)

running the algorithm once for each point in the data set and weighting exponentially past and future points.

For each method we selected free parameters values that minimize the generalization error. According to Tax and Duin, for the original OC-SVM case, the error comes from two sources: patterns that are rejected even though they belong to the concept, and patterns that do not belong but are nevertheless accepted (Tax & Duin, 2004). This also applies to the non-stationary case.

The first source can be estimated with cross-validation. The second source is more difficult, since we do not have samples of this case. What can be done is to select the free parameters which, while maintaining a small error of the first class, result in a smaller region. Similar policies have been used before for OC-SVM (Hayton et al., 2007).

In this experiment we fixed the ν parameter to 0.01 in order to have very few points outside the region modeled by the algorithm. A grid was used for the other two parameters (σ and the window length for OC-SVM in sliding windows, σ and γ for OC-TA-SVM and σ and γ for GSVRM). We discarded the combinations of parameters which generate solutions that do not show similar training and validation errors (with a tolerance of up to 5% difference), and from the remaining pairs we selected the one which generates the solution with the smaller region. This was measured by classifying an external data set made of a uniform grid in a squared region of $\mathbf{R}^2$ that contains the original data set.

In Table 1 we show the mean error obtained over a grid of points used as a test data set, with its standard deviation, over 30 independent runs of this experiment. These are *Macro Average Errors*, i. e. $MAE = \frac{E_1}{N_1} + \frac{E_2}{N_2}$ where E_i is the number of errors produced on class i points (we have two classes: points that belong to the concept present in the training data set and points that do not). N_i is the number of points in class i.

4. An application: abrupt change detection

In this and the next section we explore the use of OC-TA-SVM in the problem of abrupt change detection. We base our proposal in a dissimilarity measure between previous and posterior models of the distribution for a time instant t.

Given that a correctly fitted OC-TA-SVM provides a sequence of these models, we can use it to measure the distance d from Eq. 4 between adjacent models. Providing an a priori fixed threshold τ, an abrupt change would be detected between models v and $v+1$ if and only if:

$$\mathrm{d}(h_v, h_{v+1}) = \|\mathbf{w}_v - \mathbf{w}_{v+1}\|^2 + (\rho_v - \rho_{v+1})^2 > \tau.$$

The threshold τ fixes the sensibility of the approach. A low value makes it more sensitive, increasing the probability of false positives, while with a high value actual changes in the data may pass undetected. Also, the distance between adjacent models may become higher because of the presence of noise in the data, thus demanding a higher threshold to avoid false positives. In this work we propose, following Desobry et al. (2005), to select the threshold with the other parameters of the method by means of a supervised optimization over a small sample of the considered sequence, where the time of each abrupt change is known. In particular, in the following experiments we took the mean of the maximum values of the index in small intervals around the known changes, and the mean of the index in the rest of the points (without known changes). The threshold was then fixed at the middle point between these two mean values.

Since the sequence obtained with TA-SVM is generally less noisy than the one obtained with SVMs trained on sliding windows (see Grinblat et al., 2011, Section 3B), it is expected that this will lead to a more robust index than the ones using sliding windows, such as KCD.

4.1. Time complexity

Given the model sequence obtained with OC-TA-SVM, the index proposed requires just to calculate the distance between neighboring models. Each one of these are, in the OC-TA-SVM solution, a weighted sum of the images of the support vectors in the feature space. This implies a computation of order $O(n_{SV})$, where n_{SV} is the number of support vectors, for each pair of neighboring models which, counting all neighbors in the sequence, gives a total computation of order $O(nn_{SV})$. Thus, the time complexity of calculating the index is basically the same as obtaining the OC-TA-SVM solution.

Since this computational cost can be excessive for very large data sets, one approach to ameliorate it is to divide the original large sequence into smaller pieces. We calculate the index in each of these pieces and append them as a final stage. In this way, if we divide the original sequence of n points into p pieces, we have p problems with the complexity corresponding to n/p points.

This simple technique can bring two problems. First, how can we determine the size of the smaller pieces? Second, how can we compensate the border effects that arise in this case at the edge of each piece?

