A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data

Kenji Yamanishi
NEC Corporation
4-1-1,Miyazaki,Miyamae,
Kawasaki,Kanagawa 216-8555,JAPAN

Jun-ichi Takeuchi
NEC Corporation
4-1-1,Miyazaki,Miyamae,
Kawasaki,Kanagawa 216-8555,JAPAN

ABSTRACT

We are concerned with the issues of outlier detection and change point detection from a data stream. In the area of data mining, there have been increased interest in these issues since the former is related to fraud detection, rare event discovery, etc., while the latter is related to event/trend change detection, activity monitoring, etc. Specifically, it is important to consider the situation where the data source is non-stationary, since the nature of data source may change over time in real applications. Although in most previous work outlier detection and change point detection have not been related explicitly, this paper presents a unifying framework for dealing with both of them on the basis of the theory of on-line learning of non-stationary time series. In this framework a probabilistic model of the data source is incrementally learned using an on-line discounting learning algorithm, which can track the changing data source adaptively by forgetting the effect of past data gradually. Then the score for any given data is calculated to measure its deviation from the learned model, with a higher score indicating a high possibility of being an outlier. Further change points in a data stream are detected by applying this scoring method into a time series of moving averaged losses for prediction using the learned model. Specifically we develop an efficient algorithms for on-line discounting learning of auto-regression models from time series data, and demonstrate the validity of our framework through simulation and experimental applications to stock market data analysis.

1. INTRODUCTION

1.1 Contribution of This Paper

Identification of outliers in a data stream has been one of the most exciting topics in data mining (e.g., [10],[3],[15]) and in statistics (e.g., [2]) since it can lead to discovery of unexpected and interesting knowledge. On the other hand, the issue of detecting change points in time series data has extensively been addressed in statistics (e.g., [7]) and has become one of the issues receiving scant attention in data mining, which is recognized as *event change detection* [5] and closely related to *activity monitoring* [4].

Let us illustrate these issues by using network access log analysis as an example. Suppose that you have a data stream of network access logs, each of which is specified by numerical variables including access time, duration, etc. We may first learn a statistical regularity from the data, then detect outliers by investigating how much each data is deviated from the regularity. Identifying such outliers may lead to network intrusion detection since criminal or suspicious activities may often induce statistical outliers (see e.g., [3],[15]). The problem of change point detection is here to identify the time when a significant change of statistical behavior of the access pattern has occurred.

Specifically we make the following requirements on the outlier/change point detection algorithm:
A) *The detection process should be on-line.* That is, an outlier or a change point should be detected immediately after it appeared, and B) *The detection should be adaptive to non-stationary data sources.* That is, an outlier or a change point should be detected even if the nature of the data source may change over time. The major contribution of this paper is to propose a unifying framework for detecting outliers and change points under the above requirements A) and B).

In [15],[16], we proposed a general framework for statistical outlier detection, under the assumption that the data source is non-stationary and each data is independently drawn according to it. In this framework, we employed a Gaussian mixture model as a statistical model for continuous variables while a histogram density as that for discrete variables. We developed an algorithm for on-line discounting learning of the model, where it can track the changing data source adaptively by gradually forgetting the effect of past data. Outliers are detected relative to the learned statistical model. Then for a new input data, its score is calculated as its deviation from the learned model, with a high score indicating a high possibility of being a statistical outlier.

In this paper we extend the framework in [15],[16] toward the following two directions: 1) *dealing with time series data*, and 2) *detecting change points in a data stream.*

As for 1), we replace a finite mixture model employed in the previous framework in [15] with the *AR (autoregression) model* (see e.g., [1]) in order to represent the underlying mechanism of generating time series data, and propose a new algorithm for on-line discounting learning of the AR model. We then employ this algorithm to detect outliers

SIGKDD 02 Edmonton, Alberta, Canda

from time series data.

As for 2), we present a new framework for change point detection by connecting it to outlier detection from time series. In this framework we reduce the problem of change point detection into that of outlier detection from time series of moving-averaged scores. We demonstrate the validity of our framework through simulation and experimental applications to real stock market data analysis.

