Novelty Detection with One-Class Support Vector Machines

John Shawe-Taylor and Blaž Žličar

Abstract In this paper we apply one-class support vector machine (OC-SVM) to identify potential anomalies in financial time series. We view anomalies as deviations from a prevalent distribution which is the main source behind the original signal. We are interested in detecting changes in the distribution and the timing of the occurrence of the anomalous behaviour in financial time series. The algorithm is applied to synthetic and empirical data. We find that our approach detects changes in anomalous behaviour in synthetic data sets and in several empirical data sets. However, it requires further work to ensure a satisfactory level of consistency and theoretical rigour.

Keywords Financial time series • Novelty detection • One-class SVM

1 Introduction

We apply one-class support vector machine (OC-SVM) to synthetic and empirical data and test its ability to detect anomalous behaviour in a time series. Anomalous behaviour in this case is a combination of consecutive data points in a time series that do not belong to a distribution identified by the algorithm. We first briefly introduce the theory behind the OC-SVM. Then we present its application to novelty detection in a time series by using lagged returns as inputs. Results, main conclusions and recommendations for further research are outlined at the end.

J. Shawe-Taylor • B. Žličar (✉)
Department of CS, University College London, London, UK
e-mail: j.shawe-taylor@cs.ucl.ac.uk; b.zlicar@cs.ucl.ac.uk

I. Morlini et al. (eds.), *Advances in Statistical Models for Data Analysis*,

2 Background: Novelty Detection and One-Class SVM

We begin by quoting a result from [1] that bounds the likelihood that data generated according to the same distribution used to train OC-SVM will generate a false alarm.

Theorem 1 *Fix $\gamma > 0$ and $\delta \in (0, 1)$. Let $(\mathbf{c}, r)$ be the centre and radius of a hypersphere in a feature space determined by a kernel $\kappa(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ from a training sample $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$ drawn randomly according to a probability distribution $\mathscr{D}$. Let $g(\mathbf{x})$ be the function defined by*

$$g(\mathbf{x}) = \begin{cases} 0, & \text{if } \|\mathbf{c} - \boldsymbol{\phi}(\mathbf{x})\| \leq r; \\ \left(\|\mathbf{c} - \boldsymbol{\phi}(\mathbf{x})\|^2 - r^2\right)/\gamma, & \text{if } r^2 \leq \|\mathbf{c} - \boldsymbol{\phi}(\mathbf{x})\|^2 \leq r^2 + \gamma; \\ 1, & \text{otherwise}. \end{cases}$$

Then with probability at least $1 - \delta$ over samples of size ℓ we have

$$\mathbb{E}_{\mathscr{D}}[g(\mathbf{x})] \leq \frac{1}{\ell} \sum_{i=1}^{\ell} g(\mathbf{x}_i) + \frac{6R^2}{\gamma\sqrt{\ell}} + 3\sqrt{\frac{\ln(2/\delta)}{2\ell}},$$

where R is the radius of a ball in feature space centred at the origin containing the support of the distribution.

Hence, the support of the distribution outside the sphere of radius $r^2 + \gamma$ centred at $\mathbf{c}$ is bounded by the same quantity, since $g(\mathbf{x}) = 1$ for such inputs and is less than 1 elsewhere. Note moreover that the function $g(\mathbf{x})$ can be evaluated in kernel form if the optimisation is solved using its dual.

The theorem provides the theoretical basis for the application of the OC-SVM and it is perhaps worth dwelling for a moment on some implications for time-series analysis.

- Firstly, the assumption made by the theorem about the distribution $\mathscr{D}$ generating the training and test data has strong and weak elements:
  - It is strong in the sense that there are no assumptions made about the form of the input distribution $\mathscr{D}$. It therefore applies equally to long-tailed distributions as it does to multivariate Gaussians. We will perform some experiments with real-world data in which any assumptions about the form of the generating distribution would be difficult to justify.
  - It is weak in the sense that it assumes the training data are generated independently and identically (i.i.d.), something that will not to be strictly true for time series. This assumption becomes more reasonable when the training data are drawn from separate parts of the time series.

- The theorem is one sided in that it bounds the probability that data that arose from the training distribution are mistaken for novel outliers. It does not, however, say anything about the likelihood that novel data are not detected.

In connection with the final point, Vert and Vert [3] provide an interesting analysis showing that if we generate negative data from an artificial background measure μ and train as a 2-class SVM, in the limit of large data the SVM will identify the level sets of the pdf of the training distribution relative to μ. This suggests that it finds the minimal density with respect to μ consistent with capturing a given fraction of the input distribution. Hence, in this case, we can make assertions about the efficiency with which outliers are detected.

3 Problem: Novelty Detection in Financial Time Series

In order to apply OC-SVM to a single time series we follow the approach proposed by Ma and Perkins [2]. We extend this approach by adding an exponential decay parameter so that the more recent lags carry more weight than the older lags, since we are interested in detecting anomalies in the very short window before the occurrence of the extreme volatility, the underlying hypothesis being that the behaviour of the market changes before the occurrence of the spike in the time series.

