

Online Annotation and Prediction for Regime Switching Data Streams

Gordon J. Ross
Institute for Mathematical
Sciences
Imperial College London
gordon.ross03@ic.ac.uk

Dimitris K. Tasoulis
Department of Mathematics
Imperial College London
d.tasoulis@imperial.ac.uk

Niall M. Adams
Department of Mathematics
Imperial College London
n.adams@imperial.ac.uk

ABSTRACT

Regime switching models, in which the state of the world is locally stationary, are a useful abstraction for many continuous valued data streams. In this paper we develop an online framework for the challenging problem of jointly predicting and annotating streaming data as it arrives. The framework consists of three sequential modules: prediction, change detection and regime annotation, each of which may be instantiated in a number of ways. We describe a specific realisation of this framework with the prediction module implemented using recursive least squares, and change detection implemented using CUSUM techniques. The annotation step involves associating a label with each regime, implemented here using a confidence interval approach. Experiments with simulated data show that this methodology can provide an annotation that is consistent with ground truth. Finally, the method is illustrated with foreign exchange data.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

Algorithms, Performance, Experimentation

Keywords

streaming annotation, change detection, CUSUM, recursive estimation

1. INTRODUCTION

Many instances of streaming data can be viewed as if they were generated by a dynamical system made up of several distinct regimes. Each regime corresponds to one possible type of behaviour for the data stream, and we view the dynamics of the stream as being constant within each regime.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'09 March 8-12, 2009, Honolulu, Hawaii, U.S.A.

Copyright 2009 ACM 978-1-60558-166-8/09/03 ...\$5.00.

This representation allows a stream with time varying dynamics to be segmented into a series of stationary processes. This paper develops an online framework for identifying, or annotating, these stationary processes.

The regime-switching model is a special case of Hidden Markov Models (HMMs [18]) where each hidden state corresponds to a particular set of values for the dynamical parameters of the system. Every state change in the HMM therefore represents a change in the dynamics of the stream.

Regime switching models have been studied under several different names in the literature including change-point regression, segmented regression, and switching state-space models. These models have many real-world applications, such financial time-series where the dynamics may change in response to macro-economic events [3], sensor networks where the regimes are different configurations of the sensors [14], and speech recognition [5] where a stream of conversations needs to be given labels which denote the current speaker.

We conceive fitting regime switching models to data streams as a three stage process. **Estimation** is the process of estimating the particular dynamics in a given regime of the stream. **Change detection** is the process of segmenting the stream into different regimes by identifying those points where the stream dynamics changes. Finally, given a list of change-points, **annotation** is the process of labelling these regimes such that different regimes having the same dynamics are given the same label.

Most work on fitting regime switching models has taken place in offline contexts where large amounts of data-points can be stored, and where there is no need to rapidly update the estimation and annotation in response to new data. For example, one early approach [17] to segmentation was to treat the change-points in the model as random variables, and use maximum likelihood to estimate them at the same time as the regression coefficients. However this method is computationally infeasible when dealing with either large amounts of data or large numbers of change points, and more recent approaches such as switching Kalman filters [8] use the EM algorithm to carry out the segmentation.

Although EM-based approaches reduce the computational burden compared to direct maximum likelihood, they are still too slow to be suitable for many streaming data problems. An alternative approach is to make use of methods from the literature on change-point detection. Here a system is monitored online by a change-detector that raises a warning when it appears that the parameters of the model have changed. The CUSUM algorithm introduced in [16]

is an important change-point detection algorithm of this type. [4] extended CUSUM to fit regime switching models by monitoring the either the mean or variance of the generated residuals. Changes in the mean/variance indicates a change in the underlying system. Other notable approaches for change-detection and segmentation include the generalized likelihood test [1] and the marginalized likelihood test [10].

The task of annotating the regimes after the change-points have been detected is similar to the problem of clustering time-series based on their dynamics, for which several approaches have been proposed. One common method is to put a distance metric on the space of dynamical parameters and then give two segments the same annotation if the distance between their respective parameter estimates is lower than some threshold. A standard choice here is the Euclidean distance, which is discussed in [7] and [6]. An alternative to annotation which uses statistical hypothesis testing is explored in [13] and [6].

The current paper adds to the literature by developing a modular framework in which estimation and annotation for streaming data can be performed in a fully online manner. By breaking the problem down into the three separate tasks of estimation, change detection, and annotation, we can exploit the existing literature corresponding to these three fields and synthesise it into a unified framework.

