Adaptive forgetting factors and Average Run Length in streaming data change detection

Dean Bodenham

Imperial College London

March, 2012

Supervisors: N. Adams and N. Heard

Overview

- **Change Detection: Streaming Data**
- Two Algorithms: CUSUM and EWMA
- Performance Measures: ARL0 and ARL1
- Forgetting Factors Mean
- Assuming Normality
- Adaptive Forgetting Factors
- Experiments and Results
- Related Aspects
- Future Work

Change Detection: Streaming Data

We define a **data stream** to be a sequence of random observations $x_1, x_2, \ldots$. We assume that these observations are:

- sequential
- unpredictable **when** and **how** they change

For convenience, we treat the observations as arriving at regularly spaced intervals.

Goals:

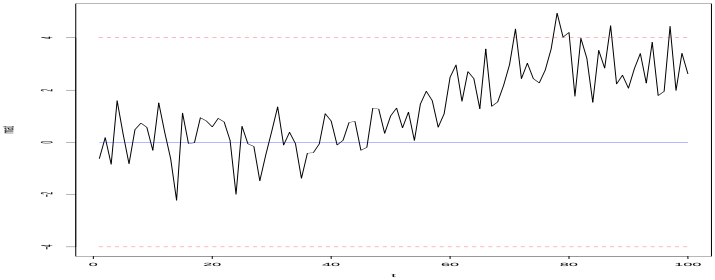
- To detect changes in the data stream **sequentially**,
- Algorithm should **restart** and **continue** after a change

We are mainly interested in **regime changes**, such as mean/variance change, e.g. $\mathrm{N}(0,1) \to \mathrm{N}(2,1)$

Control Charts

A **control chart** consists of points $z_1, z_2, \ldots$ representing a **statistic** and control limits a, b, where $a < b$. When

- $z_k \in (a, b) \Rightarrow$ process is **in-control**
- $z_k \notin (a, b) \Rightarrow$ process is **out-of-control**



We call τ the **changepoint** of the data stream if

- $z_\tau \notin (a, b)$, but
- $z_k \in (a, b)$ for all $k < \tau$

Overview

- Change detection: Streaming Data
- **Two algorithms: CUSUM and EWMA**
- Performance Measures: ARL0 and ARL1
- Forgetting Factors Mean
- Assuming Normality
- Adaptive Forgetting Factors
- Experiments and results
- Related aspects
- Future Work

CUSUM (Page, 1954)

Parameters chosen: d, B

Observations: $x_1, x_2 \ldots,$ with $\mathrm{E}[x_k] = \mu$ and $\mathrm{Var}[x_k] = \sigma^2$.

To detect an **increase** in the mean, define:

$$
\begin{aligned}
T_k &= x_k - \mu + d\sigma \\
\Rightarrow \mathrm{E}[T_k] &= d\sigma
\end{aligned}
$$

Now define the CUSUM:

$$
\begin{aligned}
S_0 &= 0 \\
S_{k+1} &= \max\{0, S_k + T_{k+1}\}
\end{aligned}
$$

A change is detected when:

$$S_k > B\sigma$$

EWMA (Roberts, 1959)

Parameters chosen: $r, L,$ ($L = 3$, usually)

Observations: $x_1, x_2 \ldots,$ as before.

To detect an **increase** in the mean, define:

$$z_0 = \mu$$
$$z_k = r x_k + (1 - r) z_{k-1}, \qquad k > 0$$

It can be shown that the standard deviation of z_k is

$$\sigma_{z_k} = \sqrt{\frac{r}{2 - r}\left[1 - (1 - r)^{2k}\right]}\sigma$$

A change is detected when:

$$z_k > \mu + L\sigma_{z_k}$$

Review of CUSUM and EWMA

Although both CUSUM and EWMA are excellent sequential change-detection algorithms, we would like an algorithm that:

- does not require knowledge of the parameters of the underlying distributions
- does not require a "subjective" **choice** of free **parameters**
  - CUSUM needs (d, B), EWMA needs (r, L)
- can operate well on a stream; does not require/**learns** new parameters after a change

Overview

- Change Detection: Streaming Data
- Two Algorithms: CUSUM and EWMA
- **Performance Measures: ARL0 and ARL1**
- Forgetting Factors Mean
- Assuming Normality
- Adaptive Forgetting Factors
- Experiments and Results
- Related Aspects
- Future Work

Performance Measures: Average Run Length (ARL)

ARL0: average number of observations until a **false alarm**.
ARL1: **average delay** in detecting a changepoint.