1.2 Related Work

Outlier detection for non-stationary sources has been considered in [3], [15], [11], in which adaptive outlier detection algorithms have been proposed. However, they did not make any explicit statistical modeling so that time series data can be dealt with.

We employ the AR model to represent a statistical behavior of time series data for the purpose of detecting outliers and change points. The AR model is one of the most typical statistical models for time series, which has extensively been explored (see e.g.,[1],[9],[13]) in statistics. Most existing algorithms of estimating parameters for the AR model are designed under the assumption that the data source is stationary. One approach to dealing with non-stationary sources is to introduce an AR model whose coefficients are time varying (see e.g., [9],[13]). In contrast, we deal with non-stationary sources by modifying the AR model estimation algorithm so that the effect of past examples can gradually be discounted as time goes on.

The standard approach to change point detection in statistics has been to apriori determine the number of change points and decide the stationary model to be used for fitting between successive change points (see e.g., [1],[6],[7],[8]). However, the locally stationary assumption should be eliminated since the statistical regularity may be changing over time in real applications.

In [5] the issue of change point detection was addressed without making any assumption that the data source is locally stationary. Instead, a piecewise segmented function was used in [5] to fit the time-dependent data where the change points are defined as the points between successive segments. In their approach a change point may be detected by finding the point such that the total errors of local model fittings of segments to the data before and after that point is minimized. However, it is basically computationally expensive to find such a point since the local model fitting task is required as many times as the number of points between the successive points every time data is input. Further it is not guaranteed that it works well when the data source cannot be well represented by a simple piecewise segmented function.

In contrast, we take a more general approach: We don't fit a piecewise segmented function to the data but rather fit the AR model and update its parameter estimates incrementally so that the effect of past examples is gradually discounted. Then we give a score to each data/each time point, with a higher score indicating a high possibility of being an outlier/a change point. This makes the process more efficient than the approach in [5] and adaptive to the source that cannot be represented by a simple piecewise segmented model.

Further our approach is distinguished from previous work in the following regards:
1) Although in most previous work outlier detection and change point detection have not been related explicitly, this paper gives a clear connection between them and presents a unifying framework for dealing with both of them from the viewpoint of on-line discounting leaning of non-stationary time series. In our framework, we can detect outliers and change points simultaneously on the basis of an identical learning algorithm.
2) The proposed change-detection algorithm itself is computationally efficient and achieves high detection accuracy.

1.3 Organization of This Paper

In our framework, each of outlier detection and change-point detection consists of two parts: *data modeling* and *scoring*. In the data modeling part, we incrementally learn a probability density function from a data sequence. In scoring part, we give a score to each data or each time point on the basis of the learned model.

We consider the two types of data modeling; independent model and time series model. Section 2.1 overviews, according to [15], the finite mixture model as an independent model and the SDEM (sequentially discounting EM) algorithm for learning of it. Section 2.2 introduces the AR model as a time series model and the SDAR (sequentially discounting AR model estimation) algorithm for learning of it. Section 3.1 gives a method of scoring each data for the purpose of outlier detection. Section 3.2 gives a method of scoring each time point for the purpose of change point detection. This is basically reduced to detecting outliers from time series data modeled by the AR model. Section 4 shows experimental results for change-point detection using both the AR model and the independent model. Section 4.1 gives simulation results. Section 4.2 gives experimental results using real data. Section 5 gives concluding remarks.

2. DATA MODELING AND ON-LINE LEARNING ALGORITHMS

We denote a data sequence as $\{x_t : t = 1, 2, \cdots\}$ where t is a time variable. We construct a sequence of probability density functions denoted as $\{p_t : t = 1, 2, \cdots\}$, which describes the underlying mechanism of generating data. This sequence should be incrementally learned from $\{x_t\}$ every time a data x_t is input. We may also construct a sequence of prediced values $\{\hat{x}_t : t = 1, 2, \cdots\}$. Here $\hat{x}_t$ should be calculated on the basis of $\{p_t\}$ and $\{x_t\}$ every time a data x_t is input. Below we describe the models and their learning algorithms for constructing $\{p_t\}$ and $\{\hat{x}_t\}$.