3.1 Data Preprocessing

A data matrix is constructed in such a way so that the first column represents the original time series and every next column is a lag of the previous column. More specifically, for a time series variable x composed of observations $x(t)$ where $t = 1, \ldots, T$ (T being the number of time points, observations) we perform a vector-to-matrix transformation so that the dimensionality of the original column vector $\mathbf{x}$ changes from *Tx1* to *(T-d-1)* x d forming a data matrix X. Here d represents our choice of the dimensionality expansion, i.e. the number of all columns in the newly formed matrix X in effect reflecting the number of lags we chose to include in the analysis. Alternatively, we can think of this in terms of extending the dimensionality of a data point *x(t)* to a row vector $\mathbf{x}$(t) so that

$$\mathbf{x}(t) = [x(t) \ldots x(t-d+2) \;\; x(t-d+1)] \tag{1}$$

Then the newly formed data matrix in terms of row vectors becomes

$$X = [\mathbf{x}(d), \ldots, \mathbf{x}(T)]^{\mathsf{T}} \tag{2}$$

with dimensions *(T-d-1)* x d (as suggested by Ma and Perkins [2]). We then take a step further and add a decay parameter c so that the weight of each next column falls exponentially with each lag. If we denote a row vector $\mathbf{d} = [1, \ldots, d]$ then we define $c^{\mathbf{d}}$ to be the row vector

$$\mathbf{c} = c^{\mathbf{d}} = [c^1, \ldots, c^d] \tag{3}$$

where c is an arbitrarily chosen decay parameter $0 < c < 1$. The new data matrix taking into account the exponential time decay is then

$$X_c = X \odot D \tag{4}$$

where D is a matrix of decay factors $D = \mathbf{1}^{\mathsf{T}} * \mathbf{c}$ and multiplication between X and D is element by element multiplication. Alternatively, if we denote the number of columns in X as $j = 1, \ldots, d$, then we can simply define the matrix D as having entries $D_{ij} = c^j$. X_c is then centred row-wise in a standard manner using a centering matrix C

$$C = I_d - \frac{1}{d} O_d \tag{5}$$

so that the final centred data matrix with a time decay is:

$$X_c^c = X_c \, C \tag{6}$$

3.2 Novelty Detection Algorithm

In this section we present a step-by-step pseudo-algorithm of OC-SVM based novelty detection in time-series analysis.

Input: a time series *x(t)* of length T. **Output:** points in time identified as novelties.

---

(1) Vector-matrix transformation: Calculate X_c^c using a range of different lags $d = [2 : 20]$ to obtain 19 matrices of different dimensions $XE = [X_2^c \ldots X_{20}^c]$. The value of the decay parameter is set arbitrarily at $c = 0.97$.

(2) Data sets: Each X_E^c is split in a train set (*X-train*) of length $\frac{2}{3}T$ and the remaining third of observations comprise a test set (*X-test*). Further split *X-train* in half, that is into a *sub-X-train* and a *val-X-train* set.

(3) Train OC-SVM: Apply OC-SVM to *sub-X-train* so as to obtain α_i for each X_c^c in the array of matrices $XE = [X_2^c \ldots X_{20}^c]$ individually and for all values of

$\gamma = 2^i$ (where $i = [-10 : 10]$)[1] and $\nu = 2^j$ where ($j = [-15 : -1]$). Then find pseudo-optimal d_o, γ_o and sensitivity parameter ν_o by locating the OC-SVM that achieved the highest accuracy on *val-X-train*.[2]

(4) Test optimal OC-SVM(d_o,γ_o,ν_o): Use pseudo-optimal values determined in (3) to train OC-SVM(d_o,γ_o,ν_o) on *X-train*. Test the model on *X-test* and obtain the novelty signal for the test set.

4 Experiments

Firstly, we describe the construction of synthetic time series and present the empirical data sets (three stock market indices). Next, we comment on the results and outline the challenges.

4.1 Data

Both synthetic and empirical data sets are of about the same length ($T \approxeq 5800$). Synthetic time series are comprised of an original signal in the training set while in the test sets we add anomalies (i.e. time intervals where the original time series is corrupted by an anomalous signal) on the intervals $T = [5000{:}5050, 5300{:}5350, 5600{:}5650]$. We train the OC-SVM algorithm on a data set comprised solely of original (non-anomalous) time series and then test the optimal specification of the model on a test set that includes pre-defined intervals with anomalies. Synthetic time series are constructed as follows:

$$x(t) = \begin{cases} x_a(t) \; for \; t \in [5000, 5050] \wedge [5300, 5350] \wedge [5600, 5650] \\ x_o(t) \; for \; t \notin [5000, 5050] \wedge [5300, 5350] \wedge [5600, 5650] \end{cases} \tag{7}$$

Here $x_o(t)$ denotes the original time series and $x_a(t)$ denotes the anomalous time series. Synthetic data sets are then the following three time series types:

1. **Synthetic 1** time series is a sinusoid with a small standardised random noise in the training set, but with increased standard deviation of the error process on

[1] We use radial basis kernel (RBF) so that $k(\mathbf{x}, \mathbf{y}) = exp(-\gamma||\mathbf{x} - \mathbf{y}||^2)$.

[2] We use a term pseudo-optimal since we simply choose a specification that is able to contain all training data and consequently label *sub-X-train* data sample as novelty-free. Clearly, this is not necessarily the optimal solution nor is it the only solution and presents one of the main challenges related to novelty detection with OC-SVM.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.