The CuSum algorithm - a small review

Pierre Granjon

The CUSUM algorithm
a small review

Pierre Granjon
GIPSA-lab

March 13, 2014

Contents

Chapter 1

The CUSUM algorithm

Lai [Lai 1995] and Basseville [Basseville 1993] give a full treatment of the field, from the simplest to more complicated sequential detection procedures (from Shewhart control charts [Shewhart 1931] to GLR algorithm first given in [Lorden 1971]).

1.1 Algorithm

It is Page who first proposes different forms of the *cumulative sum* (CUSUM) algorithm in [Page 1954a]:

- direct or recursive forms.
- one- or two-sided forms,

1.1.1 The problem

Let $\mathfrak{X}[n]$ be a discrete random signal with independent and identically distributed samples. Each of them follows a probability density function (PDF) $p\left(x[n], \theta\right)$ depending on a deterministic parameter θ (for example the mean $\mu_{\mathfrak{X}}$ or the variance $\sigma^2_{\mathfrak{X}}$ of $\mathfrak{X}[n]$). This signal may contain one abrupt change occurring at the change time n_c. This abrupt change is modeled by an instantaneous modification of the value of θ occurring at the change time n_c. Therefore $\theta = \theta_0$ before n_c and $\theta = \theta_1$ from n_c to the current sample.
Under these assumptions, the whole PDF of the signal $p_{\mathfrak{X}}$ observed between the first sample $x[0]$ and the current one $x[k]$ can take two different forms.

- Under the *no change* hypothesis ($\mathcal{H}_0$), the PDF of $\mathfrak{X}[n]$ is given by:

$$p_{\mathfrak{X}|\mathcal{H}_0} = \prod_{n=0}^{k} p\left(x[n], \theta_0\right). \tag{1.1}$$

- Under the *one change* hypothesis ($\mathcal{H}_1$), this PDF becomes:

$$p_{\mathfrak{X}|\mathcal{H}_1} = \prod_{n=0}^{n_c-1} p\left(x[n], \theta_0\right) \prod_{n=n_c}^{k} p\left(x[n], \theta_1\right). \tag{1.2}$$

In this problem, the PDF of each sample $p\left(x[n], \theta\right)$ and the values of the parameter before (θ_0) and after (θ_1) the abrupt change are supposed to be known. Finally, the only unknowns to be determined are:

- the lack or occurrence of an abrupt change between $n = 0$ and $n = k$,
- the value of the possible change time n_c.

In order to develop the desired algorithm, we follow an on-line approach. This means that the abrupt change possibly contained in the past of the signal has to be sequentially detected sample after sample. Therefore, at each new sample $x[k]$, one of the two previous hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ has first to be *decided*. Then, in the case a change has been detected ($\mathcal{H}_1$ decided), the change time has to be estimated thanks to an estimator $\widehat{n_c}$. The general form of this sequential algorithm is given in algorithm 1.

initialization
  if necessary
end
while *the algorithm is not stopped* **do**
  measure the current sample $x[k]$
  decide between $\mathcal{H}_0$ (no change) and $\mathcal{H}_1$ (one change)
  if $\mathcal{H}_1$ *decided* **then**
    store the detection time $n_d \leftarrow k$
    estimate the change time n_c
    stop or reset the algorithm
  end
end

Algorithm 1: general form of a sequential change detection algorithm.

Two important steps appear in this algorithm :

detection step: How to decide between $\mathcal{H}_0$ and $\mathcal{H}_1$?

estimation step: How to efficiently estimate the change time n_c?

These steps are further detailed in the two following sections.

1.1.2 The different steps

Detection step

The problem here is to decide between two possible hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ from the measured samples $x[0], \ldots, x[k]$, which is also termed a binary hypothesis

testing problem. The solution, given by the detection theory, is to use the so-called likelihood ratio test [Kay 1998, chap. 3]:

test 1. *Let the log-likelihood ratio $L_{\mathfrak{x}}$ be defined by:*

$$L_{\mathfrak{x}} = \ln\left(\frac{p_{\mathfrak{x}|\mathcal{H}_1}}{p_{\mathfrak{x}|\mathcal{H}_0}}\right). \tag{1.3}$$

Then, decide $\mathcal{H}_1$ if $L_{\mathfrak{x}} > h$ (else $\mathcal{H}_0$), where h is a threshold set by the user.

In our case, the theoretical expression of $L_{\mathfrak{x}}$ at the current sample $x[k]$ is obtained by reporting (1.1) and (1.2) in (1.3):

$$L_{\mathfrak{x}}\left[k, n_c\right] = \ln\left(\frac{p_{\mathfrak{x}|\mathcal{H}_1}\left[k, n_c\right]}{p_{\mathfrak{x}|\mathcal{H}_0}[k]}\right) = \sum_{n=n_c}^{k} \ln\left(\frac{p\left(x[n], \theta_1\right)}{p\left(x[n], \theta_0\right)}\right). \tag{1.4}$$

Unfortunately, this quantity can not be calculated since it depends on the unknown change time n_c. Once again, the detection theory leads to an efficient solution which consists of replacing all the unknowns in $L_{\mathfrak{x}}$ by their maximum likelihood estimates. Such a test is termed the generalized likelihood ratio test, and is given by [Kay 1998, chap. 6]:

test 2. *Let the generalized log-likelihood ratio $G_{\mathfrak{x}}$ be defined by:*

$$\begin{aligned} G_{\mathfrak{x}}[k] &= \max_{1 \leq n_c \leq k} L_{\mathfrak{x}}\left[k, n_c\right] \\ &= \max_{1 \leq n_c \leq k} \sum_{n=n_c}^{k} \ln\left(\frac{p\left(x[n], \theta_1\right)}{p\left(x[n], \theta_0\right)}\right). \end{aligned} \tag{1.5}$$

Then, decide $\mathcal{H}_1$ if $G_{\mathfrak{x}}[k] > h$ (else $\mathcal{H}_0$), where h is a threshold set by the user.

The quantity defined in (1.5) can now be calculated at each new sample and is thus used to decide between $\mathcal{H}_0$ and $\mathcal{H}_1$. It is also termed a *decision function*.