2.1 Independent Model

We first consider the situation where data is independently drawn at each time. Suppose that the multi-dimensional domain $X \subset \mathbf{R}^n$ is continuous and we represent a random variable on X as x. We may represent a probability density function of non-stationary independent data generation using a Gaussian mixture model of the following form:

$$p(x|\theta) = \sum_{i=1}^{k} c_i p(x|\mu_i, \Lambda_i),$$

where k is a positive integer, $c_i \geq 0$, $\sum_{i=1}^{k} c_i = 1$ and each $p(x|\mu_i, \Lambda_i)$ is a d-dimensional Gaussian distribution with density specified by mean μ_i and covariance matrix Λ_i:

$$p(x|\mu_i, \Lambda_i) = \frac{1}{(2\pi)^{d/2}|\Lambda_i|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_i)^T \Lambda_i^{-1}(x-\mu_i)\right)$$

where $i = 1, \cdots, k$ and d is the dimension of each datum. We set $\theta = (c_1, \mu_1, \Lambda_1, \cdots, c_k, \mu_k, \Lambda_k)$.

An on-line discounting learning algorithm for Gaussian mixtures, which we call *SDEM (sequentially discounting EM algorithm)* was proposed in [15]. It is described as follows:

SDEM Algorithm (r, α, k: given)

Step 1. /* Initialization */
Set $\mu_i^{(0)}, c_i^{(0)}, \bar{\mu}_i^{(0)}, \Lambda_i^{(0)}, \bar{\Lambda}_i^{(0)} (i = 1, ..., k)$.
$t := 1$

Step 2. /* Parameter Updating */
while $t \leq T$ (T:sample size)
Read x_t
for $i = 1, 2, ..., k$

$$\gamma_i^{(t)} := (1 - \alpha r) \frac{c_i^{(t-1)} p(x_t | \mu_i^{(t-1)}, \Lambda_i^{(t-1)})}{\sum_{i=1}^k c_i^{(t-1)} p(x_t | \mu_i^{(t-1)}, \Lambda_i^{(t-1)})} + \frac{\alpha r}{k}$$

$$c_i^{(t)} := (1 - r) c_i^{(t-1)} + r \gamma_i^{(t)}$$
$$\bar{\mu}_i^{(t)} := (1 - r) \bar{\mu}_i^{(t-1)} + r \gamma_i^{(t)} \cdot x_t$$
$$\mu_i^{(t)} := \bar{\mu}_i^{(t)} / c_i^{(t)}$$
$$\bar{\Lambda}_i^{(t)} := (1 - r) \bar{\Lambda}_i^{(t-1)} + r \gamma_i^{(t)} \cdot x_t x_t^T$$
$$\Lambda_i^{(t)} := \bar{\Lambda}_i^{(t)} / c_i^{(t)} - \mu_i^{(t)} \mu_i^{(t)T}$$

$t := t + 1$

The SDEM algorithm was introduced by extending Neal and Hinton's incremental EM algorithm [12] so that the effect of past examples can be gradually discounted as time goes on. The SDEM algorithm updates the estimates of parameters with a weighted average depending on the discounting parameter $r (> 0)$ where a smaller value of r indicates that the SDEM algorithm has a larger influence of past examples.

In the SDEM algorithm, a parameter α is introduced in order to improve the stability of the estimates of c_i, which is set to $1.0 \sim 2.0$. Usually $c_i^{(0)} = 1/k$ and $\mu_i^{(0)}$s are set so that they are uniformly distributed over the data space.

The computation time for the SDEM algorithm at each round is $O(d^3 k)$ where d is the dimension of the data and k is the number of Gaussian distributions.

2.2 Time Series Model

Next we consider the situation where a data sequence forms a time series. We employ here an AR(auto-regression) model to represent a statistical behavior of time series data. The AR model is one of the most typical models for representing time series, which has extensively been explored (see e.g., [1],[9],[13]) in statistics.