We would like our algorithm to have:

- High ARL0
- Low ARL1

Overview

- Change Detection: Streaming Data
- Two Algorithms: CUSUM and EWMA
- Performance Measures: ARL0 and ARL1
- **Forgetting Factor Mean**
- Assuming Normality
- Adaptive Forgetting Factors
- Experiments and Results
- Related Aspects
- Future Work

Tracking the mean

Suppose we want to monitor a stream $x_1, x_2, \ldots, x_N, \ldots$.

We could calculate the mean $\bar{x}_N$ of the first N observations as

$$\bar{x}_N = \frac{1}{N}\sum_{k=1}^{N} x_k \tag{1}$$

Or, we could calculate it **sequentially** as

$$\begin{aligned}
m_k &= m_{k-1} + x_k, & m_0 &= 0 & &\text{(mass)} & (2)\\
w_k &= w_{k-1} + 1, & w_0 &= 0 & &\text{(weight)} & (3)\\
\bar{x}_N &= \frac{m_N}{w_N} & & & &\text{(mean)} & (4)
\end{aligned}$$

However, this formulation gives **equal importance** (weight) to each observation.

Calculating the mean: Forgetting Factor

We introduce an exponential forgetting factor $\lambda \in [0, 1]$, and calculate the **forgetting factor mean** $\bar{x}_{N,\lambda}$

$$\bar{x}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \sum_{k=1}^{N} \lambda^{N-k} x_k \tag{5}$$

where

$$w_{N,\lambda} = \sum_{k=1}^{N} \lambda^{N-k} \tag{6}$$

Example: $N = 3$ and $\lambda = 0.9$:

$$\bar{x}_{3,\lambda} = \left[(0.9)^2 x_1 + (0.9) x_2 + x_3\right] \cdot \frac{1}{w_{3,\lambda}}$$
$$w_{3,\lambda} = (0.9)^2 + (0.9) + 1$$

Forgetting Factor Mean

In general:

$$\bar{x}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \left[\left(\lambda^{N-1}\right) x_1 + \left(\lambda^{N-2}\right) x_2 + \cdots + \left(\lambda\right) x_{N-1} + x_N \right]$$

The extreme cases of $\lambda = 1$ and $\lambda = 0$:

- when $\lambda = 1$, $\bar{x}_{N,\lambda} = \bar{x}_N$ (unweighted mean, no forgetting)
- when $\lambda = 0$, $\bar{x}_{N,\lambda} = x_N$ (last observation, forgets everything)

The forgetting factor $\lambda \in (0, 1)$:

- downweights early observations $(x_1, x_2, \ldots)$, and therefore
- more weight on recent observations $(\ldots, x_{N-1}, x_N)$

Forgetting Factor Mean

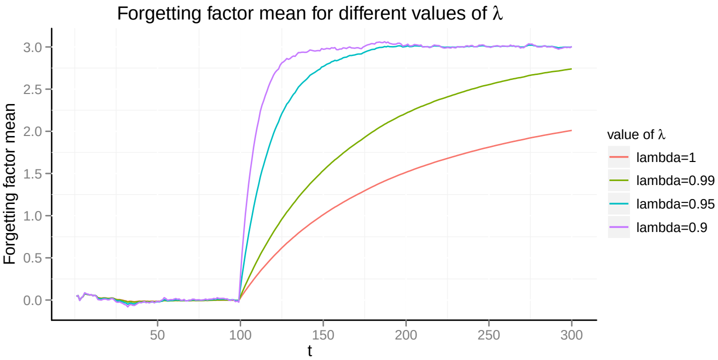
We can also define $\bar{x}_{N,\lambda}$ **sequentially** as:

$$\begin{aligned} m_{k,\lambda} &= \lambda m_{k-1,\lambda} + x_k, & m_{0,\lambda} &= 0 & &\text{(mass)} \\ w_{k,\lambda} &= \lambda w_{k-1,\lambda} + 1, & w_{0,\lambda} &= 0 & &\text{(weight)} \\ \bar{x}_{N,\lambda} &= \frac{m_{N,\lambda}}{w_{N,\lambda}} & & & &\text{(mean)} \end{aligned}$$

Compared to the unweighted mean from before:

$$\begin{aligned} m_k &= m_{k-1} + x_k, & m_0 &= 0 & &\text{(mass)} \\ w_k &= w_{k-1} + 1, & w_0 &= 0 & &\text{(weight)} \\ \bar{x}_N &= \frac{m_N}{w_N} & & & &\text{(mean)} \end{aligned}$$

Plots of the forgetting factor mean $\bar{x}_{N,\lambda}$



Data: $x_1, \ldots, x_{100} \sim \mathrm{N}(0,1)$, $x_{101}, \ldots, x_{300} \sim \mathrm{N}(3,1)$
Number of runs: 100

Overview

- Change detection: Streaming Data
- Two algorithms: CUSUM and EWMA
- Forgetting Factor Mean
- **Assuming Normality**
- Adaptive Forgetting Factors
- Experiments and results
- Related aspects
- Future Work

Control chart for $\bar{x}_{N,\lambda}$: assuming normality

If $x_1, x_2, \ldots, x_N \sim \mathrm{N}(\mu, \sigma^2)$, then

$$\bar{x}_{N,\lambda} \sim \mathrm{N}(\mu, (u_{N,\lambda})\sigma^2) \tag{7}$$

where $u_{N,\lambda}$ is a function of N and λ.

We can then calculate a confidence interval (a, b) for $\bar{x}_{N,\lambda}$ (quantile function of normal distribution). Then

- $\bar{x}_{N,\lambda} \in (a, b) \Rightarrow$ in-control
- $\bar{x}_{N,\lambda} \notin (a, b) \Rightarrow$ out-of-control

Overview

- Change detection: Streaming Data
- Two algorithms: CUSUM and EWMA
- Forgetting Factors
- Assuming Normality
- **Adaptive Forgetting Factors**
- Experiments and results
- Related aspects
- Future Work

How do we choose forgetting factor λ?

What value should we choose for λ? 0.9? 0.95? 0.8?

Same situation as CUSUM or EWMA: need to subjectively choose a parameter λ.

One approach: instead of having a fixed forgetting factor λ, we use a forgetting factor $\overrightarrow{\lambda}$ that changes after every observation

$$\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_N, \dots)$$

Adaptive Forgetting Factor Mean $\bar{x}_{N,\vec{\lambda}}$

We then define the adaptive forgetting factor (AFF) mean $\bar{x}_{N,\vec{\lambda}}$ as

$$\begin{aligned} m_{N,\vec{\lambda}} &= \lambda_{N-1} m_{N-1,\vec{\lambda}} + x_N, & m_{0,\vec{\lambda}} &= 0 \\ w_{N,\vec{\lambda}} &= \lambda_{N-1} w_{N-1,\vec{\lambda}} + 1, & w_{0,\vec{\lambda}} &= 0 \\ \bar{x}_{N,\vec{\lambda}} &= \frac{m_{N,\vec{\lambda}}}{w_{N,\vec{\lambda}}} \end{aligned} \tag{8}$$

There are non-recursive definitions, e.g.

$$m_{N,\vec{\lambda}} = \sum_{k=1}^{N} \left[\left(\prod_{p=k}^{N-1} \lambda_p \right) x_k \right] \tag{9}$$

Adaptive Forgetting Factor Mean $\overrightarrow{\lambda}$

To clarify the difference, below are:

The fixed forgetting factor (FFF) mean $\bar{x}_{3,\lambda}$:

$$\bar{x}_{3,\lambda} = \frac{1}{w_{3,\lambda}} \left[\lambda^2 x_1 + \lambda x_2 + x_3 \right] \tag{10}$$

The AFF mean $\bar{x}_{3,\overrightarrow{\lambda}}$:

$$\bar{x}_{3,\overrightarrow{\lambda}} = \frac{1}{w_{3,\overrightarrow{\lambda}}} \left[\lambda_2 \lambda_1 x_1 + \lambda_2 x_2 + x_3 \right] \tag{11}$$

Control chart: Assuming normality

As before, if $x_1, x_2, \ldots, x_N \sim \mathrm{N}(\mu, \sigma^2)$, then

$$\bar{x}_{N,\vec{\lambda}} \sim \mathrm{N}(\mu, (u_{N,\vec{\lambda}})\sigma^2) \tag{12}$$

where $u_{N,\vec{\lambda}}$ is defined recursively.

Again, we calculate a confidence interval (a, b) for $\bar{x}_{N,\vec{\lambda}}$, and

- $\bar{x}_{N,\vec{\lambda}} \in (a, b) \Rightarrow$ in-control
- $\bar{x}_{N,\vec{\lambda}} \notin (a, b) \Rightarrow$ out-of-control

Updating $\overrightarrow{\lambda}$: $\lambda_N \to \lambda_{N+1}$

We update our AFF $\lambda_N \to \lambda_{N+1}$ in three steps:

1. Choose a cost function C_{N+1} we would like to minimize, e.g.

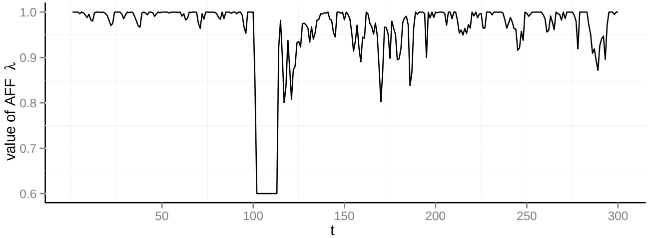
$$C_{N+1} = [x_{N+1} - \bar{x}_{N,\overrightarrow{\lambda}}]^2 \tag{13}$$

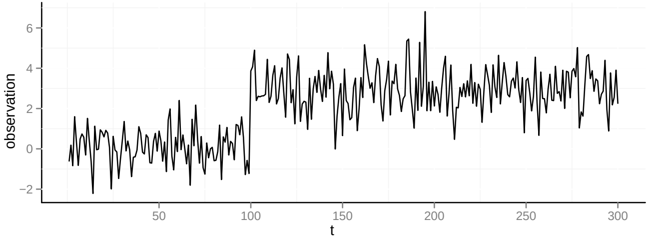
2. Find $\frac{\partial}{\partial \overrightarrow{\lambda}} C_{N+1}$ (later)
3. Update $\overrightarrow{\lambda}$:

$$\lambda_{N+1} = \lambda_N - \eta \frac{\partial}{\partial \overrightarrow{\lambda}} C_{N+1} \tag{14}$$

(One-step gradient descent, $\eta << 1$)

Adaptive Forgetting Factor: one simulation



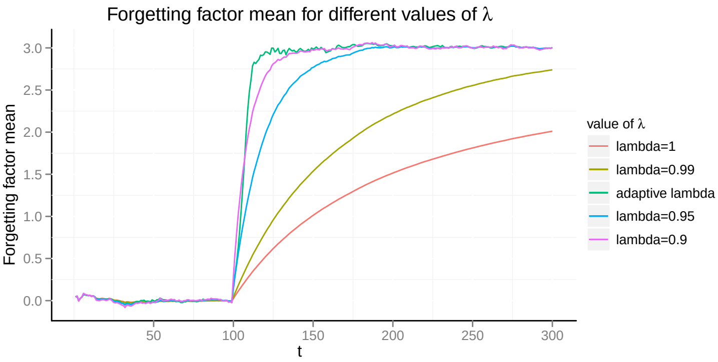


Data: $x_1, \ldots, x_{100} \sim \mathrm{N}(0, 1)$, $x_{101}, \ldots, x_{300} \sim \mathrm{N}(3, 1)$