Estimation step

Once $\mathcal{H}_1$ has been decided and an abrupt change has been detected, the problem is to efficiently estimate the change time n_c from the measured samples $x[0], \ldots, x[k]$. One way to solve this problem is to use its *maximum likelihood estimate*, which is the value of n_c maximizing the likelihood $p_{\mathfrak{x}|\mathcal{H}_1}\left[k, n_c\right]$ [Kay 1993, chap. 7]:

$$\begin{aligned} \widehat{n_c} &= \arg\max_{1 \leq n_c \leq k} p_{\mathfrak{x}|\mathcal{H}_1}[k, n_c] = \arg\max_{1 \leq n_c \leq k} L_{\mathfrak{x}}\left[k, n_c\right] \\ &= \arg\max_{1 \leq n_c \leq k} \sum_{n=n_c}^{k} \ln\left(\frac{p\left(x[n], \theta_1\right)}{p\left(x[n], \theta_0\right)}\right). \end{aligned} \tag{1.6}$$

1.1.3 The whole algorithm

Direct form

In order to obtain a simplified form of the whole algorithm, let's define from (1.4) the *instantaneous log-likelihood ratio* at time n by:

$$s[n] = L_{\mathfrak{X}}[n, n] = \ln\left(\frac{p\left(x[n], \theta_1\right)}{p\left(x[n], \theta_0\right)}\right), \tag{1.7}$$

and its cumulative sum from 0 to k:

$$S[k] = \sum_{n=0}^{k} s[n]. \tag{1.8}$$

From (1.7) and (1.8), the log-likelihood ratio $L_{\mathfrak{X}}[k, n_c]$ defined in (1.4) can be rewritten as:

$$L_{\mathfrak{X}}\left[k, n_c\right] = S[k] - S[n_c - 1]. \tag{1.9}$$

Reporting this expression in (1.5) and (1.6), we obtain new expressions for the decision function $G_{\mathfrak{X}}[k]$ and for the change time estimate $\widehat{n_c}[k]$:

$$G_{\mathfrak{X}}[k] = S[k] - \min_{1 \leq n_c \leq k} S[n_c - 1] \tag{1.10}$$

$$\widehat{n_c} = \arg\min_{1 \leq n_c \leq k} S[n_c - 1]. \tag{1.11}$$

Equation (1.10) shows that the decision function $G_{\mathfrak{X}}[k]$ is the current value of the cumulative sum $S[k]$ minus its current minimum value. Equation (1.11) shows that the change time estimate is the time following the current minimum of the cumulative sum. Therefore, each step composing the whole algorithm relies on the same quantity: the cumulative sum $S[k]$ defined through (1.7) and (1.8). This explains the name of *cumulative sum* or *CUSUM* algorithm. By using equations (1.7), (1.8), (1.10) and (1.11) into algorithm 1, the direct form of the CUSUM algorithm 2 is obtained.

Recursive form

The previous algorithm can easily be rewritten in a recursive manner. Indeed, the cumulative sum (1.8) can be calculated through the simple recursive formula:

$$S[k] = S[k-1] + s[k]. \tag{1.12}$$

Moreover, as shown in [Basseville 1993, chap. 2], since the decision function is compared to a positive threshold h, it can be rewritten as:

$$G_{\mathfrak{X}}[k] = \left\{G_{\mathfrak{X}}[k-1] + s[k]\right\}^{+}, \tag{1.13}$$

where $\{z\}^{+} = \sup(z, 0)$. Finally, by using (1.12) and (1.13) into the direct form algorithm 2, the recursive form of the CUSUM algorithm 3 is obtained. Clearly, the obtained algorithm is very simple and very interesting to use in real-time or on-line applications.

initialization
  set the detection threshold $h > 0$
  $k = 0$
end
while *the algorithm is not stopped* **do**
  measure the current sample $x[k]$
  $s[k] = \ln\left(\frac{p(x[k],\theta_1)}{p(x[k],\theta_0)}\right)$
  $S[k] = \sum_{n=0}^{k} s[n]$
  $G_{\mathfrak{X}}[k] = S[k] - \min_{1 \leq n_c \leq k} S[n_c - 1]$
  if $G_{\mathfrak{X}}[k] > h$ **then**
    $n_d \leftarrow k$
    $\widehat{n_c} = \arg\min_{1 \leq n_c \leq k} S[n_c - 1]$
    stop or reset the algorithm
  end
  $k = k + 1$
end

Algorithm 2: CUSUM algorithm, direct form.

initialization
  set the detection threshold $h > 0$
  $S[-1] = G_{\mathfrak{X}}[-1] = 0$
  $k = 0$
end
while *the algorithm is not stopped* **do**
  measure the current sample $x[k]$
  $s[k] = \ln\left(\frac{p(x[k],\theta_1)}{p(x[k],\theta_0)}\right)$
  $S[k] = S[k-1] + s[k]$
  $G_{\mathfrak{X}}[k] = \{G_{\mathfrak{X}}[k-1] + s[k]\}^{+}$
  if $G_{\mathfrak{X}}[k] > h > 0$ **then**
    $n_d \leftarrow k$
    $\widehat{n_c} = \arg\min_{1 \leq n_c \leq k} S[n_c - 1]$
    stop or reset the algorithm
  end
  $k = k + 1$
end

Algorithm 3: CUSUM algorithm, recursive form.

1.1.4 Example

Let us consider the particular case where the signal $\mathfrak{X}[n]$ is constituted of independent and identically distributed samples following a Gaussian distribution with mean value $\mu_{\mathfrak{X}}$ and variance $\sigma^2_{\mathfrak{X}}$. This signal is supposed to undergo a possible change in the mean at a change time n_c. The changing parameter θ is then $\mu_{\mathfrak{X}}$, which takes the values $\mu_{\mathfrak{X}0}$ before and $\mu_{\mathfrak{X}1}$ after the change. Therefore, the theoretical expression of the PDF followed by each sample is:

$$p\left(x[n], \mu_{\mathfrak{X}_j}\right) = \frac{1}{\sigma_{\mathfrak{X}}\sqrt{2\pi}} e^{-\frac{\left(x[n]-\mu_{\mathfrak{X}_j}\right)^2}{2\sigma^2_{\mathfrak{X}}}},$$

where $j = 0$ before and $j = 1$ after the change.
A particular realization $x[n]$ of $\mathfrak{X}[n]$ is represented on top of Fig. 1.1 with $\mu_{\mathfrak{X}0} = 0$, $\mu_{\mathfrak{X}1} = 1$, $\sigma^2_{\mathfrak{X}} = 1$ and $n_c = 1000$. It is clear from this curve that a simple threshold applied to $x[n]$ is insufficient to efficiently detect the presence of an abrupt change and estimate the change time n_c.
The CUSUM algorithm is applied to this signal by using its recursive form 3, and a detection threshold set to $h = 100$. The values of $\mu_{\mathfrak{X}0}$, $\mu_{\mathfrak{X}1}$ and $\sigma^2_{\mathfrak{X}}$ are supposed to be known, contrarily to n_c. Therefore, the instantaneous log-likelihood ratio $s[n]$ can be calculated whatever the time n through the expression:

$$s[n] = \frac{\mu_{\mathfrak{X}1} - \mu_{\mathfrak{X}0}}{\sigma^2_{\mathfrak{X}}} \left(x[n] - \frac{\mu_{\mathfrak{X}1} + \mu_{\mathfrak{X}0}}{2} \right). \tag{1.14}$$

Fig. 1.1 shows the typical behaviour of this algorithm. As expected, the decision function $G_{\mathfrak{X}}[n]$ stays close to 0 before and grows continuously after the change time. The presence of the change is detected when $G_{\mathfrak{X}}[n] > h$ at $n_d = 1201$ (see the red line), leading to a detection delay of $n_d - n_c = 201$ samples. Once the change has been detected, the time following the current minimum of the cumulative sum $S[n]$ gives the estimated change time $\widehat{n_c} = 1001$. Consequently, the estimation error of the change time $|n_c - \widehat{n_c}| = 1$ is very small for this example.