A time series supposing that its mean of an initial value is zero, is denoted as $\{z_t : t = 1, 2, \cdots\}$. An AR mode of the k-th order is denoted by

$$z_t = \omega z_{t-k}^{t-1} + \varepsilon,$$

where $z_{t-k}^{t-1} = (z_{t-1}, z_{t-2}, \cdots, z_{t-k}), \omega = (\omega_1, \cdots, \omega_k) \in \mathbf{R}^k$, ε is a normal random variable generated according to the Gaussian distribution with mean 0 and covariance matrix Σ: $\mathcal{N}(0, \Sigma)$. Let a time series which we actually observe be $\{x_t : t = 1, 2, \cdots\}$ where

$$x_t = z_t + \mu$$

Then letting $x_{t-k}^{t-1} = (x_{t-1} \cdots x_{t-k})$, the probability density function of x_t represented by the AR model is given by

$$p(x_t | x_{t-k}^{t-1} : \theta) \quad (1)$$
$$= \frac{1}{(2\pi)^{k/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x_t - w)^T \Sigma^{-1} (x_t - w)\right)$$

where $w = \omega(x_{t-k}^{t-1} - \mu) + \mu$. We set $\theta = (\omega_1, \cdots, \omega_k, \mu, \Sigma)$.

We first review a standard batch algorithm for estimating parameters of AR model according to [13]. Define:

$$\hat{\mu} = \frac{1}{t-k} \sum_{i=k+1}^{t} x_i \quad (2)$$

$$C_j = \frac{1}{t-k} \sum_{i=k+1}^{t} (x_i - \hat{\mu})(x_{i-j} - \hat{\mu})^T \quad (3)$$

Eq.(2) denotes an estimate of μ and Eq.(3) denotes an estimate of a covariance function of the $x_1, \cdots, x_t$. Further the coefficients $\omega_1, \cdots \omega_k$ of the AR model are calculated by solving the following linear equation:

$$C_j = \sum_{i=1}^{k} \omega_i C_{j-i} \quad (j = 1, \cdots, k). \quad (4)$$

Let the solution to Eq.(4) be $\hat{\omega}_1, \cdots \hat{\omega}_k$, an estimate of Σ is calculated as

$$\hat{\Sigma} = C_0 - \sum_{i=1}^{k} \hat{\omega}_i C_i \quad (5)$$

Note that in the above algorithm of learning AR models it is assumed that the data source is stationary and the parameters are estimated after we see the entire data.

We introduce here the *SDAR(sequentially discounting AR model learning) algorithm*by modifying the above algorithm in the following two regards:

1) *On-line estimation*; that is, the parameters are updated every time a data is observed, and

2) *Discounting property*; introducing a discounting parameter r makes the statistics exponentially decay with a multiplicative factor $(1 - r)$ as the time goes on. This enables us to deal with the non-stationary source.

SDAR algorithm is described as follows:

SDAR Algorithm ($0 < r < 1$: given)

Step 1. **Initialization**
Set $\hat{\mu}_0, C_j, \hat{\omega}_j$ $(j = 1, \cdots, k), \hat{\Sigma}$.

Step 2. **Parameter Updating**
For each time $t (= 1, 2 \cdots,)$,
read x_t, proceed:

$$\hat{\mu} := (1 - r)\hat{\mu} + r x_t$$
$$C_j := (1 - r) C_j + r (x_t - \hat{\mu})(x_{t-j} - \hat{\mu})^T$$

Solve the following equation:

$$C_j = \sum_{i=1}^{k} \omega_i C_{j-i} \quad (j = 1, \cdots, k). \quad (6)$$

Letting the solution to Eq.(6) be $\hat{\omega}_1, \cdots \hat{\omega}_k$, then calculate

$$\hat{x}_t := \hat{\omega}(x_{t-k}^{t-1} - \hat{\mu}) + \hat{\mu}$$
$$\hat{\Sigma} := (1 - r)\hat{\Sigma} + r (x_t - \hat{x}_t)(x_t - \hat{x}_t)^T$$

For each time t, the SDAR algorithm updates the estimates of parameters with a weighted average depending on

the discounting parameter $r(>0)$ where a smaller value of r indicates that the SDAR algorithm has a larger influence of past examples.

