Change Point Detection in Time Series

Rasmus Erlemann

NTNU, Department of Mathematical Sciences

Time Series

Time series is a series of data points indexed by time stamps.

Change point detection **online** or **offline**

Methods are **parametric** or **nonparametric**

Methods assume the time series is **stationary** or **non-stationary**

Also, by different dependancy structures (for example AR(1))

Time Series

Definition

Time series $\{X_t\}$ is called (weakly) stationary if

- $E(X_t)$ is independent of time
- The covariance function

$$\text{Cov}(X_{t+h}, X_t) = E\left[(X_{t+h} - E(X_{t+h}))(X_t - E(X_t))\right]$$

is independent of time for each h

Definition

Let $\{X_t\}$ be a stationary time series. Autocorrelation at lag h is defined as

$$\rho_X(h) = \text{Cov}(X_{t+h}, X_t)$$

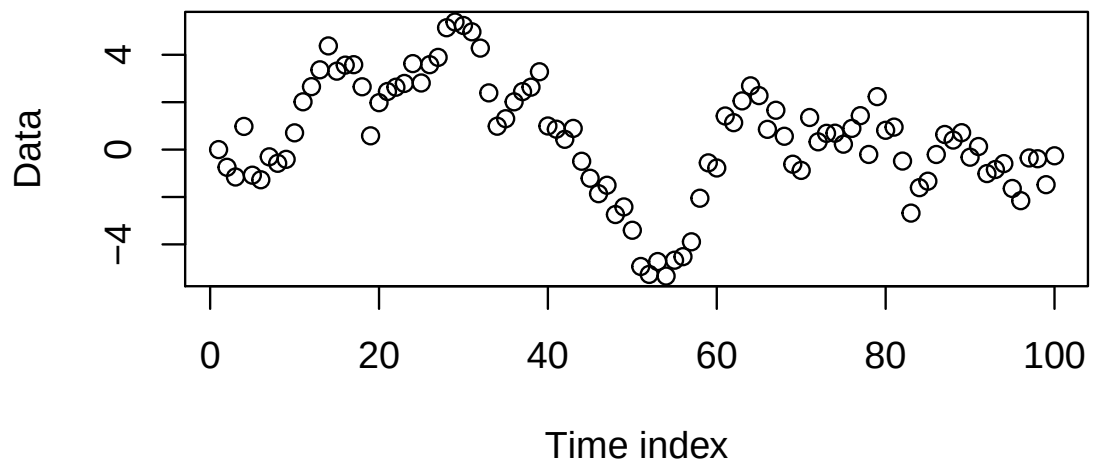
Time Series

Definition

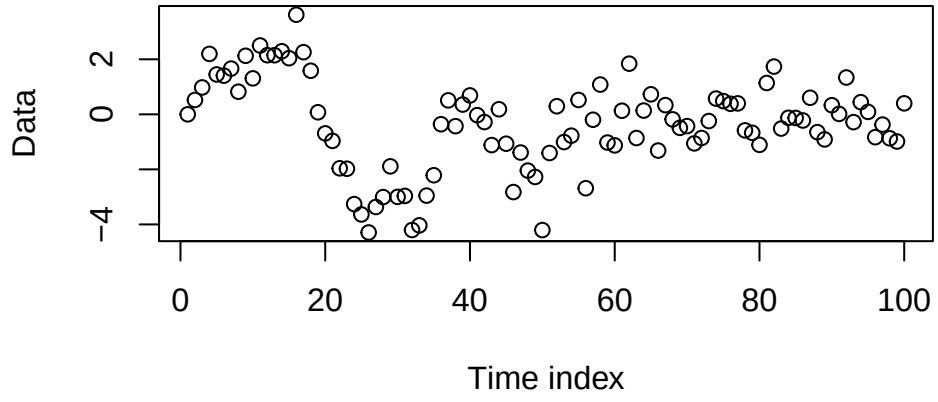
Let $\{X_t\}$ be a stationary time series. Autoregressive model AR(1) assumes it satisfies

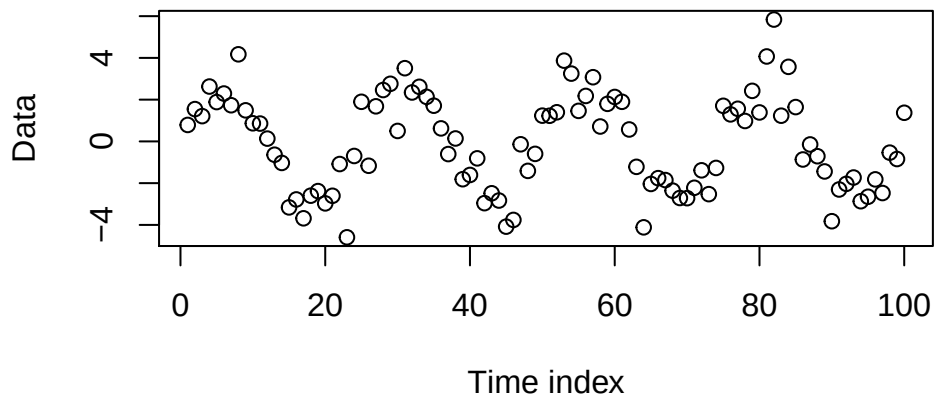
$$X_t = \phi X_{t-1} + Z_t, \quad t = 0, 1, \ldots$$

where $\{Z_t\} \sim WN(0, \sigma^2)$, $|\phi| < 1$ and Z_t is uncorrelated with X_s for each $s < t$.