For the first problem, we can use a long enough time window. This is not the typical problem with time windows, where a long one could be harmful. In this case, a long time window just makes the algorithm to consume more time, but does not make it less accurate. Hence, the size of the pieces should be large enough as to only discard the points with negligible influence, given the coupling parameter γ. The second problem can be solved by just discarding the border values and overlapping the small pieces. In the real example shown in the next section, we applied this policy using batches of 500 points and discarding the first and last 125 of each one. The γ used is such that the influence of a point 125 steps away can be disregarded. As a result, the index obtained is almost the same as would be obtained if the whole sequence were used, but in a fraction of the time.

4.2. Illustrative example

We began by performing a simple experiment to verify the usefulness of the method. We trained an OC-TA-SVM with a two-dimensional data set made of 1,000 samples taken from two Gaussian distributions that change their centers abruptly. More specifically, the first component of each data point is taken with equal probability from $\mathbf{N}(x,1)$ or $\mathbf{N}(-x,1)$, with $x = 0, 2, 4, 6, 8$ in the intervals 1 to 200, 201 to 400, 401 to 600, 601 to 800 and 801 to 1000 respectively. The second component is taken from $\mathbf{N}(0,1)$ for the whole data set. It is worth noting that, even though the mean of each Gaussian changes with time, the mean of the whole data set does not.

Fig. 3 shows the index obtained — the average over 30 trials with its standard deviation — by an OC-TA-SVM trained with free parameters $\nu = 0.1, g = 0.1$ and $\gamma = 10^4$. It can be seen that, as suspected, the maximum distance between adjacent hyperplanes appears in the time instants where there is an abrupt change.

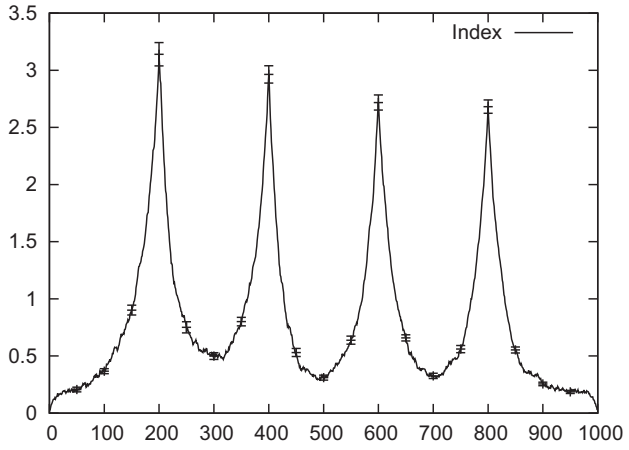

Fig. 3. Distance between adjacent hyperplanes. The line shows the average value over 30 trials while the bars show their standard error.

5. Empirical evaluation

In this section we apply the proposed method to two artificial and one real problem and compare it with KCD and PPM to evaluate its performance. In the three considered problems the free parameters were tuned using an independent data set. In the artificial tests we conducted 100 trials of the experiment, and here report the mean results of these trials. It was also compared with a widely used multivariate version of CUSUM first introduced by Pignatiello and Runger (1990), Bersimis, Psarakis, and Panaretos (2007) and Golosnoy, Ragulin, and Schmid (2009).

For PPM we used the version of Loschi et al. (2003), based on Gibbs sampling, whose code is publicly available from the authors. In that work the authors assume that the sequence to analyze is univariate, and that the data between consecutive changes have a normal distribution with parameters σy μ having a conjugate normal-inverted-gamma prior distribution. For more details refer to Loschi et al. (2003).

In order to apply PPM to the following examples, their version was extended to the multivariate case, maintaining the assumption of data normality. That is, in a d-dimensional space we assume that the data have a normal distribution between consecutive changes with parameters $\sigma_1, \ldots, \sigma_d$ and $\mu_1, \ldots, \mu_d$, where each pair σ_i, μ_i has a conjugate normal-inverted-gamma prior distribution.

5.1. Change detection under noise

In the first artificial experiment we explore the performance of the new method in noisy situations. As stated before, the sequence of hyperplanes obtained with TA-SVM is less noisy than the one obtained with sliding windows. Because of this, we expect that an abrupt change detector based on OC-TA-SVM will be more resistant to noise than one based on sliding windows.

To study this situation we designed the next problem. We have 500 sequences of 500 two-dimensional data points. Half of the sequences does not present any changes and the other half has an abrupt change in the middle of the sequence (point 250). To add noise, each point of each sequence is replaced, with probability p, with points taken from a uniform distribution in a box containing the data. The experiment was repeated for different values of $p : 0, 0.1, 0.2$ and 0.3.