1.2 Performance

1.2.1 Criteria

Several criteria have been proposed to measure the performance of change detection algorithms, but the following ones are the most popular.

Average run length function The *average run length* (ARL) function proposed in [Page 1954a] is defined as the expected number of samples before an action is taken:

$$\mathrm{ARL} = \mathrm{E}_\theta\left[N_d\right], \tag{1.15}$$

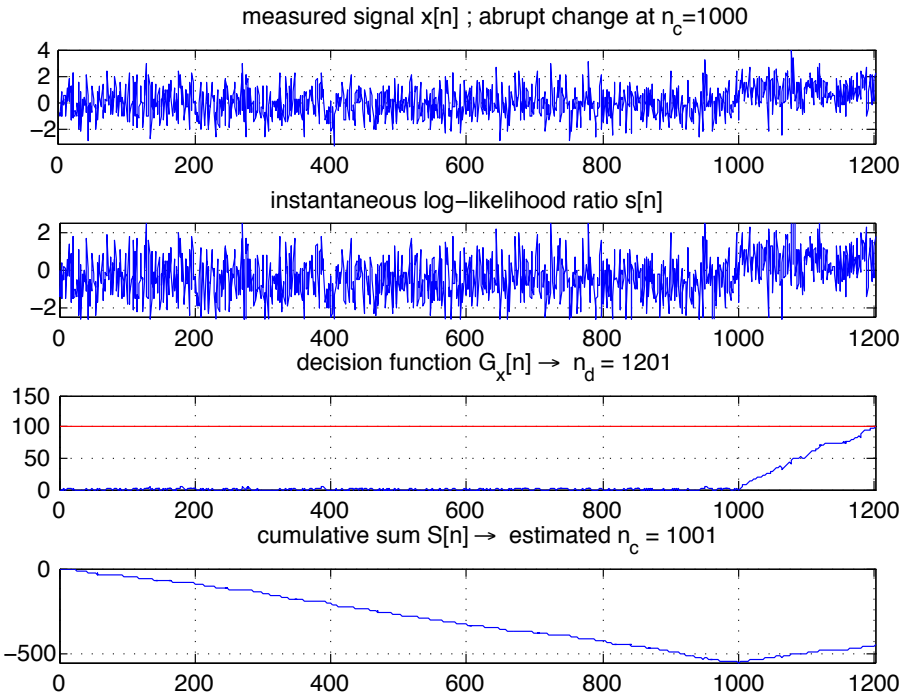


Figure 1.1: typical behaviour of the CUSUM algorithm in the case of an iid Gaussian signal with a change in the mean at time $n_c = 1000$.

where N_d is the detection time of the change detection algorithm, and the parameter θ is assumed constant for all the signal samples. More particularly, the ARL function takes two interesting values with respect to θ:

- If $\theta = \theta_0$, (1.15) becomes $\mathrm{ARL}_0 = \mathrm{E}_{\theta_0}[N_d]$, the ARL to false alarm. This quantity corresponds to the expected number of samples before a false alarm is signaled, and can be viewed as the average time between two false detections. Obviously, this quantity has to be as large as possible to minimize the rate of false detections.
- If $\theta = \theta_1$, (1.15) becomes $\mathrm{ARL}_1 = \mathrm{E}_{\theta_1}[N_d]$, the ARL to change detection. In this case, it corresponds to the expected number of samples before the detection of a change, and is similar to an average detection delay. Therefore, this quantity has to be as small as possible to minimize the reaction time of the algorithm.

Worst case detection delay This criterion has been first proposed in [Lorden 1971]. Let first define the conditional expected detection delay by $\mathrm{E}_{\mathcal{H}_1}\left[(N_d - n_c + 1)^+ \,|\mathfrak{X}[0], \ldots, \mathfrak{X}[n_c - 1]\right]$, where the expectation is taken under

the one change hypothesis $\mathcal{H}_1$ (PDF given by (1.2)). A minimax performance criterion is given by its supremum taken over $(n_c, \mathfrak{X}[0], \ldots, \mathfrak{X}[n_c-1])$:

$$\overline{\mathrm{E}}_{\mathcal{H}_1}[N_d] = \sup_{n_c \geq 0} \left\{ \text{ess sup } \mathrm{E}_{\mathcal{H}_1} \left[(N_d - n_c + 1)^+ \,|\mathfrak{X}[0], \ldots, \mathfrak{X}[n_c-1] \right] \right\}. \quad (1.16)$$

This quantity is the *worst case detection delay* of the change detection algorithm. It obviously has to be as small as possible in order to minimize the quickness of reaction to an abrupt change in the signal.

1.2.2 Optimality

Page uses ARL_0 and ARL_1 in [Page 1954a] to conclude on the good the performance of the CUSUM algorithm, but gives no result about its optimality.
Next, Lorden [Lorden 1971] shows the asymptotic optimality of the CUSUM algorithm with respect to the worst case detection delay. More precisely, let the detection threshold h (see algorithm 2 or 3) be so chosen that the ARL to false alarm verifies $\mathrm{ARL}_0 \geq \gamma > 0$. Clearly, this condition is equivalent to limit the rate of false detections by a given maximum value. When $\gamma \to \infty$, the CUSUM algorithm minimizes the worst case detection delay $\overline{\mathrm{E}}_{\mathcal{H}_1}[N_d]$. Moreover, the value of this delay can then be approximated by:

$$\overline{\mathrm{E}}_{\mathcal{H}_1}[N_d] \sim \frac{\ln \gamma}{\mathrm{I}\left(p_{\theta_0}, p_{\theta_1}\right)}, \quad (1.17)$$

where $\mathrm{I}\left(p_{\theta_0}, p_{\theta_1}\right) = \mathrm{E}_{\theta_1}\left[\ln\left(\frac{p(\mathfrak{X}[n],\theta_1)}{p(\mathfrak{X}[n],\theta_0)}\right)\right]$ denotes the Kullback-Leibler information number.
Later, Moustakides [Moustakides 1986] and Ritov [Ritov 1990] generalize this result to the non asymptotic case (for finite ARL_0) by using non-Bayesian and Bayesian approaches.