Time Series


Data
2
0
−4
0
20
40
60
80
100
Time index


Data
4
0
−4
0
20
40
60
80
100
Time index

Multiple Change Points and Autoregression

In the normal IID case we used the at most one change (AMOC) model. Looking for more than 1 change point is computationally intensive. We need to find number of change points m and check all their configurations, with sequence length n, there is

$$\binom{n}{m}$$

If $n = 173$ is it is in the upcoming real life example, we need to consider 1.2×10^{52} cases. We can use genetic algorithms to decrease the search space. [7] uses MDL scores to fit the best model to our data set.

Instead of the IID case, we will work with a simple first-order autoregression AR(1), which allows the variables to be dependant.

Multiple Change Ponts and Autoregression

We develop the MDL score function for the precipitation example.

We will construct a lognormal model for $X_1, \ldots, X_N$ that allows for a reference series $Y_1, , Y_N$ and autocorrelation. Let $\mu_1, \ldots, \mu_{m+1}$ be the location parameters with m change points. If X_t and Y_t are independent and lognormally distributed, then we can model $S_t = \ln(X_t/Y_t)$ as a correlated Gaussian series with a simple first-order autoregression (AR(1)). ϕ is the autocorrelation coefficient and σ^2 is the white noise variance. Such model satisfies

$$S_t = \mu_{r(t)} + \epsilon_t, \quad \epsilon = \phi_{t-1} + Z_t$$

where $\{Z_t\} \sim WN(0, \sigma^2)$. For a fixed number of data points, say m, occuring at times $\tau_1, \ldots, \tau_m$ let $r(t)$ denote the regime number at which regime number at which the time t data point is sampled from. So, if the first change happened at $t = 10$, then $r(t) = 1, \ t < 10$.

Multiple Change Points and Autoregression

The likelihood of this model, allowing for a mean shift at each change point time, is

$$L(\mu_1, \ldots, \mu_{m+1}, \phi, \sigma^2) =$$

$$= (2\pi)^{-N/2} \left(\prod_{t=1}^{N} v_t \right)^{-1/2} \exp \left[-\frac{1}{2} \sum_{t=1}^{N} \frac{(S_t - \hat{S}_t)^2}{v_t} \right]$$

$\hat{S}_t$ is the best linear prediction of S_t and the history $S_1, \ldots, S_{t-1}$ and $v_t = E[(S_t - \hat{S}_t)^2)]$ is the mean squared prediction error. The AR(1) dynamics give

$$\hat{S}_t = \mu_{r(t)} + \phi[S_{t-1} - \mu_{r(t-1)}]$$

for $t \geq 2$ and $\hat{S}_1 = \mu_1$. The prediction errors are $v_t = \sigma^2$ for $t \geq 2$ with the first one $v_1 = \sigma^2/(1 - \phi^2)$.

Multiple Change Points and Autoregression

While optimizing this likelihood is more complex with AR(1) than the IID normal case, it is still not overly difficult and there are also methods for AR(p) models. The mean estimator for l-th segment is

$$\hat{\mu}_l = \frac{1}{\tau_l - \tau_{l-1}} \sum_{t \in R_l}^{N} S_t$$

This estimator is asymptotically adjusted for edge effects. The variance parameter σ^2 and autocorrelation coefficient ϕ are estimated from all data points

$$\hat{\sigma}^2 = \frac{\sum_{t=1}^{N} (S_t - \hat{S}_t)^2}{N}$$

$$\hat{\phi} = \frac{\sum_{t=2}^{N} [S_t - \hat{\mu}_{r(t)}][S_{t-1} - \hat{\mu}_{r(t-1)}]}{\sum_{t=2}^{N} [S_{t-1} - \hat{\mu}_{r(t-1)}]^2}$$

Multiple Change Points and Autoregression

Inserting the estimators into the likelihood L, we get

$$-\ln(L_{\text{opt}}) = -\ln(L[\hat{\mu}_1, \ldots, \hat{\mu}_{m+1}, \hat{\sigma}^2, \hat{\phi}]) = \frac{N}{2}\left[1 + \ln(2\pi) + \ln(\hat{\sigma}^2)\right].$$

The model penalty term for MDL is given by

$$P = \log_2(m) + \sum_{i=2}^{m} \log_2(\tau_i) + 2\log_2(N) + \sum_{i=1}^{m+1} \frac{\log_2(\tau_i - \tau_{i-1})}{2}.$$

Multiple Change Points and Autoregression

We take the optimized likelihood and the penalty term to get MDL.

Changing base 2 logarithms to natural logarithms and ignoring terms constant in N we get

$$\mathrm{MDL} = -\log_2(L_{\mathrm{opt}}) + P = \frac{N}{2}\ln(\hat{\sigma}^2) + \sum_{i=1}^{m+1}\frac{\ln(\tau_i - \tau_{i-1})}{2} +$$

$$+ \ln(m) + \sum_{i=2}^{m}\ln(\tau_i),$$

where L_{opt} is the optimized likelihood and P is a penalty term

Multiple Change Points and Autoregression

Next, we want to determine the optimal model and we do it by minimizing the minimum length score. We want to find m and the change point locations $\tau_1, \ldots, \tau_m$. We use genetic algorithms.