We repeated the experiment with different distributions for the points in the sequences with no changes and the first half of the sequences with changes (cluster I), and for the points in the second half of the latter ones (cluster II). In the first case, the components of each point of cluster I are taken from $\mathbf{N}(0, \frac{1}{3})$, while for cluster II the first component of each point is taken with equal probability from $\mathbf{N}(3, \frac{1}{3})$ or from $\mathbf{N}(-3, \frac{1}{3})$. In the second case, the standard deviations of the Gaussians were changed, using $\mathbf{N}(0, \frac{1}{2})$ for cluster I and $\mathbf{N}(\pm 3, \frac{1}{4})$ for cluster II. In the third one, the standard deviations were even bigger, using $\sigma = 3$ for cluster I and $\sigma = 1.5$ for cluster II. These are similar to the distance between the Gaussians. In the last case the change is present only in the standard deviation. Each component of the points of cluster I are taken from $\mathbf{N}(0, 1)$ and for the points of cluster II from $\mathbf{N}(0, \frac{1}{2})$.

For each method we counted how many of the 500 sequences were evaluated correctly. In this case, correct means not detecting any changes in the 250 sequences with no changes, and, in the other 250 sequences, detecting one change only in a small interval around the real change and no false alarms outside that interval.

To tune the free parameters we used a sequence with an abrupt change (constructed in the same way as before). The free parameters selected were the ones that maximized the distance of the maximum value obtained in an interval around the change time and the mean value of the rest of the sequence, scaled with the standard deviation of this value.

As PPM directly returns the sequence of change points (with its probability) the parameters selected for this method were the ones that maximized the probability of a correct evaluation of this

Table 2
Number of sequences evaluated correctly by each method. The results are the average over 100 trials, with the standard deviation of this mean in parenthesis.

Dataset	Noise(%)	OC-TA-SVM	KCD	PPM
1	0	499.5 (0.1)	480.2 (4.6)	476.0 (2.5)
	10	499.8 (0.1)	465.9 (3.2)	344.4 (3.5)
	20	497.2 (0.4)	410.4 (7.2)	335.3 (3.6)
	30	483.2 (1.9)	375.7 (9.5)	330.8 (5.4)
2	0	497.3 (0.4)	483.3 (5.1)	436.3 (4.8)
	10	499.6 (0.1)	443.2 (6.9)	352.7 (4.1)
	20	495.8 (0.4)	419.8 (7.4)	341.8 (4.1)
	30	477.2 (2.2)	402.6 (6.7)	337.2 (3.9)
3	0	189 (13)	87 (10)	384.8 (3.1)
	10	142 (12)	88 (10)	259.9 (2.3)
	20	88 (10)	71.7 (8.3)	241.7 (1.9)
	30	43.3 (7.4)	61.9 (7.8)	245.1 (0.6)
4	0	243 (13)	151 (11)	411.0 (4.9)
	10	198 (13)	193 (11)	316.1 (6.4)
	20	181 (11)	168.8 (9.8)	237.5 (5.1)
	30	129 (11)	134 (11)	227.2 (4.8)

