Adaptive Quickest Change Detection with Unknown Parameters

Chengzhi Li, Huaiyu Dai and Husheng Li

Abstract

In this report quickest detection of an abrupt distribution change with unknown time varying parameters is considered. A novel adaptive approach, Adaptive CUSUM Test, is proposed to tackle this problem, which is shown to outperform the celebrated Parallel CUSUM Test. Performance is evaluated through theoretical analysis and numerical simulations.

I. INTRODUCTION

Quickest detection is a technique to detect distribution changes as quickly as possible based on sequential observations [4]. It admits a wide range of applications such as quality control, medical diagnosis and intrusion detection. Recently it is also applied in the study of Cognitive Radio [3]. There are two standard mathematical formulations for the quickest detection problem: Bayesian and minimax. The Bayesian formulation [5] assumes that the change point has a geometric distribution, while the minimax formulation [7], which is also our focus in this report, has no such an assumption. The goal of the minimax formulation is to minimize the worst case detection delay given some constraint on the false alarm. When both pre-change and post-change distributions are completely specified, one well-known procedure under the minimax formulation is the Cumulative Sum (CUSUM) test proposed by Page in [6]. Lorden showed that Page’s CUSUM Test is optimal for independent observations [7], and Lai extended this study to dependent observations [8]. In many practical situations, however, the post-change distribution involves unknown parameters. The Generalized Likelihood Ratio (GLR) Test [7] is an optimal procedure to tackle such problems, but unbounded memory requirement makes it infeasible in practice.
To improve efficiency in storage and computation, Nikiforov proposed the Parallel CUSUM Test [9], in which multiple Page’s CUSUM Tests are carried out simultaneously on some specifically chosen values of the unknown parameters.

In this report, we propose an adaptive CUSUM algorithm for quickest detection when there are unknown parameters in the post-change distribution and the parameters may be varying during the detection process. Our new approach can narrow down the range of the unknown parameters to desired precision and track their changes adaptively, thus achieves significant performance improvement over the Parallel CUSUM Test.

The remainder of this report is organized as follows. The system model is given in Section II. After discussing existing approaches to the unknown-parameter problem in Section III, we propose the adaptive CUSUM algorithm in Section IV, together with some performance analysis. The simulation results and conclusions are provided in Section V and Section VI, respectively.

II. SYSTEM MODEL

Suppose a sensor is monitoring some property in the environment. Denote by \( x(t) \) \((t = 1, 2, \ldots)\) its (independent) observation at time slot \( t \), whose probability density function belongs to \( \{p_\theta\}_{\theta \in \Theta} \), where \( \theta = (\theta_1, \theta_2, \ldots, \theta_d)^T \) is a \( d \)-dimensional vector. We assume the value of \( \theta \) is changed from \( \lambda = [\lambda_1, \lambda_2, \ldots, \lambda_d]^T \) to \( \varphi = [\varphi_1, \varphi_2, \ldots, \varphi_d]^T \) at some unknown time instant \( v \), i.e., the distribution of \( x(t) \) is changed from \( p_\lambda \) to \( p_\varphi \), where the vector \( \varphi \) is unknown but lies within a given set \( \Phi \), and \( \lambda \) is a known vector, outside of \( \Phi \). Therefore, two hypotheses of interest are:

\[
\begin{align*}
H_0 : \theta &= \lambda \notin \Phi \\
H_1 : \theta &= \varphi \in \Phi.
\end{align*}
\]

(1)

Correspondingly, the log likelihood ratio (LLR) is defined as:

\[
l_\varphi(t) = \log \left( \frac{p_\varphi(x(t))}{p_\lambda(x(t))} \right).
\]

(2)

For performance measurement, we use detection delay and mean time between false alarms [7], which
are defined respectively as\(^1\)

\[ T_1 = \sup_{v \geq 1} \{\text{esssup}_{E_v} \left( (T^* - v + 1)^+ | x(1), \ldots, x(v - 1) \right) \} , \] (3)

\[ T_0 = E_{\infty} [T^*] , \] (4)

where \( T^* \) is the stopping time determined by the detection algorithms. \( E_v \) denotes the expectation, with respective to \( p_\varphi \), under the assumption that the change happens at time slot \( v \). \( v = \infty \) means that the change never happens and \( E_{\infty} \) denoted the expectation with respective to \( p_\lambda \).

### III. EXISTING ALGORITHMS

In this section, we briefly review four existing algorithms of quickest detection: Page’s CUSUM Test, GLR Test, Window-limited GLR Test and Parallel CUSUM Test. The first one applies to the case where complete knowledge of \( p_\varphi \) and \( p_\lambda \) is available, while the last three apply to the case where \( \varphi \) is unknown.