Each parameter configuration is expressed as a chromosome of the form $(m, \tau_1, \ldots, \tau_m)$. Chromosomes for 200 were first simulated at random.

Children of the first generation are made by combining the fitter individuals of the first generation $(m, \tau_1, \ldots, \tau_m)$ and $(j, \eta_1, \ldots, \eta_j)$. Their chromosomes are combined to produce the child $(m + j, \delta_1, \ldots, \delta_{m+j})$. So, the child has both sets of change points.

Next, we thin the change point times of the child by a coin flip for each point. In other words, there is an equal chance to keep or throw away each change point. Some extra possibilities are added to provide more variety.

Iterations run until a termination condition is met.

Real Life Example (Multiple Change Points)

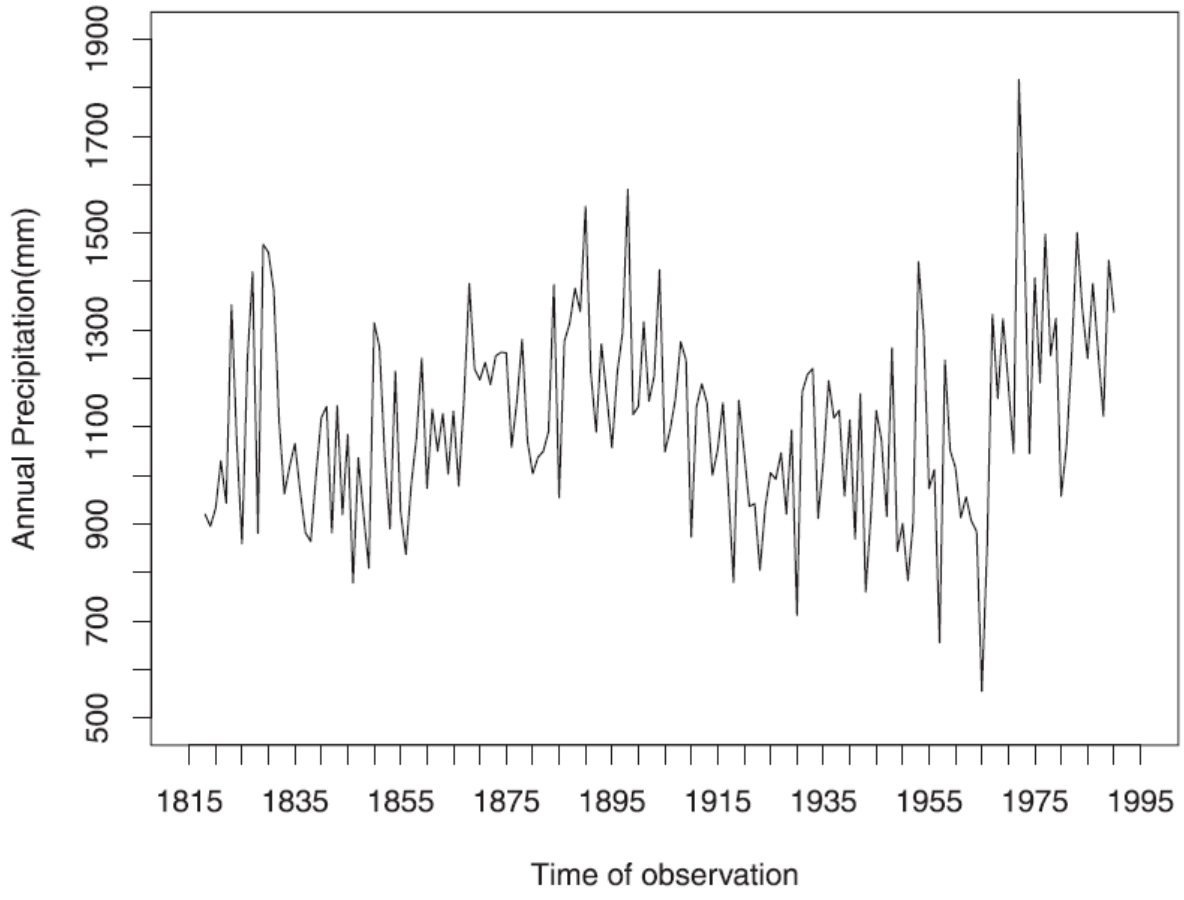


Figure: Annual precipitation series from New Bedford, Massachusetts, during 1818–1990.

Real Life Example (Multiple Change Points)