The remainder of this paper proceeds as follow: In Section 2.1 we give an overview of our approach to prediction and annotation, stressing the modular aspects of our framework. We then discuss techniques for estimating the parameters of a stationary system in Section 2.2. Section 2.3 presents techniques for segmenting the system into piecewise stationary regimes, and section 2.5 discusses methods for annotating these segments. Section 2.5 deals with implementation issues such as the setting algorithm control parameters, and finally Section 3 shows the method applied to both simulated and real-world data.

2. THE FRAMEWORK

We use the following multiple regression model to represent the regime switching stream:

$$y_t = X_t \theta(i) + \epsilon_t \quad (1)$$

where y_t denotes the observed values of some time series, X_t is a vector of the covariates, and $\{\epsilon_t\}$ is a sequence of independent and identically distributed zero-mean random variables. This model is sufficiently general to include many common time-series representations, such as AR, ARX, ARMA and ARMAX [2].

The piecewise stationarity of this model implies that the vector $\theta(i)$ of regression coefficients is allowed to change over time. The values possible values of θ are restricted to the discrete set $S = \{\theta(1), \theta(2), \dots\}$. Our model assumes that θ takes on a constant value, say $\theta(1)$, for some period of time before its value spontaneously jumps to $\theta(j)$, $j \neq i$. We use the term **change-points** to denote the points in the series where θ jumps to a new value, and refer to the states of the system in-between each change-point as being **regimes** or segments.

We present a modular framework for predicting the values of (1) while simultaneously carrying out an annotation of the different regimes of the stream, all of which can be performed

in an online manner on streaming data. The framework has three components:

- 1) **Estimation:** Under the assumption that the dynamics of the system are stationary within each regime, an online recursive method is used to estimate the parameter vector θ , treating it as being time independent. This estimate is used for predicting future values of y_i .
- 2) **Segmentation:** A change-detection method operates parallel to the parameter estimator, to monitor the system and detect changes in the dynamics.
- 3) **Annotation:** Every time a change point is detected, an annotation method assigns a label to the newly identified segment. Parameter estimation then restarts.

The modularity of this approach implies that different methods may be used for each of the three stages, with the best choice of method being picked to suit the problem being studied. We proceed to give an instantiation of the framework which uses Recursive Least Squares (RLS) for the estimation, CUSUM for the change detection, and a confidence interval based method for the annotation.

2.1 Estimation

With no computational constraints it is straightforward to estimate θ for a linear system such as (1), within a regime using ordinary least squares (OLS) regression [11]. To handle the demands of streaming data, this estimation can be implemented using Recursive Least Squares (RLS) regression, which updates the estimate of θ and the covariance matrix $X^T X$ recursively rather than recomputing them from scratch whenever a new point is received. The RLS update equations are well known and can be found in [11].

2.2 Segmentation

We formulate the problem of detecting a change in a stream as follows: let $(y_k)_{1 \leq k \leq n}$ be an observed sequence of random variables with conditional density $p_\theta(y_k | y_{k-1}, \dots, y_1)$. A change in the stream is flagged at time $t = t_0$ if before t_0 the parameter θ of this distribution has a constant value θ_0 , while after t_0 it has value $\theta_0 \neq \theta_1$. The task of a change detection algorithm is firstly to detect that a change has occurred, and secondly to provide an accurate estimate of the change time t_0 .

Two useful performance metrics for change detection are the *mean time between false alarms* defined as $T_0 = E_{\theta_0}(t_0)$, and the *mean delay* defined as $T_1 = E_{\theta_1}(t_0)$. The former quantity measures how often we expect to see false positive reports of a change, whereas the latter measures how quickly we expect to detect a change after it occurs. These two quantities can be generalised into the Average Run Length (ARL) function $T(\theta) = E_\theta(t_0)$ which measures the expected time between detections for any given value of θ . This is analogous to the power function in classical hypothesis testing.

The CUSUM algorithm, introduced in [16], is a standard technique for this change detection problem. We assume that θ has a known value of θ_0 and we wish to detect if/when this value changes to a known value θ_1 . We do this by repeatedly carrying out a sequence of sequential likelihood ratio tests[21], beginning a new test whenever the previous one concludes that no change occurs. Formally, we follow [2] and define a decision function:

$$g_k = \max \left(g_{k-1} + \ln \frac{p_{\theta_1}(y_k)}{p_{\theta_0}(y_k)}, 0 \right) \quad (2)$$

A change is flagged whenever g_k exceeds a certain threshold h . The choice of this threshold h is discussed later.

In practice, the value of θ_1 after the change-point will now be known, so we instead replace it with $\theta_1 = \theta_0 + v$ where v is the minimum magnitude of change which we are interested in detecting. When no sensible lower bound can be stated, the CUSUM algorithm is inappropriate and a different approach such as GLR [1] or MLR [10] based segmentation should be used in this module instead.