#### A. Page’s CUSUM Test

With full knowledge about the pre-change and post-change distributions, Page’s CUSUM Test provides an optimal scheme minimizing the worst-case detection delay in (3) [7]. Specifically, the stopping time \( T^* \) in Page’s CUSUM Test is given by

\[ T^* = \inf \left\{ t \left| \max_{1 \leq k \leq t} \sum_{r=k}^{t} l_\varphi(r) \geq h \right\} , \]

where \( h \) is a predetermined threshold and the metric \( l_\varphi(r) \) is defined in (2). An alternative expression of \( T^* \) is:

\[ T^* = \inf \left\{ t \left| s(t) \geq h \right\} , \] (5)

with \( s(t) = \max(s(t - 1) + l_\varphi(t), 0) \) (and \( s(0) = 0 \)). It is equivalent to the above one but more efficient in computation and memory.

\(^1\text{esssup} \) refers to the worst case of all the pre-change distributions and \( [x]^+ = \max(x, 0) \).
B. GLR Test and Window-Limited GLR Test

Unknown parameters in the post-change distribution make the detection more challenging. In such a scenario the optimal procedure is the GLR Test [7], in which the stopping time is given by:

\[ T^* = \inf \left\{ t \mid \max_{1 \leq k \leq t} \sup_{\varphi \in \Phi} \sum_{r=k}^{t} l_{\varphi}(r) \geq h \right\}. \]

The \( \sup \) operation implicates an optimal estimation for the parameters, which results in its asymptotical optimality [7] at the cost of large computation and memory requirement (since no recursive expression is available).

As a cost-effective alternative, Window-Limited GLR (WL-GLR) Test is proposed in [8], [10], where the stopping time is given by:

\[ T_{m,\tilde{m}}^* = \inf \left\{ t > m \mid \max_{m \leq t - m \leq \tilde{m}} \sup_{\varphi \in \Phi} \sum_{r=k}^{t} l_{\varphi}(r) \geq h \right\}, \]

where \( 0 \leq \tilde{m} \leq m \) and \( m \) is the size of a moving window. Since the number of samples are limited by a moving window, the computation cost of WL-GLR is considerably reduced. However, the window size still needs to be large enough (especially at low SNRs) to detect a change reliably due to uncertainty about the parameter. Therefore its computation cost may still be unacceptable in practice.

C. Parallel CUSUM Test

To further reduce the computation burden, Nikiforov [9] proposed the Parallel CUSUM Test. Instead of online estimation of the unknown parameter, the Parallel CUSUM Test carries out a collection of Page’s CUSUM Tests over \( L \) specially chosen values of \( \varphi \), denoted by \( c_1, c_2, ..., c_L \). The stopping time is given by:

\[ T^* = \inf \{ T_i, i = 1, 2, ..., L \}, \]

where \( T_i = \inf \left\{ t \mid s_{c_i}(t) \geq h \right\} \) and metric \( s_{c_i}(t) \) is computed recursively by \( s_{c_i}(t) = \max \left( s_{c_i}(t-1) + l_{c_i}(t), 0 \right) \).

The Parallel CUSUM Test reduces computation complexity but is suboptimal in nature. Its suboptimality lies in the fact that the candidates won’t be changed during the detection process so that the inaccuracy of the parameters affects the performance throughout the process. \( L \) can be fairly large to
achieve satisfactory detection delay $\bar{T}_1$, which, however, increases the computation burden and decreases the mean time between false alarms $\bar{T}_0$.

IV. ADAPTIVE CUSUM ALGORITHM

The exploration in the unknown-parameters problem is under development. Especially the study is still lacking on the practical scenarios where the unknown parameters vary over time, e.g., due to channel fading in the wireless environment. In such situations, all algorithms originally designed with the assumption of fixed parameters (such as those discussed in Section III) will degrade.

A straightforward idea to solve the unknown parameters problem is separate operation, estimating the parameters first and then performing the detection. This idea is simple, however it hurts the performance of both estimation and CUSUM test. The estimation should not be stopped since more samples could lead to more accurate estimation. And the CUSUM test should start from the beginning no matter the estimates are accurate or not. Therefore joint operation, detecting the change while estimating the parameters, is more preferred in terms of performance.

In this section, we propose a new quickest detection algorithm, the Adaptive CUSUM Test, which achieves a better tradeoff between performance and complexity and performs stably in changing environment. Our algorithm is recursive in nature, with each recursion comprising two main interleaved steps: parameter tracking and modified CUSUM test.