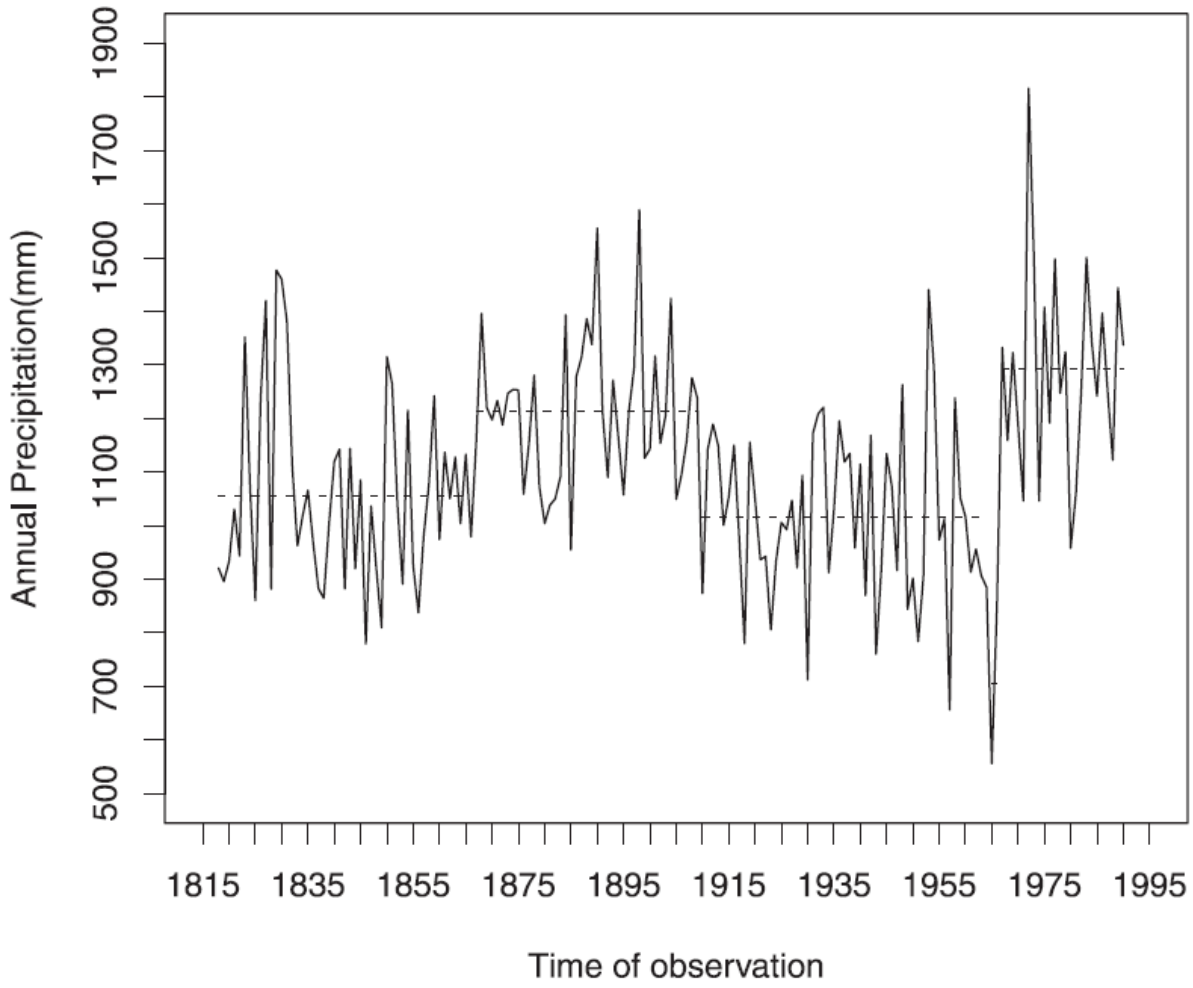


Figure: The algorithm without autoregression converged to a model with for change points at times 1867, 1910, 1965, and 1967. The algorithms converged to the same change points with autoregression and the autocorrelation coefficients came out to be $\hat{\phi} = 0.021$.

Bernoulli Model (CUSUM)

Let $X_1, X_2, \ldots$ be a sequence of Bernoulli random variables with parameters $p_1, p_2, \ldots$. We want to test if there is a change point and the parameter increases at some time stamp k. We monitor for the change point sequentially with the CUSUM test statistic

$$B_k = \max(0, B_{k-1} + (X_k - r)), \quad k = 1, 2, \ldots$$

with $B_0 = 0$. The parameter $0 < r < 1$ is the reference value. The test signals when $B_k > H$, where H is a control limit.

How to choose r and H? Reference [6] recommends $r = \frac{|p_+ - p_-|}{2}$, where p_+ is the parameter value for "in control" process and p_- is the parameter for "out of control" process. The control limit H should be 5 times the process standard deviation.

Real Life Examples (Bernoulli CUSUM)