1.3 Practical considerations

The goal of this section is to mention the main problems encountered when the CUSUM algorithm is used in practice, and provide some possible solutions.

1.3.1 Dealing with unknowns

Section 1.1 shows that the CUSUM algorithm entirely relies on the instantaneous log-likelihood ratio $s[n]$ defined by Eq. (1.7). This quantity depends on the probability density function of the signal samples, and thus on its different parameters. For example, the case of a change in the mean in a iid Gaussian signal studied in paragraph 1.1.4 leads to Eq. (1.14), where $s[n]$ depends on the mean values before and after the change $\mu_{\mathfrak{X}_0}$ and $\mu_{\mathfrak{X}_1}$ and on the variance $\sigma^2_{\mathfrak{X}}$. Until now, all these parameters were supposed known, which is a quite unrealistic assumption for practical applications. In the case where different

parameters are unknown, the online evaluation of $s[n]$ becomes impossible and the CUSUM algorithm cannot be employed.
To overcome this problem, the optimal method consists of using the generalized likelihood ratio test principle by replacing in $s[n]$ all the unknown parameters by their maximum likelihood estimates. This leads to the generalized likelihood ratio or GLR algorithm initially presented in [Lorden 1971]. Unfortunately, this optimal algorithm cannot be written in a recursive manner and its complexity grows with the number of available samples, contrarily to the CUSUM algorithm. This explains that the GLR algorithm cannot be used for online applications.
Another possibility is to preserve the simple and recursive structure of the CUSUM algorithm, even if this leads to suboptimal algorithms. Following this point of view, different situations may be encountered:

θ_1 **unknown:** This is the most common practical case. Indeed, in most applications the true value of the changing parameter after change is not precisely known. In that case, the usual solution is to use the classical CUSUM algorithm where θ_1 is *a priori* set by the user, *i.e.* employed as an additional parameter. A logical and efficient way to set this parameter is to choose the most likely value that θ_1 should take after the change. The precision of this setting clearly depends on the amount of *a priori* information the user has about the signal. The resulting algorithm is generally suboptimal, but results of section 1.2 show that it is still optimal to sequentially detect changes from θ_0 to the chosen value of θ_1.

θ_0 **and/or constant parameters unknown:** In that case, the situation is not as critical as the previous one and a simple solution exists. Indeed, these unknowns can be replaced by their maximum likelihood estimates using the available samples at time k, *i.e.* $x[0], \ldots, x[k]$. Moreover, efficient implementations can be obtained by using recursive estimators. Finally, the resulting algorithms use the generalized likelihood ratio test principle and are nearly optimal.

As an example, let's once again consider the case studied in paragraph 1.1.4 of a change in the mean in an iid Gaussian signal. $\mu_{\mathfrak{X}_1}$ is usually rewritten with respect to $\mu_{\mathfrak{X}_0}$ as:

$$\mu_{\mathfrak{X}_1} = \mu_{\mathfrak{X}_0} + \delta, \tag{1.18}$$

where δ is the change magnitude of the mean value.
By using (1.18) into (1.14), the corresponding instantaneous log-likelihood ratio then becomes:

$$s[n] = \frac{\delta}{\sigma^2_{\mathfrak{X}}} \left(x[n] - \mu_{\mathfrak{X}_0} - \frac{\delta}{2} \right). \tag{1.19}$$

This relation shows that $s[n]$ depends on the mean value before change $\mu_{\mathfrak{X}_0}$, the constant variance of the signal $\sigma^2_{\mathfrak{X}}$ and the change magnitude δ. If all these parameters are unknown, the two strategies described in this paragraph have to be jointly applied:

δ **unknown:** replace the true value of δ in (1.19) by a parameter $\tilde{\delta}$ set to the *a priori* most likely change magnitude,

$\mu_{\mathfrak{X}_0}$ **and** $\sigma^2_{\mathfrak{X}}$ **unknown:** replace the true values of $\mu_{\mathfrak{X}_0}$ and $\sigma^2_{\mathfrak{X}}$ in (1.19) by their maximum likelihood estimates $\widehat{\mu_{\mathfrak{X}_0}}$ and $\widehat{\sigma^2_{\mathfrak{X}}}$ at the current sample.

This finally leads to algorithm 4, where two parameters have now to be *a priori* set: the detection threshold h and the change magnitude $\tilde{\delta}$.

initialization
  set $\tilde{\delta}$ to the most likely change magnitude
  set the detection threshold $h > 0$
  $S[-1] = G_{\mathfrak{X}}[-1] = 0$
  initialize the estimators $\widehat{\mu_{\mathfrak{X}_0}}$ and $\widehat{\sigma^2_{\mathfrak{X}}}$
  $k = 0$
end
while *the algorithm is not stopped* **do**
  measure the current sample $x[k]$
  calculate the current estimates $\widehat{\mu_{\mathfrak{X}_0}}[k]$ and $\widehat{\sigma^2_{\mathfrak{X}}}[k]$
  $s[k] = \frac{\tilde{\delta}}{\widehat{\sigma^2_{\mathfrak{X}}}[k]} \left(x[k] - \widehat{\mu_{\mathfrak{X}_0}}[k] - \frac{\tilde{\delta}}{2} \right)$
  $S[k] = S[k-1] + s[k]$
  $G_{\mathfrak{X}}[k] = \{G_{\mathfrak{X}}[k-1] + s[k]\}^+$
  if $G_{\mathfrak{X}}[k] > h > 0$ **then**
    $n_d \leftarrow k$
    $\widehat{n_c} = \arg\min_{1 \leq n_c \leq k} S[n_c - 1]$
    stop or reset the algorithm
  end
  $k = k + 1$
end

Algorithm 4: suboptimal CUSUM algorithm, jump in the mean - iid Gaussian case.

In order to compare optimal and suboptimal CUSUM algorithms, algorithm 4 is applied to the signal previously used in section 1.1.4, with the same detection threshold $h = 100$ and a change magnitude perfectly set to $\tilde{\delta} = 1$. Results, illustrated in Fig. 1.2, show that the two algorithms have the same general behavior, but a slightly more important detection delay is obtained with the suboptimal algorithm.
Finally, this methodology leads to a suboptimal algorithm having the same global complexity and behavior as its optimal version obtained in section 1.1.3, and reaching almost similar performance. The main difference is that this algorithm necessitates the setting of one additional parameter, the change magnitude, thanks to *a priori* information the user may have concerning abrupt changes occurring in the signal.