Adaptive Forgetting Factor: on average

Data: $x_1, \ldots, x_{100} \sim \mathrm{N}(0, 1)$, $x_{101}, \ldots, x_{300} \sim \mathrm{N}(3, 1)$
Number of runs: 100

Plots of $\bar{x}_{N,\lambda}$ and $\bar{x}_{N,\overrightarrow{\lambda}}$



Data: $x_1, \ldots, x_{100} \sim \mathrm{N}(0,1)$, $x_{101}, \ldots, x_{300} \sim \mathrm{N}(3,1)$
Number of runs: 100

Everything is sequential!

There are sequential update equations for

- $\bar{x}_{N,\overrightarrow{\lambda}}$, the AFF mean,
- $u_{N,\overrightarrow{\lambda}}$, (used for the confidence interval of normal $\bar{x}_{N,\overrightarrow{\lambda}}$),
- $\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda}}$, (long derivation), and therefore
- $\frac{\partial}{\partial \overrightarrow{\lambda}} F(\bar{x}_{N,\overrightarrow{\lambda}})$, for some function F,
  e.g. $F(\bar{x}_{N,\overrightarrow{\lambda}}) = [x_{N+1} - \bar{x}_{N,\overrightarrow{\lambda}}]^2$

Overview

- Change detection: Streaming Data
- Two algorithms: CUSUM and EWMA
- Forgetting Factors
- Assuming Normality
- Adaptive Forgetting Factors
- **Experiments and results**
- Related aspects
- Future Work

Experiments and Results

Usual comparison: fix ARL0 and then compare ARL1.

Instead, we are just trying to show that the pairs are roughly comparable - the advantage of the FF methods is that they will not rely (too much) on chosen parameters.

Algorithm	Parameters	Values	ARL0	ARL1
CUSUM	$(d,\ B)$	(0.25, 8)	382.17	2.97
EWMA	$(r,\ L)$	(0.2, 3)	618.09	2.40
FFF	$(\lambda,\ p)$	(0.95, 0.99)	488.67	3.41
AFF	$(\eta,\ p)$	(0.01, 0.99)	761.53	3.74

Table: ARL0: number of observations = 100000, ARL1: 10000 runs of $N(0,1) \to N(3,1)$ at $\tau = 50$

Note: $p = 0.99$ indicates we are using a 99% confidence interval.

Related aspects

- Combining AFF and FFF: **tuned** forgetting factor
  - using a burn-in period and the AFF algorithm to tune fixed λ
- Forgetting factor variance
- Non-parametric – Chebyshev's Inequality

Future Work

- Self-starting/unsupervised restarting
  - Estimation of parameters (μ, σ^2) during burn-in
  - Non-parametric methods
- Multivariate case
- AFF - different cost functions, choice of η

References

Page, E. S. (1954). Continuous Inspection Schemes, Biometrika, Vol. 41, No. 1/2 (Jun., 1954), pp. 100-115

Roberts, S. W. (1959). Control Chart Tests Based on Geometric Moving Averages, Technometrics, Vol. 1, No. 3 (Aug., 1959), pp. 239-250

Åström, K. J. and Wittenmark, B. (1973). On self tuning regulators, Automatica, Vol. 9, No. 2, pp. 185–199, Elsevier

Åström, K. J.,Borisson, U., Ljung, L. and Wittenmark, B. (1977). Theory and applications of self-tuning regulators, Automatica, Vol. 13, No. 5, pp. 457–476, Elsevier

Thanks to N. Adams, N. Heard, G. Ross, and C. Anagnostopoulos.
Graphs produced using ggplot2.