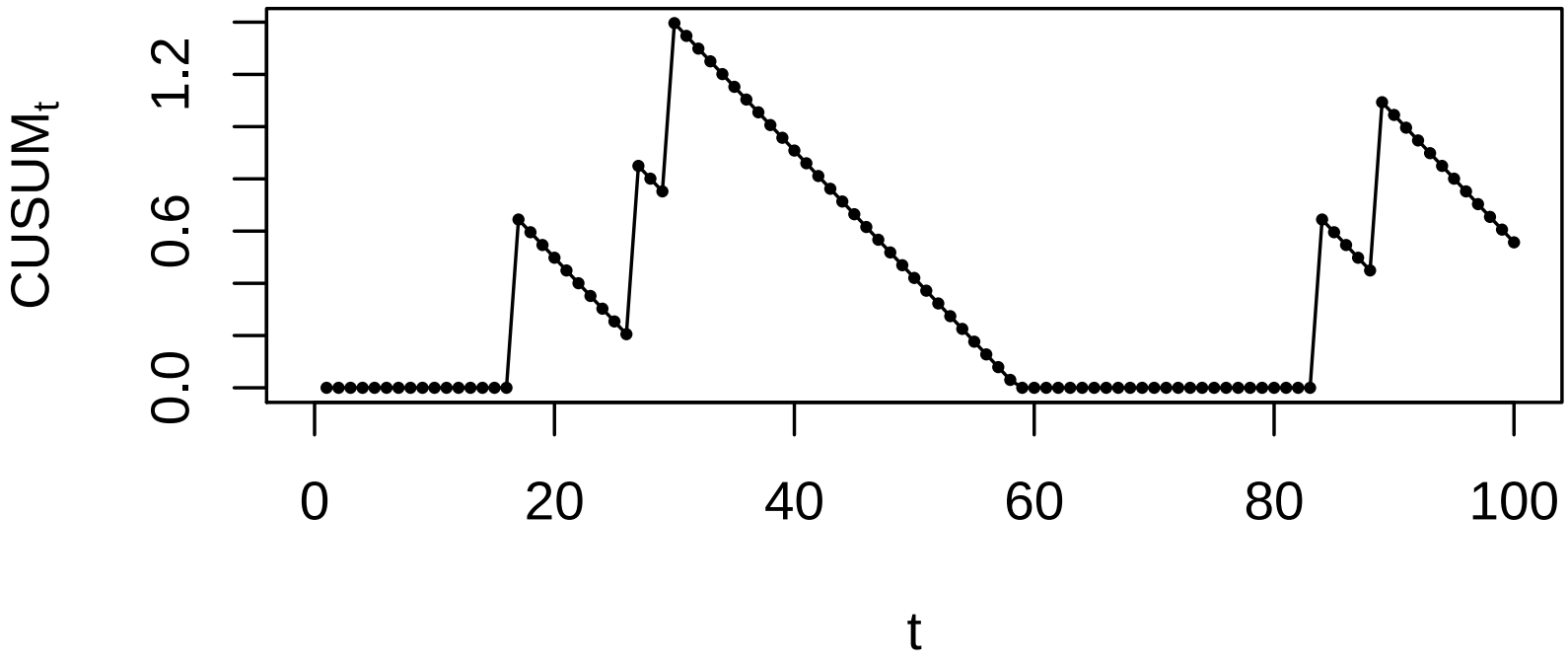


Figure: Data $x_1, x_2, \ldots, x_{100}$ from the Bernoulli distribution with parameter $p = 0.05$. Control limit $H = 2.96$ wasn't achieved.

Real Life Example (Bernoulli CUSUM)

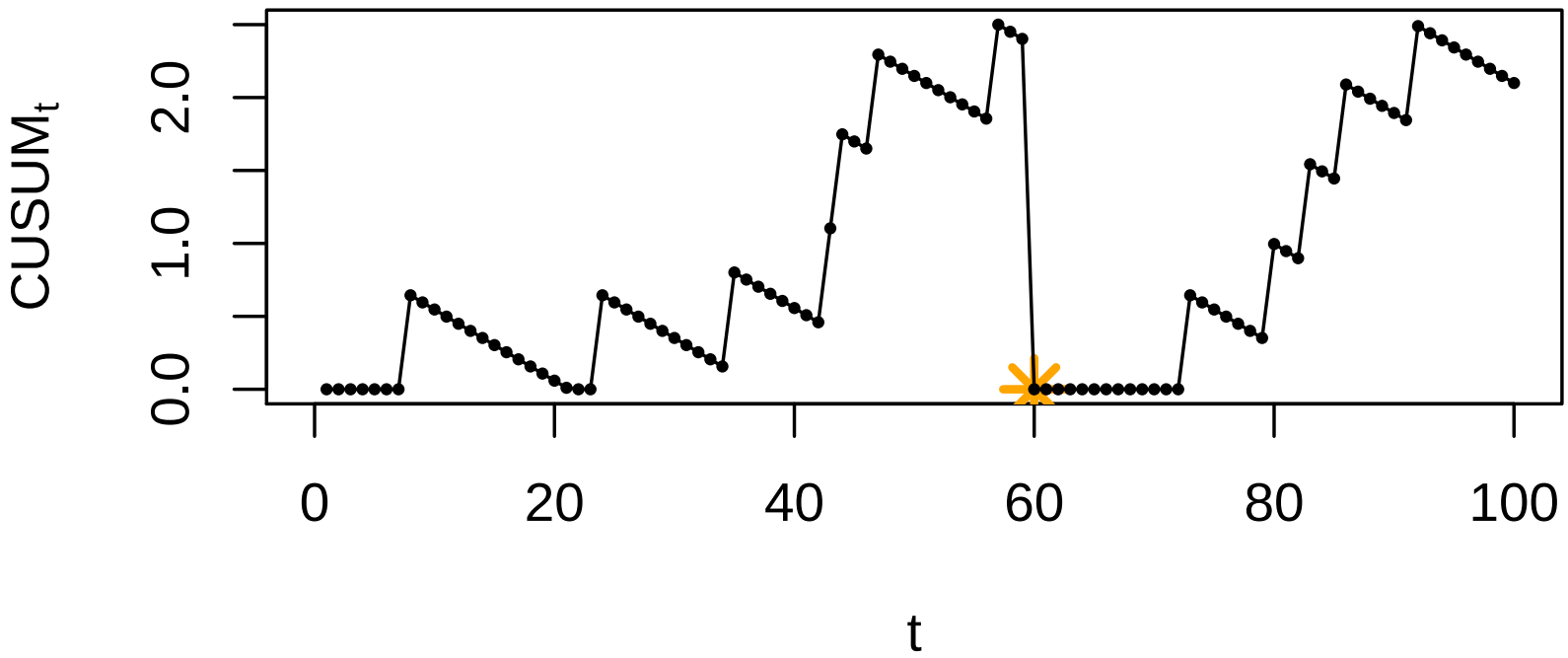


Figure: Data $x_1, x_2, \ldots, x_{30}$ from the Bernoulli distribution with parameter $p = 0.05$ and for $x_{30}, x_{31}, \ldots, x_{100}$ the parameter changed to $p = 0.2$. Yellow marking is when $B_k > H$ and the test signalled.