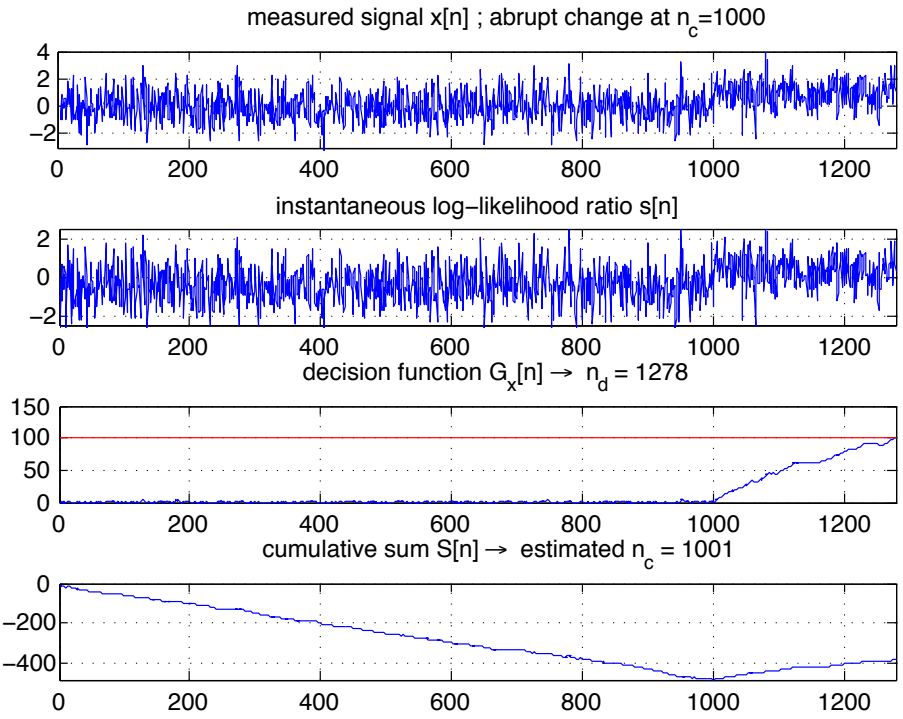


Figure 1.2: typical behaviour of the suboptimal CUSUM algorithm in the case of an iid Gaussian signal with a change in the mean at time $n_c = 1000$.

1.3.2 Setting the parameters

The CUSUM algorithm have several tuning parameters the user has to correctly set:

- in any case the detection threshold h,
- in case of the previous suboptimal CUSUM algorithm, θ_1 or equivalently the change magnitude.

the change magnitude The previous section shows that the user must have *a priori* knowledge about the signal to correctly set this parameter. Indeed, an efficient setting for the change magnitude is the *a priori* most likely change magnitude that should appear in the signal. In case several magnitudes of jump are possible, the best choice is the minimum one. In any case, the resulting change detection algorithm is only optimal to sequentially detect the chosen change magnitude. Notice that an *a posteriori* choice of the most likely change magnitude leads to the GLR algorithm [Lorden 1971].

the detection threshold The classical way to set this parameter is to use the average run length function (1.15), and more particularly the mean time between false alarms ARL_0 and the mean detection delay ARL_1. These two specific values of the ARL function depend on the detection threshold h, and can thus be used to set the performance of the CUSUM algorithm to a desired value for a particular application. A first possibility is to choose the mean time between false alarms to a desired value N_{FA}:

- fix the desired mean time between false alarms $N_{FA} = \mathrm{ARL}_0(h_o)$,
- compute the corresponding optimal threshold value $h_o = {\mathrm{ARL}_0}^{-1}(N_{FA})$,
- deduce the resulting mean detection delay $\mathrm{ARL}_1(h_o)$.

Another possibility is to first choose the mean detection delay to a value N_D the user desire:

- fix the desired mean detection delay $N_D = \mathrm{ARL}_1(h_o)$,
- compute the corresponding optimal threshold value $h_o = {\mathrm{ARL}_1}^{-1}(N_D)$,
- deduce the resulting mean time between false alarms $\mathrm{ARL}_0(h_o)$.

Obviously, such a method requires the knowledge of the different values of the ARL function, and several approaches has been proposed in the literature to evaluate this function in the CUSUM case [Basseville 1993, chap. 5]:

Integral equation approach: Page demonstrates in [Page 1954a] that this ARL function is the solution of an integral equation of Fredholm's type. Unfortunately, finding an analytical solution to this equation is generally impossible. To overcome this problem, several methods have been proposed to numerically evaluate this solution, see for example [Goel 1971]. This first approach can lead to very accurate results, but is computationally intensive.

Approximation approach: Different analytical approximations of this ARL function relying on the theory of sequential analysis [Wald 1947] have been proposed, see for example [Page 1954b] and [Siegmund 1985]. This second approach is simpler than the previous one, but generally leads to less accurate results.

Markov chain approach: Following this approach, it is not only possible to evaluate the average run length function, but also the statistical properties of the CUSUM algorithm such as the run length distribution. It has been introduced in the discrete case in [Brook 1972], adapted to the continuous case in [Lucas 1982a, Lucas 1982b], and further investigated in [Yashchin 1985a, Yashchin 1985b].

As an example, Fig. 1.3 shows Siegmund's approximation [Siegmund 1985] of $\mathrm{ARL}_0(h)$ and $\mathrm{ARL}_1(h)$ in case of a change in the mean of magnitude $\delta = 1$ in