Derivatives with respect to $\overrightarrow{\lambda}$

Sequential definition of $m_{N,\overrightarrow{\lambda}}$:

$$m_{N,\overrightarrow{\lambda}} = \lambda_{N-1} m_{N-1,\overrightarrow{\lambda}} + x_N \tag{15}$$

Non-recursive definition of $m_{N,\overrightarrow{\lambda}}$

$$m_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \left[\left(\prod_{p=k}^{N-1} \lambda_p \right) x_k \right] \tag{16}$$

We consider an ϵ-perturbation around $\overrightarrow{\lambda}$:

$$m_{N,\overrightarrow{\lambda}+\epsilon} = \sum_{k=1}^{N} \left[\left(\prod_{p=k}^{N-1} (\lambda_p + \epsilon) \right) x_k \right] \tag{17}$$

and define

$$\frac{\partial}{\partial \overrightarrow{\lambda}} m_{N,\overrightarrow{\lambda}} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left[m_{N,\overrightarrow{\lambda}+\epsilon} - m_{N,\overrightarrow{\lambda}} \right] \tag{18}$$

using “first principles”.

Derivatives with respect to $\overrightarrow{\lambda}$

Main part is to show

$$m_{N,\overrightarrow{\lambda}+\epsilon} = \sum_{k=1}^{N}\left[\left(\prod_{p=k}^{N-1}(\lambda_p+\epsilon)\right)x_k\right]$$
$$= m_{N,\overrightarrow{\lambda}} + \epsilon\Delta_{N,\overrightarrow{\lambda}} + O(\epsilon^2) \tag{19}$$

Then

$$\frac{\partial}{\partial\overrightarrow{\lambda}} m_{N,\overrightarrow{\lambda}} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[m_{N,\overrightarrow{\lambda}+\epsilon} - m_{N,\overrightarrow{\lambda}}\right]$$
$$= \lim_{\epsilon\to 0}\left[\Delta_{N,\overrightarrow{\lambda}} + O(\epsilon)\right]$$
$$= \Delta_{N,\overrightarrow{\lambda}}$$

follows easily.

Derivatives with respect to $\overrightarrow{\lambda}$

We can define $\frac{\partial}{\partial \overrightarrow{\lambda}} m_{N,\overrightarrow{\lambda}}$ sequentially:

$$\begin{aligned} \Delta_{N,\overrightarrow{\lambda}} &= \frac{\partial}{\partial \overrightarrow{\lambda}} m_{N,\overrightarrow{\lambda}} \\ \Delta_{1,\overrightarrow{\lambda}} &= 0 \\ \Delta_{N+1,\overrightarrow{\lambda}} &= \lambda_N \Delta_{N,\overrightarrow{\lambda}} + m_{N,\overrightarrow{\lambda}} \end{aligned} \tag{20}$$

Similar for $\frac{\partial}{\partial \overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}}$.

With sequential equations for the derivatives of $m_{N,\overrightarrow{\lambda}}$ and $w_{N,\overrightarrow{\lambda}}$, we get sequential equations for

$$\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_N = \frac{\partial}{\partial \overrightarrow{\lambda}} \left[\frac{m_{N,\overrightarrow{\lambda}}}{w_{N,\overrightarrow{\lambda}}} \right]$$

Using Chebyshev's inequality

Suppose X is a random variable with known expected value $\mu = \mathrm{E}[X]$ and variance $\sigma^2 = \mathrm{Var}[X]$. Then for any real number $k > 0$ we have Chebyshev's inequality

$$\Pr\left(|X - \mathrm{E}[X]| \geq k\sigma\right) \leq \frac{1}{k^2}. \tag{21}$$

Stream $x_1, x_2, \ldots,$ with $\mathrm{E}[x_k] = \mu$, $\mathrm{Var}[x_k] = \sigma^2$.

For a 99% confidence interval for $\bar{x}_{N,\lambda}$, choose $k = 10\sqrt{2}$ to get

$$C = \left(\mu - 10\sqrt{2}\sigma\sqrt{(u_{N,\lambda})}, \mu + 10\sqrt{2}\sigma\sqrt{(u_{N,\lambda})}\right)$$