2.3 Online Annotation

Detecting a change-point provides evidence that the system has now jumped into a new state. At this point, a new RLS filter is initialized from scratch. This is equivalent to increasing a 'forgetting factor' to its maximum value, to cause all of the data seen so far to be forgotten immediately.

For the purpose of annotation, we need to know whether the state just left was another instantiation of a previously seen state, or a new and unseen state. A method used in [7] is to measure the distance of θ_t estimated for this state with the parameter values estimated for all previously seen states, and to classify it using a threshold d . A problem with this approach is that it uses a fixed threshold d , which ignores extra information provided by the RLS parameter estimates in the form of confidence intervals. Instead of using a fixed threshold, we can adapt d to take into account uncertainty about the θ estimate. More precisely, we say that the state we have just estimated is the same as a previously seen state if it lies within the 99% confidence interval for one of these states.

One problem which can cause annotation errors is **state-blending**. Generally, the change-detection algorithm will not detect the change-point immediately, and its estimate of the change-point will lag behind the true change point. This causes problems for the parameter estimation algorithm since several data-points from the new state will be incorporated into the estimate of θ for the old state. If the detection delay is too long, this error may cause the final estimated value of θ to be significantly different from the true value. This may result in the annotation algorithm assigning an incorrect label to the state.

One way to deal with this is to adjust the estimate of the change time to make it more accurate. The online approach used in CUSUM can be combined with an offline maximum likelihood approach by running a window backwards over the series which stores the last n data-points. When a change-point is detected, an offline change-detection algorithm such as those described in [2] can be used to find the most likely location of the change point in this window. The estimate of θ corresponding to this adjusted change-point can then be used instead.

2.4 Implementation Issues

The threshold h in the CUSUM update equation (2) needs to be chosen. A general approach for selecting h is given in [2], where selection is performed in a manner similar to the choice of the significance level in classical Neyman-Pearson hypothesis testing. First, an acceptable rate of false alarms is decided on, and then the ARL function is used to choose h so that the algorithm has this rate. By the optimality

of CUSUM, the resulting test will minimize the time until detection.

However, finding the ARL of the CUSUM algorithm is non-trivial and requires solving a pair of Fredholm integral equations, which is often done using numerical methods – see [12] for a brief review. However in many situations we are interested only in detecting changes in the mean and variance of a Gaussian distribution, and this particular case has been studied in detail, with tables of solutions existing in the literature. See [9] for tables detailing the CUSUM ARL under various values of h for detecting mean changes in a Gaussian sequence, and [19] for similar tables for variance changes. In the case of non-Gaussian data, Monte Carlo methods may be used to approximate the ARL – see [20] for details.

3. EXPERIMENTS

We present some examples of the algorithm described above, first using synthetic data, and then showing a real world example which uses a financial time series.

3.1 Simulations

First, consider an AR(2) process

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \epsilon_t$$

where ϵ_t is a sequence of iid Gaussian variables with mean 0 and standard deviation 1, and the coefficients a_i change values every 500 data points. We consider a scenario with three unique regimes, with the coefficients drawn uniformly. For this illustration, the coefficients for each regime were:

	a_1	a_2
State 1	0.7	0.1
State 2	-0.1	0.4
State 3	0.2	0.2

A realisation of this process with 20 change-points was generated and is shown in figure 1. The above algorithm is applied to this series, with CUSUM being used to monitor for changes in both the mean and variance of the residuals. The thresholds for CUSUM were chosen so that a false positive occurs roughly every 2000 time-iterations. Since the possible states of θ are relatively far apart, and the time between changes is high, all change-points were successfully detected. The segmentation produced is shown along with the true segmentation in the following table

True	1 2 3 2 1 3 1 2 3 1 2 3 1 2 3 1 3 2 3 1
Found	1 2 3 2 1 3 1 2 4 1 2 3 1 2 3 1 3 2 3 1

This segmentation contains a single error, where one occurrence of state 3 was labelled as being an unseen state. This occurred because a false positive change-point was detected just before the true change-point which resulted in the state blending described earlier. A closer investigation revealed that this false positive was the result of several extreme values of the series which occurred close together and caused the detection algorithm to flag a change. One possible method for reducing the effect of these kind of outliers would be to adapt methods from robust statistics to transform the residuals in a way which reduces the impact of extreme values.