We denote as p_t the probability density function of Eq.(1) specified by the parameters updated by the SDAR algorithm at time t. Then we can obtain a sequence of probability densities: $\{p_t : t = 1, 2, \cdots\}$.

3. SCORING

3.1 Outlier Detection

For each input x_t, we calculate the score of x_t by the following formula:

$$Score(x_t) = -\log p_{t-1}(x_t) \tag{7}$$

where the lefthand side of Eq.(7) denotes a prediction loss for x_t relative to a probability density function p_{t-1}, which we call the *logarithmic loss*. From the viewpoint of information theory, the logarithmic loss can be thought of as the codelength required to encode x_t into a binary sequence under the assumption that a data is generated according to the probability density p_{t-1}.

We may also define the score as a statistical deviation between before and after learning from x_t.

$$Score(x_t) = d(p_{t-1}, p_t) \tag{8}$$

where $d(*,*)$ are defined as, for example,

$$\begin{aligned} d(p,q) &= \int |p(x) - q(x)| dx \quad \text{(variation distance)} \\ d(p,q) &= \int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx \quad \text{(Hellinger distance)} \\ d(p,q) &= \int |p(x) - q(x)|^2 dx \quad \text{(quadratic distance)} \end{aligned}$$

where $p(*), q(*)$ are probability density functions. Intuitively, this score measures how large the probability density function p_t has moved from p_{t-1} after learning from x_t. Note that a higher $Score(x_t)$ indicates that x_t is an outlier with higher possibility.

3.2 Change-Point Detection

Letting T be a positive constant and $\{x_t\}$ be a data sequence, we define y_t as the *$T-$averaged score* over $\{Score(x_i) : i = t - T + 1, \cdots, t\}$ as:

$$y_t = \frac{1}{T} \sum_{i=t-T+1}^{t} Score(x_i) \tag{9}$$

where $Score(x_i)$ is calculated according to (7) or (8). Then we obtain a time series $\{y_t : t = 1, 2, \cdots\}$.

We then use the AR model for representing the time series $\{y_t\}$ and employ the SDAR algorithm again to construct a sequence of probability density functions determined by the AR models, which we denote as $\{q_t : t = 1, 2, \cdots\}$. Then letting T' be given, we define the *T'-averaged score at time t* as with Eq.(7) (logarithmic loss) as follows:

$$Score(t) = \frac{1}{T'} \sum_{i=t-T'+1}^{t} (-\ln q_{i-1}(y_i)). \tag{10}$$

Or we may use the following score as with Eq.(8) as follows:

$$Score(t) = \frac{1}{T'} \sum_{i=t-T'+1}^{t} d(q_{i-1}, q_i), \tag{11}$$

where d is the function measuring the difference between two probability density functions as in the previous section.

The key of this approach is to reduce the problem of change point detection into the outlier detection from time series of the $T-$averaged scores. Thereby we can deal with outlier detection and change point detection on the basis of an identical learning algorithm in the same paradigm. This gives a strong connection between the two problems. Note that a higher $Score(t)$ indicates that the time point t is a change point with higher possibility.

In Eq.(9), in the case where T is small, outliers and change points can be detected immediately after they appear, however, they may be difficult to be discriminated from one another. In the case where T is large, it leads to time delay for detecting change points, however, outliers are filtered and only change points are detected accurately.

4. EXPERIMENTAL RESULTS

4.1 Simulation

We evaluated our methods by numerical simulation using two types of datasets. The first dataset is a data sequence

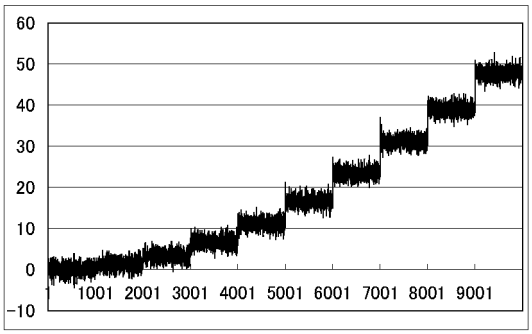

Figure 1: 1st Dataset for Simulation

s.t. each data between change points was drawn according to the following AR model:

$$x_t = 0.6x_{t-1} - 0.5x_{t-2} + \epsilon_t, \tag{12}$$

where ϵ_t is the Gaussian random variable with mean 0 and variance 1. This dataset consists of 10,000 records. The change points occur at time $x \times 1,000$ ($x = 1, 2, \cdots, 9$). Let the difference between the value of the $x-1$-th change point and that of the x-th change point be $\Delta(x)$, which we call *change size* at x. In this case we set $\Delta(x) = x$. It is easier to detect change points of larger x.