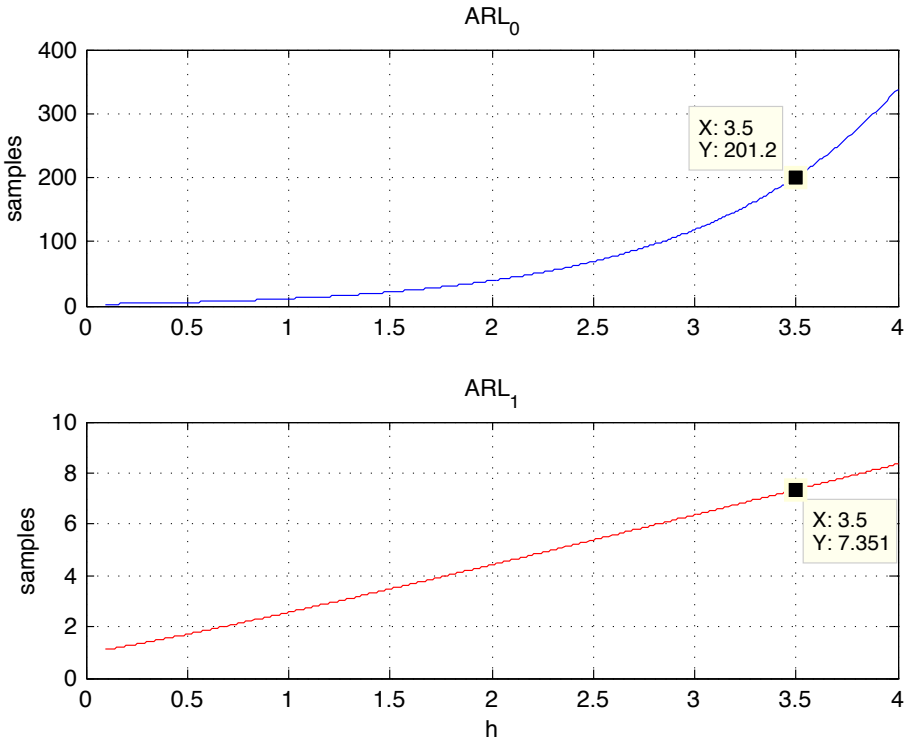


Figure 1.3: Siegmund's approximation of $\mathrm{ARL}_0(h)$ and $\mathrm{ARL}_1(h)$, case of a change in the mean in an iid Gaussian signal with a SNR $= \frac{\delta}{\sigma_{\mathfrak{X}}} = 1$.

an iid Gaussian signal of variance $\sigma^2_{\mathfrak{X}} = 1$. This corresponds to a signal to noise ratio SNR $= \frac{\delta}{\sigma_{\mathfrak{X}}} = 1$. Several things are highlighted by this figure. First, the mean time between two false detections ARL_0 is clearly much more important than the mean detection delay ARL_1 whatever the value of h. For example, the two data-tips show that the CUSUM algorithm applied to such a signal with a detection threshold $h = 3.5$ generates a false detection every 200 samples in mean, and detects a change in the mean of magnitude 1 with a delay of 7 samples in mean. Second, h is directly related to the algorithm reactivity or sensitivity: the smaller h, the more reactive or sensitive the algorithm. Indeed, abrupt changes are detected after very small delays for small h, but the price to pay is to obtain at the same time a small time between false detections, or equivalently a high false alarm rate. On the contrary, the same algorithm generates small false detection rates for large thresholds, but with large detection delays.
The methodology explained in this section finally allows the user to set the CUSUM parameters such that this algorithm reaches desired performance in terms of mean time between false alarms ARL_0 and mean detection delay ARL_1. Relation (1.17) can also be interesting for practical applications. Indeed, it can

be rewritten as:

$$\mathrm{ARL}_1 \sim \frac{\ln \mathrm{ARL}_0}{\mathrm{I}\left(p_{\theta_0}, p_{\theta_1}\right)}, \text{ when } \mathrm{ARL}_0 \to \infty,$$

which roughly connects ARL_1 and ARL_0 when the latter is large.

1.3.3 Two-sided algorithm

Previously developed algorithms are adequate for the detection of changes in one direction only, and are called one-sided CUSUM algorithms. However, in most applications it is necessary to also detect changes in either direction. This is for example the case of a piecewise constant signal buried in noise whose changing mean must be on-line tracked.
A simple solution proposed by Page in [Page 1954a] is to use two one-sided algorithms, one to detect an increase and one to detect a decrease in the parameter θ. This leads to two different instantaneous log-likelihood ratios, $s^i[n]$ dedicated to the increase $\theta_0 \to \theta_1^i$ and $s^d[n]$ to the decrease $\theta_0 \to \theta_1^d$ with $\theta_1^d < \theta_0 < \theta_1^i$. Two cumulative sums $S_{\mathfrak{X}}^i[n]$, $S_{\mathfrak{X}}^d[n]$ and two decision functions $G_{\mathfrak{X}}^i[n]$, $G_{\mathfrak{X}}^d[n]$ can then be computed, and a change in the parameter is finally detected thanks to the following test:

test 3. *Decide $\mathcal{H}_1$ if $\left(G_{\mathfrak{X}}^i[k] > h > 0\right) \cup \left(G_{\mathfrak{X}}^d[k] > h > 0\right)$ (else $\mathcal{H}_0$), where h is a threshold set by the user.*

Once again, the classical way to quantify the performance of this algorithm is the ARL function. As explained in [Basseville 1993, chap. 5], the ARL function of the two-sided CUSUM algorithm can be computed from the ARL functions of the two one-sided CUSUM algorithms it is constituted. Under general conditions, these three ARL functions verify the relation:

$$\frac{1}{\mathrm{ARL}} \leq \frac{1}{\mathrm{ARL}^i} + \frac{1}{\mathrm{ARL}^d}, \tag{1.20}$$

where:

- ARL is related to the two-sided CUSUM algorithm,
- ARL^i is related to the one-sided CUSUM algorithm corresponding to the change $\theta_0 \to \theta_1^i$,
- ARL^d is related to the one-sided CUSUM algorithm corresponding to the change $\theta_0 \to \theta_1^d$.

Finally, inequality (1.20) becomes a simple equality in the case of a change in the mean of an iid Gaussian signal, and the ARL function of the two-sided CUSUM algorithm can be in this case efficiently approximated thanks to methods given in the previous section.

Applied to the case of a change in the mean in an iid Gaussian signal, this general principle leads to the two following instantaneous log-likelihood ratios:

$$s^i[n] = +\frac{|\delta|}{\sigma_{\mathfrak{X}}^2}\left(x[n] - \mu_{\mathfrak{X}_0} - \frac{|\delta|}{2}\right)$$
$$s^d[n] = -\frac{|\delta|}{\sigma_{\mathfrak{X}}^2}\left(x[n] - \mu_{\mathfrak{X}_0} + \frac{|\delta|}{2}\right)$$

where $|\delta|$ is the absolute value of the change magnitude. The resulting algorithm corresponds to the well known cumulative sum control chart widely used in continuous inspection for quality control (see for example [Montgomery 2010]). Moreover, jointly applying the ideas developed in the previous section concerning unknown parameters leads to algorithm 5. This simple and efficient recursive algorithm is able to sequentially detect increasing or decreasing changes in the mean of an iid Gaussian signal thanks to only two parameters:

- the change magnitude $\tilde{\delta}$, set to the most likely magnitude of change by using *a priori* knowledge about the signal,
- the detection threshold h, set by specifying the desired performance through the algorithm ARL function.