Direct Density-Ratio Estimation

An online nonparametric method based on [9]. It likelihood based but we don't make assumptions about the dependancy. It can also detect change points for higher order autoregressive models AR(p).

Let $y(t)$ be a d-dimensional time series at time t. We split the time series into two consequetive time windows: reference and test. Let us define the forward subsequence of length k at time t

$$Y(t) = [y(t)^T, y(t+1)^T, \ldots, y(t+k-1)^T]^T.$$

The core likelihood based test is

$$s(Y) = \ln \frac{p_{\text{te}}(Y)}{p_{\text{rf}}(Y)},$$

where p_{te} is the test sequence density and p_{rf} is the reference sequence density.

Direct Density-Ratio Estimation

Let t_{rf} and t_{te} be the starting positions of the reference interval and the test interval. Let us denote

$$Y_{\mathrm{rf}}(i) = Y(t_{\mathrm{rf}} + i - 1),$$

$$Y_{\mathrm{te}}(i) = Y(t_{\mathrm{te}} + i - 1).$$

We test the following hypothesis:

$$H_0 : p(Y(i)) = p_{\mathrm{rf}}(Y(i)), \quad \text{for } t_{\mathrm{rf}} \leq i < t,$$

$$\begin{aligned} H_1 : & p(Y(i)) = p_{\mathrm{rf}}(Y(i)), \quad \text{for } t_{\mathrm{rf}} \leq i < t_{\mathrm{te}}, \\ & p(Y(i)) = p_{\mathrm{te}}(Y(i)), \quad \text{for } t_{\mathrm{te}} \leq i < t. \end{aligned}$$

Direct Density-Ratio Estimation

we can decide whether there is a change point between the reference and test intervals by monitoring the logarithm of the likelihood ratio:

$$S = \sum_{i=1}^{n_{\text{te}}} \ln \frac{p_{\text{te}}(Y(i))}{p_{\text{rf}}(Y(i))},$$

where n_{te} is the number of indices in the test interval.

If S crosses a predetermined threshold, then a change occurs.

How do we calculated the fraction in S? We model the ratio with a nonparametric Gaussian kernel.

The parameter estimation process is adjusted for the method to be online.

Direct Density-Ratio Estimation

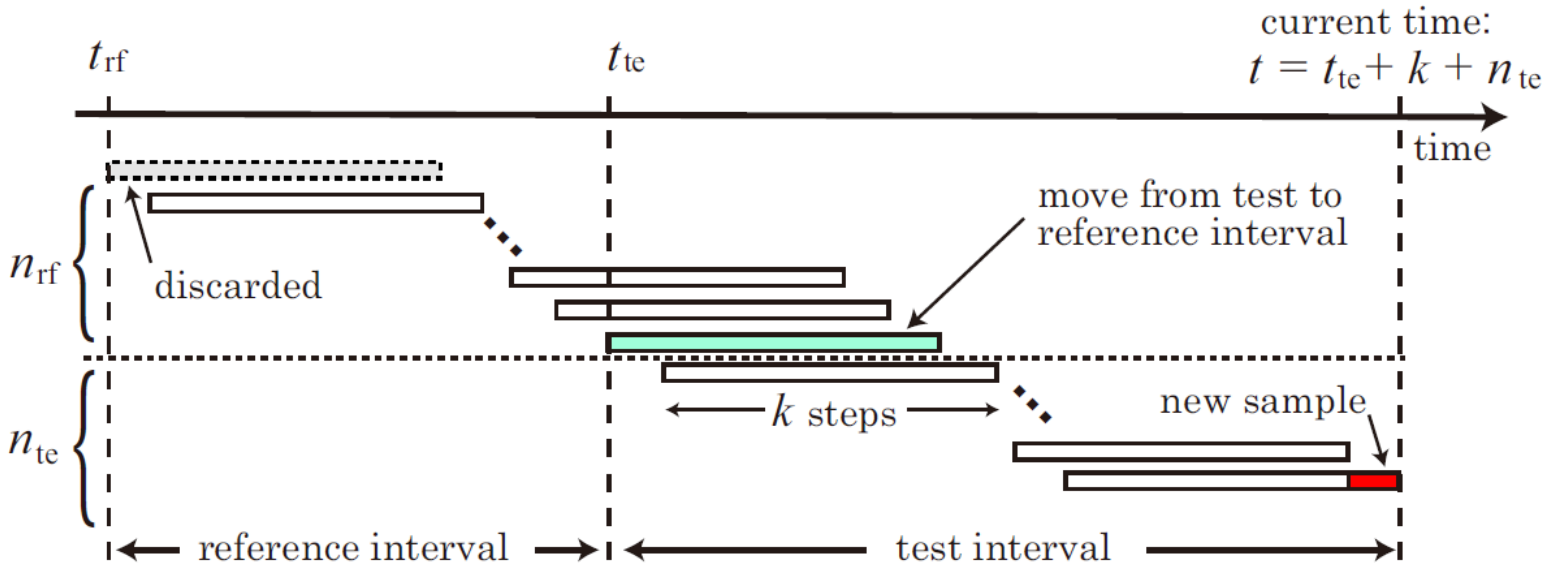


Figure: New sample is given $y(t)$ and the test and reference intervals are shifted one step to the future. Gaussian kernel parameters are updated. The logarithm of the likelihood is then calculated with the new parameters. If S is above the given threshold, we found a change point.