We tested two combinations of data modeling and scoring. In the first combination, which we denote as SDEM1, we employed the independent model (with finite mixture with 2 components) for data modeling, while the AR model of the order $k = 2$ for scoring. In the second combination, which we denote as SDAR1, we employed the AR model of the order $k = 2$ for data modeling, while the AR model of

the order $k = 2$ for scoring. Here we used the logarithmic loss as a score and the discounting parameter used in the SDAR algorithm and SDEM algorithm is $r = 0.005$. We set $T = 5$ and $T' = 5$ for scoring as in Eq.(9) and Eq.(10).

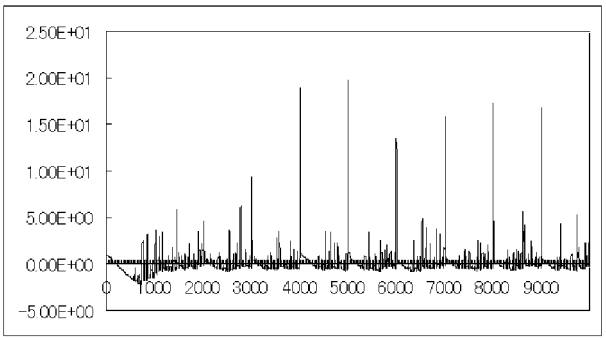

Figure 2: Change Point Detection by SDEM1

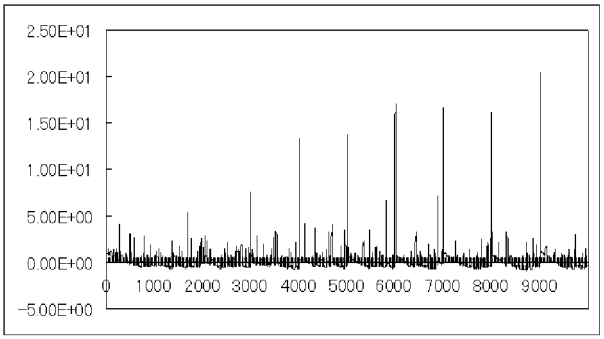

Figure 3: Change Point Detection by SDAR1

Figs. 2 and 3 show the scores calculated by SDEM1 and SDAR1, respectively. In both figures, the horizontal axis shows time while the vertical axis shows score. We observe that SDAR1 detects change points almost as well as SDEM1 while the scores given by SDAR1 reflect the degree of changes more accurately than those of SDEM1.

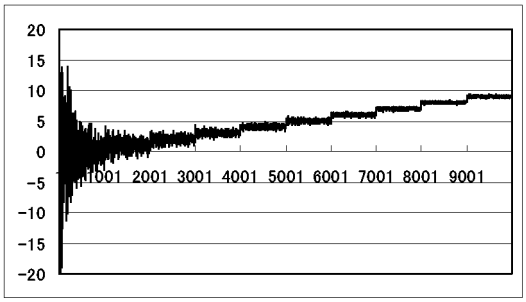

Figure 4: 2nd Dataset for Simulation

The second dataset is a data sequence s.t. each data between change points was drawn according to the AR model