Fig. 1.4 shows different results obtained thanks to algorithm 5 applied to a random signal with abrupt changes. The signal of interest is an iid Gaussian signal with a constant standard deviation $\sigma = 1$ and a piecewise constant mean with three abrupt changes. The most difficult change to detect is situated at sample 800 and has a magnitude $\delta_{\min} = 0.5$, which is smaller than the standard deviation. In Fig. 1.4, the original piecewise constant mean is represented in black and its noisy versions are represented in blue. Algorithm 5 is applied to the noisy signal with two different settings. As mentioned previously, the change magnitude parameter $\tilde{\delta}$ must be set to the minimum change magnitude $\delta_{\min} = 0.5$, but the mean time between false detections ARL_0 can be chosen differently:

- First, $\text{ARL}_0 = 1000$. This value is a little bit too small since the last part of the signal has no change in the mean during more than 1000 samples. Therefore, the algorithm should be too much sensitive and false detections are expected in this case.
- $\text{ARL}_0 = 2000$. This value seems to be correct since the maximum number of samples between two changes in the signal is less than 2000. However, the resulting algorithm should be less sensitive than in the previous case, with higher detection delays.

The red curves of Fig. 1.4 represent the piecewise constant signal estimated by algorithm 5 and the black crosses "+" denote the corresponding detection times. Fig. 1.4(a) (resp. 1.4(b)) shows the results obtained with the most (resp. less) sensitive setting. It can first be noticed that whatever the settings, all changes

initialization

set $\tilde{\delta}$ to the most likely change magnitude

set the detection threshold $h > 0$

$S^i[-1] = G^i_{\mathfrak{X}}[-1] = S^d[-1] = G^d_{\mathfrak{X}}[-1] = 0$

initialize the estimators $\widehat{\mu_{\mathfrak{X}0}}$ and $\widehat{\sigma^2_{\mathfrak{X}}}$

$k = 0$

end

while *the algorithm is not stopped* **do**

measure the current sample $x[k]$

calculate the current estimates $\widehat{\mu_{\mathfrak{X}0}}[k]$ and $\widehat{\sigma^2_{\mathfrak{X}}}[k]$

$s^i[k] = \frac{|\tilde{\delta}|}{\widehat{\sigma^2_{\mathfrak{X}}}[k]} \left(x[k] - \widehat{\mu_{\mathfrak{X}0}}[k] - \frac{|\tilde{\delta}|}{2} \right)$

$s^d[k] = -\frac{|\tilde{\delta}|}{\widehat{\sigma^2_{\mathfrak{X}}}[k]} \left(x[k] - \widehat{\mu_{\mathfrak{X}0}}[k] + \frac{|\tilde{\delta}|}{2} \right)$

$S^i[k] = S^i[k-1] + s^i[k]$; $S^d[k] = S^d[k-1] + s^d[k]$

$G^i_{\mathfrak{X}}[k] = \left\{ G^i_{\mathfrak{X}}[k-1] + s^i[k] \right\}$; $G^d_{\mathfrak{X}}[k] = \left\{ G^d_{\mathfrak{X}}[k-1] + s^d[k] \right\}$

if $\left(G^i_{\mathfrak{X}}[k] > h > 0 \right) \cup \left(G^d_{\mathfrak{X}}[k] > h > 0 \right)$ **then**

$n_d \leftarrow k$

if $\left(G^i_{\mathfrak{X}}[k] > h > 0 \right)$ **then**

$\widehat{n_c} = \arg\min_{1 \le n_c \le k} S^i[n_c - 1]$

end

if $\left(G^d_{\mathfrak{X}}[k] > h > 0 \right)$ **then**

$\widehat{n_c} = \arg\min_{1 \le n_c \le k} S^d[n_c - 1]$

end

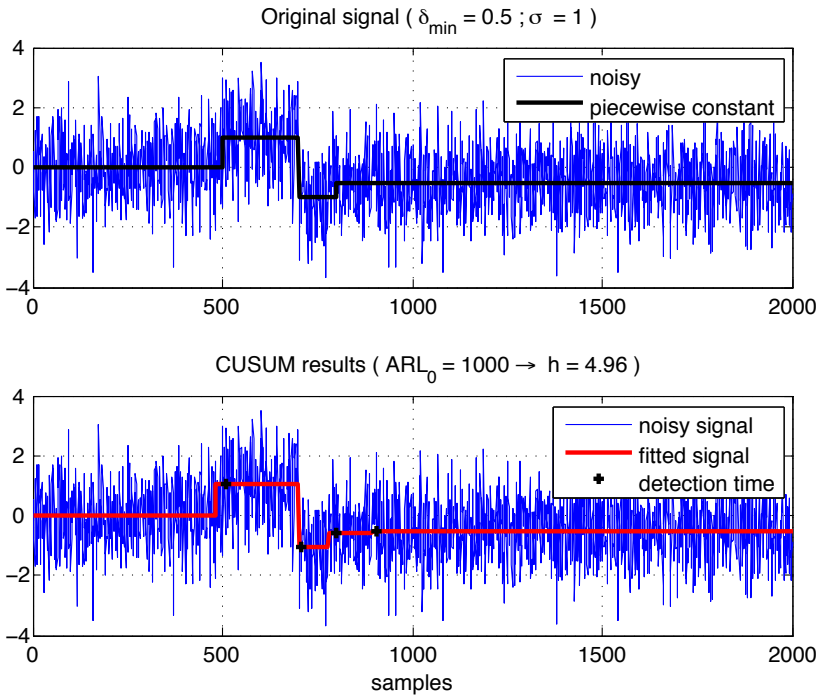
stop or reset the algorithm

end

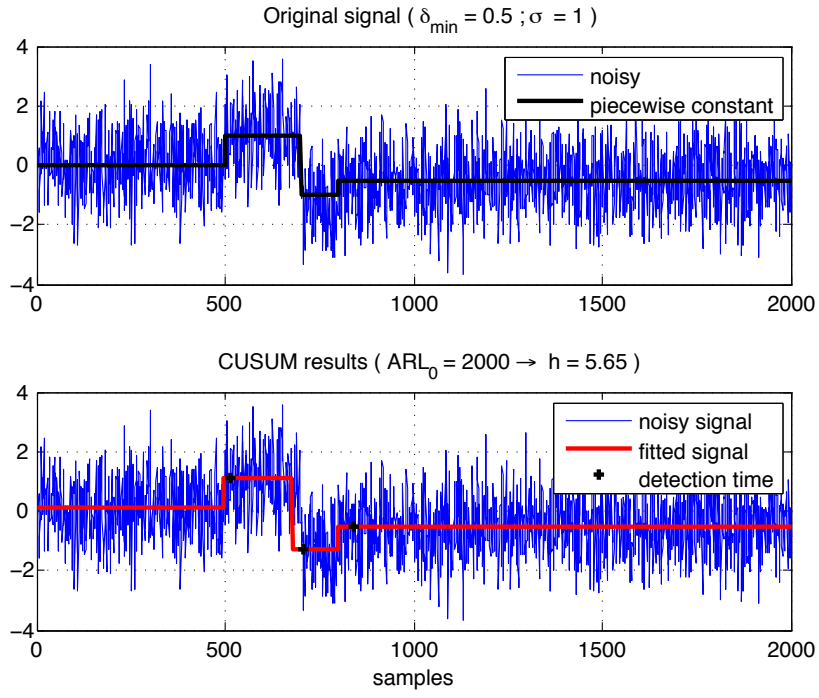
$k = k + 1$

end

Algorithm 5: suboptimal two-sided CUSUM algorithm, jump in the mean - iid Gaussian case.