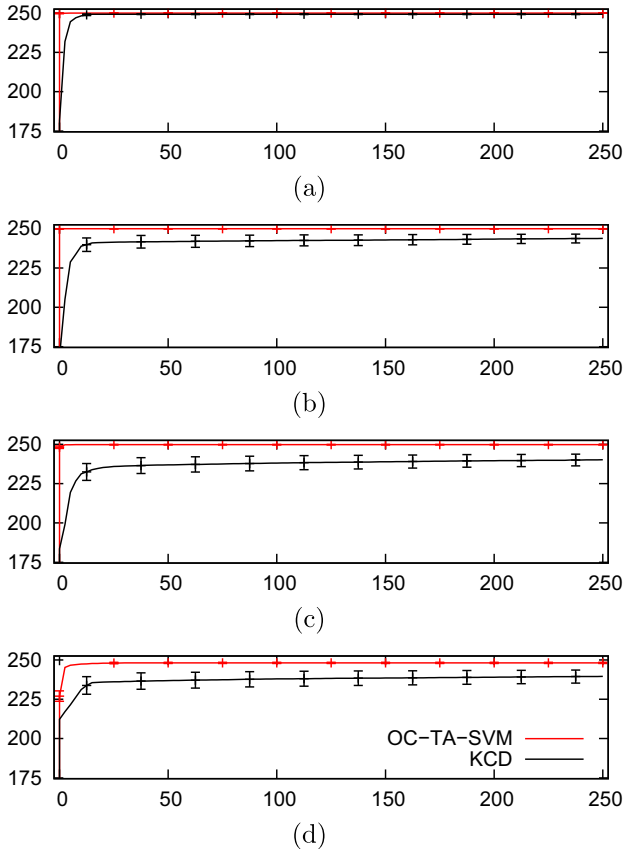

Fig. 4. Number of correctly detected changes as a function of wrongly detected ones, for the no noise (a), 10% (b), 20% (c) and 30% noise cases (d). Bars indicate standard error.

sequence (taking into account the different sequence of change points and probabilities returned by the method).

In Table 2 we show the results for OC-TA-SVM, KCD and PPM for different noise levels. We do not include CUSUM given that we were unable to find a set of operating parameter values that could evaluate correctly the train sequence (in any of the 100 trials).

From these results we observe that the proposed method is superior to KCD (excluding the cases where both methods give bad results). We can also see that both methods provide better results than PPM when the data distribution is far from the assumption made for it. When this is not the case (datasets 3 and 4), i.e. when we possess prior information about the data distribution, then PPM is the method of choice.

In order to corroborate that the observed difference between the OC-TA-SVM based method and KCD is not due to a particularly harmful policy for KCD for selecting the threshold, in Fig. 4 we show the results with Dataset 1 in a different manner. It shows the results of OC-TA-SVM and KCD for the 4 noise levels, using curves that are similar to a ROC curve. By varying the threshold

Table 3
Correctly evaluated sequences by the OC-TA-SVM based method, KCD and PPM for diverse time lapses between changes (s). The results are an average over 100 trials, with the standard deviation of this mean in parenthesis.

s	OC-TA-SVM	KCD	PPM
50	499.1 (0.2)	437.5 (9.9)	478.6 (3.0)
25	498.8 (0.2)	461.1 (6.2)	473.4 (4.9)
20	499.2 (0.1)	442.4 (11.0)	470.7 (4.3)
15	498.7 (0.2)	455.5 (9.7)	467.9 (6.6)
10	495.9 (0.8)	462.7 (7.2)	466.6 (4.4)

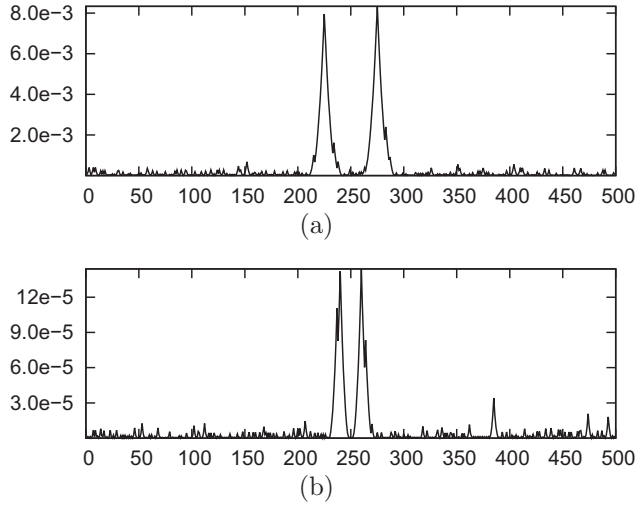

Fig. 5. Index obtained with OC-TA-SVM for an example sequence with $s = 25$ (a) and another with $s = 10$ (b).

we can obtain a different number of correctly detected changes (vertical axis) and a different number of incorrectly detected ones (horizontal axis). These have two sources: changes detected in sequences with no change, and changes detected at the wrong time. This second source prevents the curve from reaching the upper right corner in all situations, as normally happens in a standard ROC curve.

As it can be seen, there is no threshold with which KCD can obtain a superior performance to the proposed method, and the difference becomes bigger with a noise increase.

5.2. Variable time intervals between changes

In the second experiment we study the behavior of the new method in situations where the time elapsed between two adjacent changes is highly variable. In this case we may have a sequence with well-separated, marked abrupt changes for selecting the free parameters, but we may have to calculate the index over a sequence with changes that could be closer to each other.