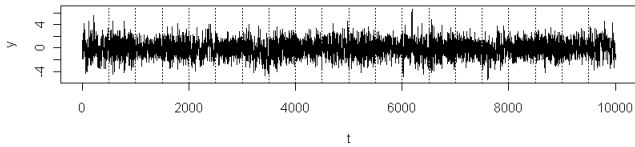


Figure 1: Realisation of an AR(3) process with change-points occurring every 500 ticks, and indicated by the dotted black lines

For a more in-depth simulation, we consider a switching regression problem. Here

$$y(t) = a_0 + a_1X_1(t) + a_2X_2(t) + a_3X_3(t) + \epsilon_t$$

Here ϵ_t again represents a sequence of iid Gaussian variables with zero mean. The X_i 's are sequences of covariates (inputs) which can be used for predicting y_t . We assume that the a_i 's have 5 different regimes and that 50 regime changes occur. Each regime transition occurs after a random time has passed, which is drawn from a uniform distribution on $[100, 500]$. The values for each X_t are drawn from $N(0, 3)$, and each a_i is drawn from a Uniform distribution on $[-1, 1]$.

Although this data is high-dimensional, a visualization of a typical realization can be produced by running a basic RLS filter with forgetting factor over the data and plotting the residuals, as is done in Figure 2. This plot illustrates that the residuals increase immediately after regime-change occurs, which provides insight into how the CUSUM algorithm is able to detect the change-points.

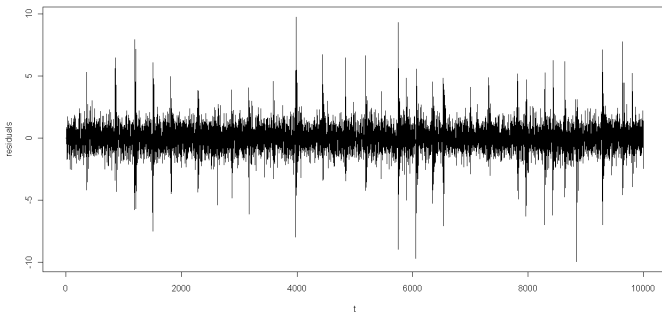


Figure 2: Residuals obtained from running an RLS filter over the AR(3) series in Figure 1. The peaks correspond to the change-points in the series and show that the one-step ahead predictions are most inaccurate directly after a regime-switch has occurred

In order to assess performance, the annotation produced must be compared to the true regime labels. If every change-point is detected then the produced annotation string has the same length as the true annotation string, as in the previous example. In this case, performance can be measured by using a string distance metric such as the Hamming distance to compare the two segmentation strings. However if some change points are not detected, or false positives are produced, these strings will have different lengths and a

straight-forward comparison is not possible. In this case, we borrow the Needleman-Wunsch algorithm [15] from bioinformatics, to perform global sequence alignment on strings with different lengths by allowing gaps to be introduced. Once the strings have been aligned, their dissimilarity is defined using the Hamming distance.

The above simulation was carried out 100 times, with the annotation performance recorded each time. On average the segmentation was 96.3% with a standard deviation of 2.1% accurate according to the above performance measure, indicating that the framework is able to recover the true annotation almost exactly.

Since the framework is designed for use with an online stream, computational speed is an issue. The main performance bottleneck is the RLS module for parameter estimation, which has a time complexity of $O(N^2)$ in the number of dimensions, although various techniques have been proposed to increase the speed of estimation such as RLS lattice filters and QR decomposition filters: see [11] for details. The framework implementation described above, written in unoptimized and uncompiled code in the R programming language, can process around 300 data-points a second when working with 3-dimensional data.

3.2 Real World Data

To illustrate streaming prediction and annotation with real data, we explore a foreign exchange time-series containing the ratio of prices of the US dollar being to Sterling.

An AR(3) model was fit to this data, and a plot of the series along with the discovered change-points its shown in Figure 3.

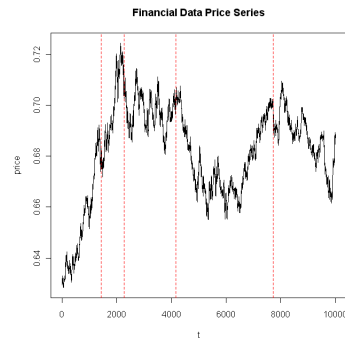


Figure 3: Plot of the financial time-series, with the discovered change-points illustrated with red lines

Although we have no ground-truth to calibrate these results, the change-points appear intuitively plausible since they correspond to patterns of different behavior – the series seems to drift upwards at the start before settling down into a more stable region of behavior.