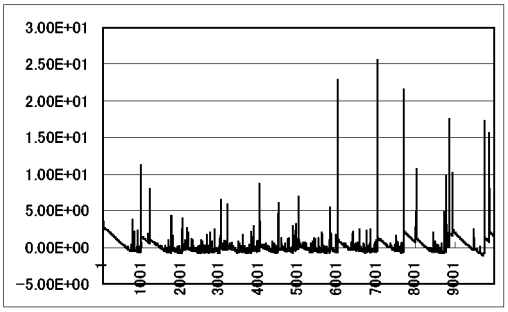

Figure 5: Change Point Detection by SDAR1

(12) in which variance as well as mean change over time. As shown in Fig. 4, the change points occur at time $x \times 1{,}000$ ($x = 1, 2, \cdots, 9$) and the standard deviation is defined as $0.1/(0.01 + time/10{,}000)$. We denote the ratio of the change size at the xth change point to the standard deviation as $R(x)$, which we call the *change-noise ratio*. In this dataset, we set $R(x) \approx x$ ($x = 1, 2, \cdots, 9$).

Fig. 5 shows the scores calculated by SDAR1. We observe that SDAR1 was able to detect change points accurately even if the variance changes over time.

Next, we examined the relation between the false alarm rate and the recall for SDAR1, where the false alarm rate is defined as the percentage of non-change points classified as a change point while the recall is defined as the ratio of change points correctly detected to the total number of change points that should have been detected. We prepared three datasets with various change-noise ratios. For each dataset 100,000 records are generated according to the AR model (12). For each dataset, change points occur at time $x \times 1{,}000$ ($x = 1, 2, \cdots, 9$). The change-noise ratios for three dataset are respectively $R(x) = 20,\ 10,\ 5$ for every change point x. The detection is considered to be correct if a detected change point is located within 50 records after the true change point. Fig. 6 shows the false alarm-recall curves (what we call ROC curve) for the two datasets. Fig. 6 shows

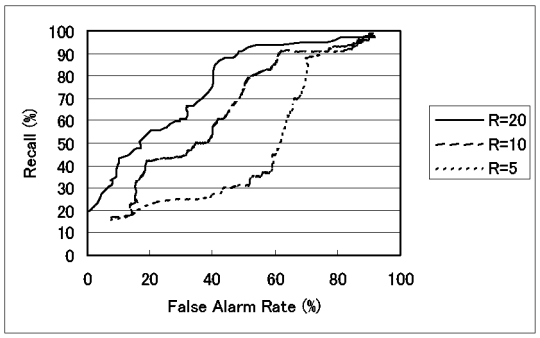

Figure 6: False Alarm Rate vs Recall Curves

that SDAR1 works well when the value of R is not less than

10, although it suffers from identifying change points with random noise when R is small.

4.2 Experiments with Real Data

We used TOPIX(Tokyo stock price index) data to see how well our method for change-point detection works for real problems. Fig.7 shows TOPIX, where the horizontal axis shows time (year: 1985-1995) while the vertical axis shows the index of total asset size of economy of Japan.

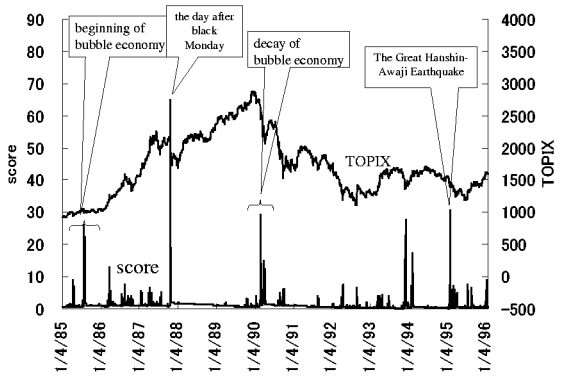

Figure 7: Change Point Detection using SDAR2

We employed the AR model of the order $k = 4$ for data modeling, while the AR model of the order $k = 4$ for scoring. Letting x_t be the original time series data and $y_t = x_t - x_{t-1}$, we dealt with 2-dimensional data (x_t, y_t). We denote this strategy as SDAR2. Here we used the logarithmic loss as a score, and the discounting parameter used in the SDAR algorithm is $r = 0.005$, and $T = 5$, $T' = 5$ for scoring as in (9) and (10).