To this end we designed the following problem. We have 500 sequences of 500 two-dimensional points each with two abrupt changes, the first in point $250 - s$ and the second in point $250 + s$. We use the same distributions as in the first experiment, dataset 1. At the beginning and end of the sequences the points are taken from the distribution used in the first experiment for the sequence with no changes, and between them they are taken from the distribution used after the change. No noise was added to the sequences. We conducted tests with $s = 50, 25, 20, 15, 10$.

To adjust the free parameters we used a sequence that alternates between the two different distributions, with abrupt changes at points 100, 200, 300 y 400. This is the same distance which can be found in the sequences with $s = 50$.

In Table 3 we show results for different values of s. As in the last experiment, it shows the number of sequences correctly evaluated by each method, i.e., the number of cases in which there is a change detected in the tolerance interval corresponding to each of the two changes in the sequence, and no other change detected outside those intervals.

In this case, the index obtained with the new method is almost constant, except for the moments when there is an actual change. This is the cause for the excellent performance in this test. Fig. 5 shows, as an example, the index corresponding to one sequence

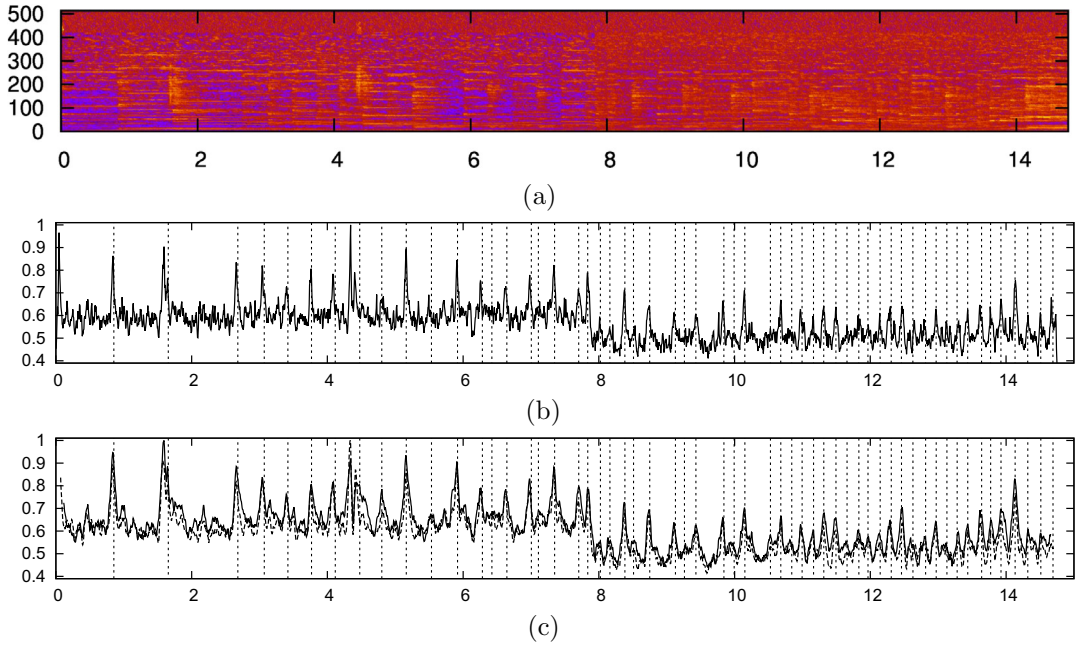

Fig. 6. The spectrogram of the first 15 s of BWV 578 (a). The horizontal axis indicates time, while the vertical axis indicates frequency. The magnitude is given by the color. Lines are used to indicate the index obtained for (b) the new method and (c) KCD (continuous line) and the simple method (dotted line). The three indexes are scaled to have its maximum value at 1. The dotted vertical lines show the actual changes of notes.

where $s = 25$ and another where $s = 10$. As it can be seen, there is no difficulty in varying s in this case.

5.3. A real-world example

Here we show the performance of the new method in a real application: music segmentation, a problem where nonparametric change detection methods have shown good results (Desobry et al., 2005; Gillet, 2007). The piece used in this example is J. S. Bach's Fugue in G Minor, BWV 578, played by Ian Tracey in Marbella in 1986[1].

We conducted this test with the first 15 s of the piece. The spectrogram of the signal was computed, using overlapping windows of 1024 points and calculating the transform every 256 points. This spectrogram, used as input for the methods, is shown in Fig. 6(a).