Real Life Example (Density-Ratio)

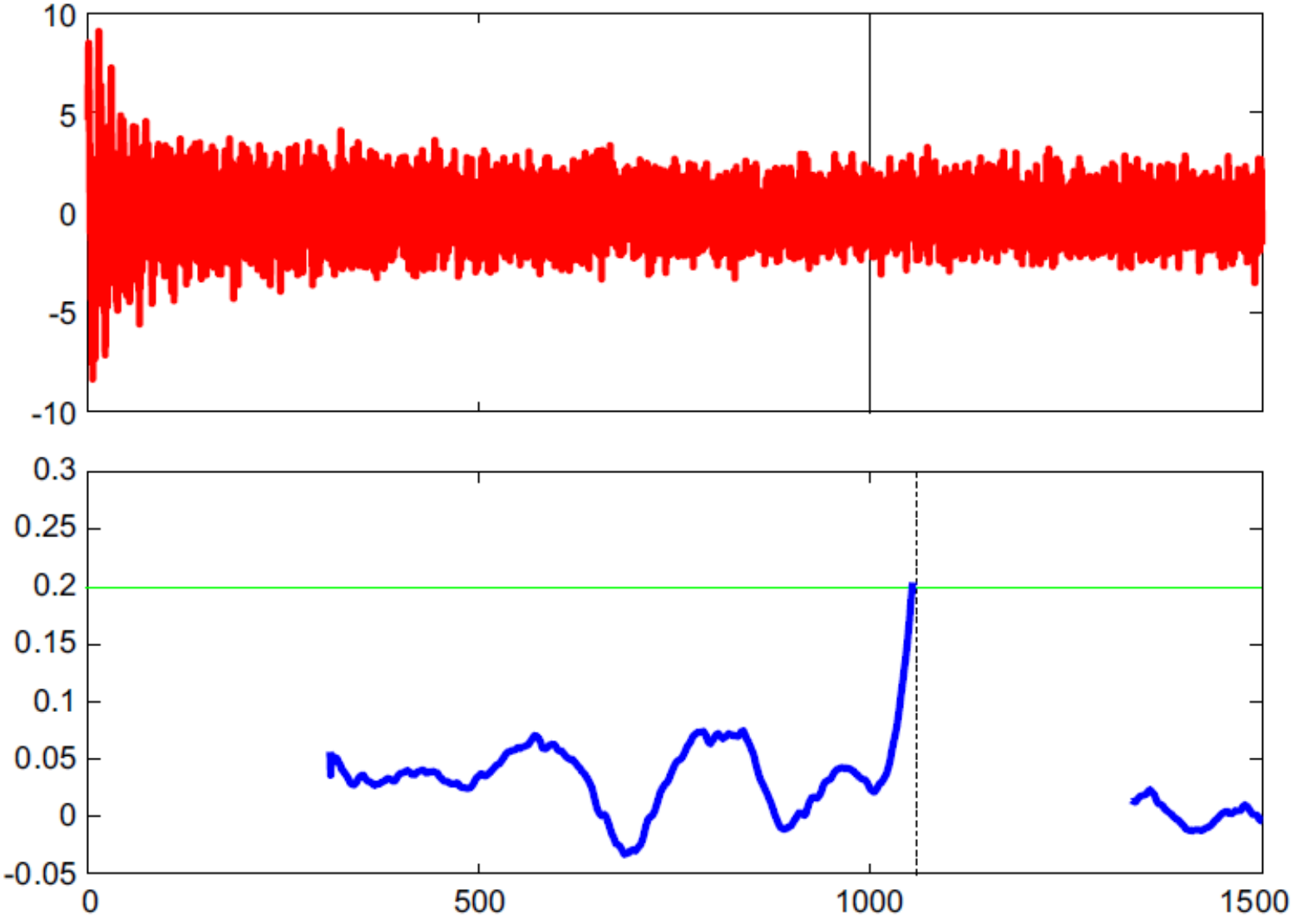


Figure: AR(4) data set with a change point at $t = 1000$.

Deep Learning for Human-Specified Change Points

Let us define a new term, breakpoint which is a human-specified change point.

[8] introduced an unsupervised method for detecting breakpoints in time series. It can also be generalized for change points.

We are given time series with N_c variables and T time stamps $d \in \mathbb{R}^{N_c \times T}$. We partition d into series of segments, according to the time window N_w. Let us denote the time windows as $s_1, s_2, \ldots, s_{N_c N_w}$.

We use unsupervised methods - autoencoders with 2 hidden layers (example). Data is mapped to hidden units with an encoder to obtain the feature representation

$$f = E_{\text{enc}}(s) = g_{\text{enc}}(Ws + b_e),$$

where g_{enc} is a nonlinear activation function and b_e is a bias vector.

Deep Learning for Human-Specified Change Points

Features are then mapped back to the data domain with

$$\tilde{s} = D_{ec}(f) = g_{\text{dec}}(W'f + b_d),$$

where usually $g_{\text{dec}} = g_{\text{enc}}$, $W' = W^T$ and b_d is a bias vector.