The values of θ in each segment are shown in the following table:

	a_1	a_2	a_3
Segment 1	0.98	-0.02	0.03
Segment 2	0.93	0.05	0.01
Segment 3	0.91	0.08	-0.01
Segment 4	0.97	0.05	-0.02
Segment 5	1.00	-0.03	0.02

Strikingly, annotation using the confidence-interval method (described earlier) reveals that there is no significant difference between the coefficients in any of these segments, and therefore it appears that the dynamics of the series were entirely stationary. This implies that all of the change-points which seem noticeable to the eye are false positives, and that there are no statistical grounds for believing that the series possesses a true segmentation. Of course, this observation is contingent on the adequacy of model.

4. CONCLUSIONS

We have presented a general framework for performing joint prediction and annotation on streaming data. The modularity of this framework allows techniques from recursive estimation, change detection, and annotation/clustering to be combined. We gave an instantiation of this framework which used the RLS algorithm for estimation, the CUSUM approach to change detection, and a confidence interval based annotation approach. When applied to synthetic data, the resulting algorithm produced an accurate segmentation and annotation. The algorithm was then applied to a financial time series and detected change points which seemed intuitively reasonable, while also revealing that there was little statistical evidence for believing them to be true change-points rather than false positives.

There are several avenues for future work. First, a more detailed investigation into which type of change detection and annotation algorithms are suitable for different situations. Second, it will be useful to investigate ways to make the algorithm less sensitive to false positives due to outliers, perhaps by incorporating results from the literature on robustness. Finally, work can be done on extending the approach to the annotation of multivariate and mixed-type data streams which do not fit easily into the regression framework.

5. ACKNOWLEDGMENTS

This work was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Systems) project and is jointly funded by a BAE Systems and the EPSRC (Engineering and Physical Research Council) strategic partnership, under EPSRC grant EP/C548 051/1.

6. ADDITIONAL AUTHORS

Additional authors: David J. Hand, Department of Mathematics, Imperial College London
email: d.hand@imperial.ac.uk.

7. REFERENCES

- [1] U. Appel and A. V. Brandt. Adaptive sequential segmentation of piecewise stationary time series. *Information Sciences*, 29(1):27–56, 1983.
- [2] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Change Theory and Application*. Prentice Hall, 1993.
- [3] N. P. B. Bollen, S. F. Gray, and R. E. Whaley. Regime switching in foreign exchange rates: Evidence from currency option prices. *Journal of Econometrics*, 94(1-2):239–276, 2000.
- [4] R. L. Brown, J. Durbin, and J. M. Evans. Techniques for testing the constancy of regression relationships over time. *Journal of the Royal Statistical Society. Series B (Methodological)*, 37(2):149–192, 1975.
- [5] M. A. Carlin and B. Y. Smolenski. Detection of speaker change points in conversational speech. *2007 IEEE Aerospace Conference, Vols 1-9*, pages 1566–1573, 2007.
- [6] M. Corduas and D. Piccolo. Time series clustering and classification by the autoregressive metric. *Computational Statistics & Data Analysis*, 52:1860–1872, 2008.
- [7] K. Deng, A. W. Moore, and M. C. Nechyba. Learning to recognize time series: Combining ARMA models with memory-based learning. *1997 IEEE International Symposium On Computational Intelligence In Robotics Automation - Cira '97, Proc.*, pages 246–251, 1997.
- [8] Z. Ghahramani and G. E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12(4):831–864, 2000.
- [9] A. L. Goel and S. M. Wu. Determination of ARL and a contour nomogram for CUSUM charts to control normal mean. *Technometrics*, 13(2):221–230, 1971.
- [10] F. Gustafsson. The marginalized likelihood ratio test for detecting abrupt changes. *IEEE Transactions on Automatic Control*, 41:66–78, 1996.
- [11] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, 1996.
- [12] S. Knoth. Computation of the ARL for CUSUM-S2 schemes. *Computational Statistics & Data Analysis*, 51(2):499–512, 2006.
- [13] E. A. Maharaj. Clusters of time series. *Journal of Classification*, 17(2):297–314, 2000.
- [14] V. Manfredi, S. Mahadevan, and J. Kurose. Switching Kalman filters for prediction and tracking in an adaptive meteorological sensing network. *Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 197–206, 2005.
- [15] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, March 1970.
- [16] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.
- [17] R. E. Quandt. The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the American Statistical Association*, 53(284):873–880, 1958.
- [18] L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, Jan 1986.
- [19] M. Srivastava. CUSUM procedure for monitoring variability. *Communications in Statistics - Theory and Methods*, 26(12):2905–2926, 1997.
- [20] G. Verdier, N. Hilgert, and J.-P. Vila. Adaptive threshold computation for CUSUM-type procedures in change detection and isolation problems. *Computational Statistics & Data Analysis*, 52(9):4161–4174, 2008.
- [21] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.