The free parameters were adjusted using the first 5 s of the signal. After that, we obtained the index for the last 10 s. It is important to note that the average duration of each note is bigger in the first 5 s than in the following 10 s.

In Fig. 6(b) we can see the index obtained for the full sequence of 15 s and the changes between notes, marked as vertical dashed lines.

PPM and KCD were also applied in this example. We could not get satisfactory results with PPM, probably because of the high dimensionality of the dataset (513 dimensions). With KCD we obtained similar results to OC-TA-SVM (Fig. 6(c)). Given that this example presents the characteristics studied in the above artificial examples (noise and a different rate of change between a training and test set), why cannot we obtain a better result?

To answer this question, we executed the same experiment but with a simpler method: using two sliding windows as in KCD, we calculate the distance between the means of each window. That is, we suppose that the mean of the distribution is enough in this case to find the note changes, so we do not need a more sophisticated method, such as One-Class SVM, to model the data distribution.

The results are shown in Fig. 6(c), dotted line. These are very similar to the ones obtained with OC-TA-SVM and KCD. This shows that the example is simple enough not to need a complex method to model the data. Nevertheless, with the new index — as with KCD — we can obtain the simple behavior needed in this case.

6. Conclusions

In this work, we first proposed a new method aimed at the estimation of the support of a high dimensional distribution for non stationary problems, the OC-TA-SVM. This is achieved by dividing the data into m consecutive time windows and creating a coupled sequence of m One-Class SVMs, each one of them being optimal in its corresponding time window. As shown in Section 3, the cost of obtaining this sequence is similar to the cost of a single OC-SVM trained with all available data. Using some simple experiments, we showed that the proposed method is sound and can generate better sequences than other techniques.

We then applied the new method to the problem of abrupt change detection. To this end, the dissimilarity between adjacent models was used as an indication of the probability of an abrupt change to be present between them.

The experiments realized in the previous sections confirmed the expected desirable features of the proposed method. It can be applied to situations where the distribution of the data is unknown, given that it is based on OC-SVM, and yields good results as shown by the experiment of Section 5.1. Of course, when data distribution information is available, a method that uses it, such as PPM, should be used, as the experiment suggests. Furthermore, the experiment with real world data of Section 5.3 showed that it can be successfully applied to high dimensional datasets. These two characteristics are shared with KCD.

There are also benefits with respect to KCD. As the experiment in Section 5.1 showed, the index of the proposed method is more robust to noise than KCD. This advantage was expected, since the

[1] A sample version can be found at http://www.el-organo.com/download/organ/bach/bwv578/bwv578.htm, accessed in May 2012.

sequence of models obtained with TA-SVM was already shown to be more robust to noise than simply using sliding windows. The experiment of Section 5.2 demonstrated another advantage: good results can be obtained even when the interval between changes is highly variable. This is important since the method (the same as KCD) has free parameters that need to be selected before training. A common strategy is to use the parameters that give best results in a small portion of data where the position of the abrupt changes is known. The experiment showed that the changes are detected even when the interval between them is much smaller than the interval between changes in the known part of the data.

Two lines of research can be pursued in the future. On one hand, OC-TA-SVM can be applied to novelty detection (i.e., the detection of points which differs significantly from the rest (Tax & Duin, 2004)) in non-stationary environments, in particular fault detection in machinery that slowly decays in time. On the other hand, the simple abrupt change detection method here proposed can be used in audio applications, probably in more complex problems of music segmentation.

Acknowledgment

We acknowledge partial support for this project from ANPCyT Grant PICT-2008 237.

References

Basseville, M., & Nikiforov, I. V. (1993). *Detection of abrupt changes – theory and application*. Prentice-Hall.

Bersimis, S., Psarakis, S., & Panaretos, J. (2007). Multivariate statistical process control charts: An overview. *Quality and Reliability Engineering International, 23*, 517–543.

Blythe, D., von Bunau, P., Meinecke, F., & Muller, K.-R. (2012). Feature extraction for change-point detection using stationary subspace analysis. *IEEE Transactions on Neural Networks and Learning Systems, 23*, 631–643.

Camci, F., & Chinnam, R. B. (2008). General support vector representation machine for one-class classification on non-stationary classes. *Pattern Recognition, 41*, 3021–3034.