In Fig. 7, the horizontal axis shows time while the vertical axis shows score. The time points of high scores show the points where significant changes of the index have occurred. We observe that SDAR2 was able to detect real significant changes of the index.For example, Fig.7 shows that among the time points of high scores that SDAR2 gave, there are time points corresponding to the beginning of bubble economy, black Monday, decay of bubble economy, and the great Hanshin-Awaji earthquake, all of which are detected accurately. This demonstrates that our method was able to discover meaningful change points in the index.

5. CONCLUDING REMARKS

We have proposed a framework for detecting outliers and change-points from non-stationary time series. It consists of two parts: data modeling and scoring. In the data modeling part, we incrementally learn a probability density function from a data sequence. Specifically, in this paper we employed the AR model and introduced the SDAR algorithm for on-line discounting learning of the AR models. The SDAR algorithm is characterized in its discounting property that the effect of past data is gradually discounted. This enabled us to deal with non-stationary sources. In scoring part, we give a score to each data or each time point on the basis of the learned model. Specifically we reduced the problem of change point detection into that of outlier detection from time series of moving-averaged scores. Thereby we were able to deal with both problems using an identical learning algorithm within the same paradigm. This gave a unifying view of outlier detection and change point detection.

The following issues remain for future study:
1) *Dealing with ARMA models.* The ARMA (autoregression moving average) model is a promising model for representing time series data. It would be challenging to extend our framework so that it can deal with the ARMA model in order to improve the outlier/change point detection accuracy for the current framework.

2) *Combining Markov models with AR models.* In this paper we have dealt with only continuous variables. In order to deal with time series of categorical variables as well as continuous ones, a time series model over the discrete domain such as a Markov model should be combined with the AR model and a design of new algorithm for on-line discounting learning of them would be required.

6. REFERENCES

[1] H. Akaike and G. Kitagawa, Practices in Time Series Analysis I,II, Asakura Shoten (in Japanese), 1994,1995.

[2] V. Barnett and T. Lewis, *Outliers in Statistical Data*, John Wiley & Sons, 1994.

[3] P. Burge and J. Shaw-Taylor, Detecting cellular fraud using adaptive prototypes, in *Proc. of AI Approaches to Fraud Detection and Risk Management*, pp:9–13, 1997.

[4] T. Fawcett and F. Provost, Activity monitoring: noticing interesting changes in behavior, in *Proc. of KDD-99*, pp:53–62, 1999.

[5] V. Guralnik and J. Srivastava, Event detection from time series data, in *Proc. KDD-99*, pp:33–42, 1999.

[6] 2S. B. Guthery, Partition regression, *Jr. Amer. Statist. Ass.*, 69:945–947, 1974.

[7] D. M. Hawkins, Point estimation of parameters of piecewise regression models, *Jr. of the Royal Statistical Society Series C*, 25(1):51–57, 1976.

[8] M.Huskova, Nonparametric procedures for detecting a change in simple linear regression models, in *Applied Change Point Problems in Statistics* (1993).

[9] G.Kitagawa and W.Gersch, *Smoothness Priors Analysis of Time Series*, Lecture Notes in Statistics, 116, Springer-Verlag (1996).

[10] E. M. Knorr and R. T. Ng, Algorithms for mining distance-based outliers in large datasets, in *Proc. of the 24th VLDB Conference*,pp:392–403, 1998.

[11] U. Murad and G. Pinkas, Unsupervised profiling for identifying superimposed fraud, in *Proc. of PKDD'99*, pp:251-261 (1999).

[12] R. M. Neal and G. E. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, ftp://ftp.cs.toronto.edu/pub/radford/www/publications.html 1993.

[13] T.Ozaki and G.Kitagawa, *A Method for Time Series Analysis*, (in Japanese), Asakura Shoten, (1995).

[14] J. Rissanen, Fisher information and stochastic complexity, *IEEE Trans. Inf. Theory*, IT-42, 1, pp. 40–47 (1996).

[15] K. Yamanishi, J.Takeuchi, G.Williams, and P.Milne, On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms, in *Proc. of KDD2000*, ACM Press, pp:250–254, (2000).

[16] K.Yamanishi and J. Takeuchi, Discovering outlier filtering rules from unlabeled data, in *Proc. of KDD2001*, pp:389-394 (2001).