(a) Setting: $\tilde{\delta} = 0.5$; $\mathrm{ARL}_0 = 1000$.



(b) Setting: $\tilde{\delta} = 0.5$; $\mathrm{ARL}_0 = 2000$.

Figure 1.4: Application of algorithm 5 to an iid Gaussian signal containing three changes in the mean with two different settings.

are correctly detected and estimated. As expected, one false detection appears around sample number 900 in Fig. 1.4(a) with the most sensitive setting. This false detection disappears with the second setting of Fig. 1.4(b), but the detection delays are apparently more important in this case.
Of course, these indicative results should be statistically validated, but this first example correctly illustrate the good properties of algorithm 5.

1.4 Variations

Variations of the classical CUSUM algorithm has also been proposed. Here are the most simple ones.

1.4.1 Fast initial response CUSUM

The classical recursive CUSUM algorithm is usually reset to zero after the detection of a change in the signal. The goal is here to increase its detection performance and to give it a head start by changing the initial value of the cumulative sums when resetting. See [Lucas 1982b] and [Yashchin 1985a].

1.4.2 Combined Shewhart-CUSUM

The classical CUSUM algorithm is optimal to detect small persistent changes in signals, but Shewhart control charts are often faster to detect very important changes. The goal is here to combine the two algorithms in order to improve the resulting detection performance. See [Westgard 1977], [Lucas 1982a] and [Yashchin 1985b].

1.4.3 Multivariate CUSUM

The CUSUM algorithm can be easily generalized to multivariate signals. See for example [Basseville 1993, chap. 7], where the basic case of r-dimensional random vectors of Gaussian independent sequences is treated in section 7.2.1.

Bibliography

[Basseville 1993] M. Basseville et I. Nikiforov. Detection of abrupt changes: Theory and application (prentice hall information and system sciences series). Prentice Hall, 1993. *5 citations on pages 2, 5, 13, 15, and 19*

[Brook 1972] D. Brook et D. A. Evans. *An Approach to the Probability Distribution of the Cusum Run Length*. Biometrika, vol. 59, no. 3, pages 539–549, December 1972. *Cited on page 13*

[Goel 1971] A. L. Goel et S. M. Wu. *Determination of A.R.L. and a Contour Nomogram for Cusum Charts to Control Normal Mean*. Technometrics, vol. 13, no. 2, pages 221–230, May 1971. *Cited on page 13*

[Kay 1993] S. Kay. Fundamentals of statistical signal processing, volume 1: Estimation theory. Prentice Hall, 1 édition, 1993. *Cited on page 4*

[Kay 1998] S. Kay. Fundamentals of statistical signal processing, volume 2: Detection theory. Prentice Hall PTR, 1998. *Cited on page 4*

[Lai 1995] T. L. Lai. *Sequential Changepoint Detection in Quality-Control and Dynamical Systems*. Journal of Royal Statistical Society - Series B, vol. 57, no. 4, pages 613–658, 1995. *Cited on page 2*

[Lorden 1971] G. Lorden. *Procedures for Reacting to a Change in Distribution*. The Annals of Mathematical Statistics, vol. 42, no. 6, pages 1897–1908, 1971. *5 citations on pages 2, 8, 9, 10, and 12*

[Lucas 1982a] J. M. Lucas. *Combined Shewhart-CUSUM Quality Control Schemes*. Journal of Quality Technology, vol. 14, no. 2, pages 51–59, April 1982. *2 citations on pages 13 and 19*

[Lucas 1982b] J. M. Lucas et R. B. Crosier. *Fast Initial Response for CUSUM Quality-Control Schemes: Give your CUSUM a Head Start*. Technometrics, vol. 24, no. 3, pages 199–205, August 1982. *2 citations on pages 13 and 18*

[Montgomery 2010] D. C. Montgomery. Statistical control quality: a modern introduction (6th ed.). Wiley, 2010. *Cited on page 16*

[Moustakides 1986] G. V. Moustakides. *Optimal Stopping Times for Detecting Changes in Distributions*. The Annals of Statistics, vol. 14, no. 4, pages 1379–1387, 1986. *Cited on page 9*

[Page 1954a] E. S. Page. *Continuous Inspection Schemes*. Biometrika, vol. 41, no. 1, pages 100–115, Jun. 1954. *5 citations on pages 2, 7, 9, 13, and 15*

[Page 1954b] E. S. Page. *An improvement to Wald's approximation for some properties of sequential tests.* Journal of Royal Statistical Society - Series B, vol. 16, no. 1, pages 136–139, 1954. *Cited on page 13*

[Ritov 1990] Y. Ritov. *Decision Theoretic Optimality of the Cusum Procedure*. The Annals of Statistics, vol. 18, no. 3, pages 1464–1469, 1990. *Cited on page 9*

[Shewhart 1931] W. A. Shewhart. Economic control of quality manufactured product. MacMillan, London, 1931. *Cited on page 2*

[Siegmund 1985] D. Siegmund. Sequential analysis - tests and confidence intervals. New York: Springer-Verlag, 1985. *Cited on page 13*

[Wald 1947] A. Wald. Sequential analysis. New York: John Wiley and Sons, 1947. *Cited on page 13*

[Westgard 1977] J. O. Westgard, T. Groth, T. Aronsson et C. H. De Verdier. *Combined Shewhart-cusum control chart for improved quality control in clinical chemistry*. Clinicla Chemistry, vol. 23, no. 10, pages 1881–1887, October 1977. *Cited on page 19*

[Yashchin 1985a] E. Yashchin. *On a Unified Approach to the Analysis of Two-Sided Cumulative Sum Control Schemes with Headstarts*. Advances in Applied Probability, vol. 17, no. 3, pages 562–593, September 1985. *2 citations on pages 13 and 18*

[Yashchin 1985b] E. Yashchin. *On the analysis and design of CUSUM-Shewhart control schemes.* IBM Journal Research and Development, vol. 29, no. 4, pages 377–391, July 1985. *2 citations on pages 13 and 19*