Chang, C.-C., & Lin, C.-J. (2011). Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology, 2*, 27:1–27:27.

D'Angelo, M. F. S. V., Palhares, R. M., Takahashi, R. H. C., & Loschi, R. H. (2011). Fuzzy/bayesian change point detection approach to incipient fault detection. *IET Control Theory and Applications, 5*, 539–551.

Desobry, F., Davy, M., & Doncarli, C. (2005). An online kernel change detection algorithm. *IEEE Transactions on Signal Processing, 53*, 2961–2974.

Fergani, B., Davy, M., & Houacine, A. (2008). Speaker diarization using one-class support vector machines. *Speech Communication, 50*, 355–365.

Giacinto, G., Perdisci, R., Rio, M. D., & Roli, F. (2008). Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion, 9*, 69–82.

Gillet, O. (2007). On the correlation of automatic audio and visual segmentations of music videos. *IEEE Transactions on Circuits and Systems for Video Technology, 17*, 347–355.

Golosnoy, V., Ragulin, S., & Schmid, W. (2009). Multivariate cusum chart: Properties and enhancements. *AStA Advances in Statistical Analysis, 93*, 263–279.

Grinblat, G., Uzal, C., Ceccatto, H., & Granitto, P. (2011). Solving non-stationary classification problems with coupled support vector machines. *IEEE Transactions on Neural Networks, 22*, 37–51.

Hartigan, J. A. (1990). Partition models. *Communications in Statistics – Theory and Methods, 19*, 2745–2756.

Hayton, P., Utete, S., King, D., King, S., Anuzis, P., & Tarassenko, L. (2007). Static and dynamic novelty detection methods for jet engine health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365*, 493–514.

Hsiao, W.-F., & Chang, T.-M. (2008). An incremental cluster-based approach to spam ltering. *Expert Systems with Applications, 34*, 1599–1608.

Loschi, R. H., Cruz, F. R. B., Iglesias, P. L., & Arellano-Valle, R. B. (2003). A gibbs sampling scheme to the product partition model: An application to change-point problems. *Computers & Operations Research, 30*, 463–482.

Manevitz, L. M., & Yousef, M. (2001). One-class svms for document classification. *Journal of Machine Learning Research, 2*, 139–154.

Montgomery, D. C. (2009). *Introduction to statistical quality control* (6th ed.). New York: Wiley.

Müller, P., & Quintana, F. A. (2010). Random partition models with regression on covariates. *Journal of Statistical Planning and Inference, 140*, 2801–2808.

Muñoz, A., & Morguerza, J. M. (2004). One-class support vector machines and density estimation: The precise relation. *Lecture Notes in Computer Science, 3287*, 216–223.

Petrovic, P., Jakovljevic, Z., & Milacic, V. (2010). Context sensitive recognition of abrupt changes in cutting process. *Expert Systems with Applications, 37*, 3721–3729.

Pignatiello, J. J., & Runger, G. C. (1990). Comparisons of multivariate cusum charts. *Journal of Quality Technology, 22*, 173–186.

Quintana, F. A. (2006). A predictive view of bayesian clustering. *Journal of Statistical Planning and Inference, 136*, 2407–2429.

Rabaoui, A. (2007). Improved one-class svm classifier for sounds classification. In *Proceedings of the IEEE conference on advanced video and signal based surveillance* (pp. 117–122).

Schölkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*, 1443–1471.

Tax, D. M. J., & Duin, R. P. W. (2004). Support vector data description. *Machine Learning, 54*, 45–66.

R Core Team. (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna, Austria. ISBN 3-900051-07-0.

Tsai, C.-J., Lee, C.-I., & Yang, W.-P. (2009). Mining decision rules on data streams in the presence of concept drifts. *Expert Systems with Applications, 36*, 1164–1178.

Ukil, A., & Zivanovic, R. (2006). Abrupt change detection in power system fault analysis using adaptive whitening filter and wavelet transform. *Electric Power Systems Research, 76*, 815–823.

Ukil, A., & Zivanovic, R. (2007). Application of abrupt change detection in power systems disturbance analysis and relay performance monitoring. *IEEE Transactions on Power Delivery, 22*, 59–66.

Wu, J., Gan, M., & Jiang, R. (2011). Prioritisation of candidate single amino acid polymorphisms using one-class learning machines. *International Journal of Computational Biology and Drug Design, 4*, 316–331.