We use the stochastic gradient descent to minimize the error of the reconstructed sample $D_{\text{ec}}(E_{\text{enc}}(s))$ to get the optimal values for W, b, b_d, W'.

$$\min J(W, b_e, b_d, W') = \min \sum_{t=1}^{T/N_w} L(s_t, D_{\text{ec}}(E_{\text{ec}}(s_t))) + \lambda \sum_{t,i} W_{t,i}^2$$

where $L(u, v)$ is a loss function depending on the input range.

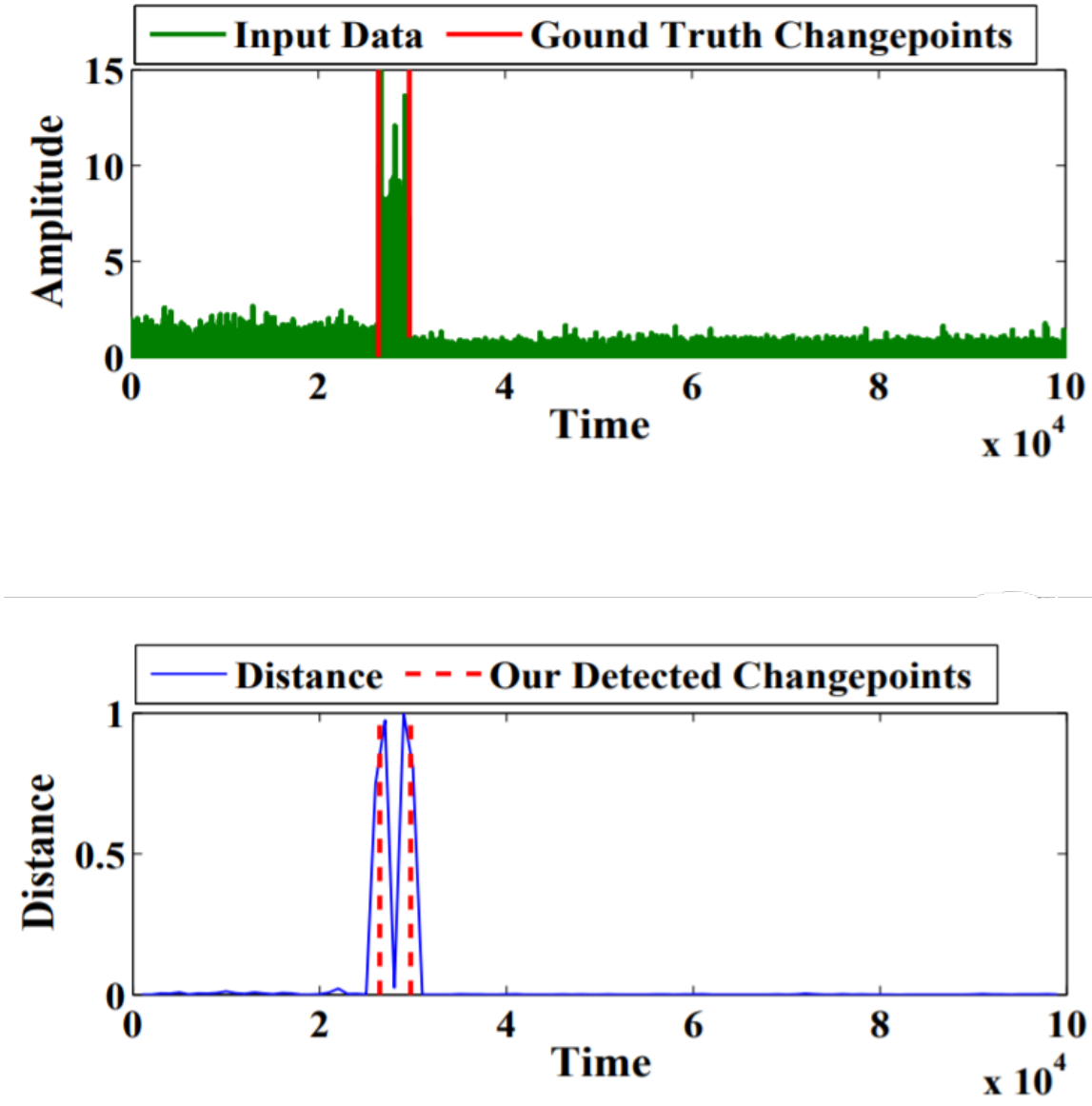
Deep Learning for Human-Specified Change Points

Breakpoints are detected when the representative features are extracted. For each feature, distance is calculated sequentially for time windows. For the t-th timestamp, the distance between the consecutive features f_t and f_{t-1} are

$$\text{Dist}_t = \frac{\|f_t - f_{t-1}\|_2}{\sqrt{\|f_t\|_2 \times \|f_{t-1}\|_2}}$$

The sequence $\{\text{Dist}_t\}_{t=1}^{T/N_w}$ describes how the features change over time. The sequence is plotted and the local maximums are the breakpoints.

Real Life Example (Deep Learning)



Summary

- Time series
- Change point detection
- Multiple change point detection with AR(1) model and evolutionary algorithm
- Real life example (multiple change points)
- Bernoulli model (CUSUM)
- Real life examples (Bernoulli CUSUM)
- Density-ratio test statistics for more general dependency structures
- Real life example AR(4) (density-ratio)
- Deep learning for human-specified change points
- Real life example (deep learning)