A. Parameter Tracking

Our adaptive parameter tracking approach narrows down each unknown parameter’s range to desired precision for a generic parameterized distribution satisfying the concavity property of $F(\hat{\varphi})$, where $\hat{\varphi}$ is an estimate of $\theta$ when $\theta = \varphi$ and $F(\hat{\varphi}) = E[l_{\hat{\varphi}}(t)]^2$, the mismatched Kullback-Leibler (KL) divergence between $p_{\varphi}$ and $p_{\lambda}$. For rigorous presentation, we restrict our attention to the exponential family, i.e.,

\[^{2}\text{Here the expectation is with respect to } p_{\varphi}.\]
\( p_{\theta \in \Theta} \) is defined as:
\[
p_{\theta}(x) = h(x) \exp \left( \sum_{i=1}^{s} \theta_i T_i(x) - A(\theta) \right),
\]
where \( T(x) = \{T_1(x), T_2(x), ..., T_s(x)\} \) is a sufficient statistic, \( A(\theta) \) is a normalization factor and \( s \geq d \).

We obtain the following result:

**Proposition 1:** For distributions in the exponential family, \( F(\hat{\varphi}) \) is a concave function with global maximum achieved at \( \hat{\varphi} = \varphi \).

**proof:**

\[
F(\hat{\varphi}) = E \left[ \log \frac{p_{\hat{\varphi}}(x(t))}{p_{\lambda}(x(t))} \right] = E \left[ \log \frac{p_{\varphi}(x(t))}{p_{\lambda}(x(t))} \right] - E \left[ \log \frac{p_{\varphi}(x(t))}{p_{\hat{\varphi}}(x(t))} \right] = E[l_{\varphi}(t)] - D(p_{\varphi} \parallel p_{\hat{\varphi}}),
\]

where \( D(p_{\varphi} \parallel p_{\hat{\varphi}}) \) is the KL divergence between \( p_{\varphi} \) and \( p_{\hat{\varphi}} \). Since \( D(p_{\varphi} \parallel p_{\hat{\varphi}}) \) is nonnegative [11], \( F(\hat{\varphi}) \) achieves the global maximum when \( \hat{\varphi} = \varphi \).

Substituting (7) in (8) and taking partial derivation with respect to \( \hat{\varphi}_i \), we have
\[
\frac{dF(\hat{\varphi})}{d\hat{\varphi}_i} = - \frac{dD(p_{\varphi} \parallel p_{\hat{\varphi}})}{d\hat{\varphi}_i} = -E \left[ \frac{d}{d\hat{\varphi}_i} \sum_i ((\varphi_i - \hat{\varphi}_i)T_i(x(t)) - (A(\varphi_i) - A(\hat{\varphi}_i))) \right] = E[T_i(x(t))] - \frac{dA(\hat{\varphi})}{d\hat{\varphi}_i}.
\]

Therefore when \( \hat{\varphi}_i = \varphi_i \), \( \frac{dF(\hat{\varphi})}{d\hat{\varphi}_i} = 0 \) due to the fact that \( E[T_i(x(t))] = \frac{dA(\varphi)}{d\varphi_i} \) [12].

For the second derivative \( \frac{d^2F(\hat{\varphi})}{d\hat{\varphi}_i d\hat{\varphi}_j} = - \frac{d^2A(\hat{\varphi})}{d\hat{\varphi}_i d\hat{\varphi}_j} \). According to the differential identities of \( A(\varphi) \) [12], \( \frac{d^2A(\hat{\varphi})}{d\hat{\varphi}_i d\hat{\varphi}_j} = cov(T_i(x(t)), T_j(x(t))) \). Therefore the second derivative matrix \( M \) is given by:
\[
M = -E \left[ (T_1(x(t)), T_2(x(t)), ..., T_s(x(t)))^T(T_1(x(t)), T_2(x(t)), ..., T_s(x(t))) \right],
\]
which is a non-positive definite matrix since the quadratic function \( aM a^T \leq 0 \), where \( a \) is any row vector with size \( s \). Thus, \( F(\hat{\varphi}) \) is a concave function.
Corollary 1: For a particular parameter $\theta_i$ and any estimates of other parameters, $F(\hat{\phi}_i)$ is also a concave function with global maximum at $\hat{\phi}_i = \phi_i$.

According to Corol. 1, for the parameter $\theta_i$ we can draw the following conclusions after distribution change:

- given a small value $\delta_i$ and estimations for other parameters, we can always find two estimates $\hat{\phi}_{ia}$, $\hat{\phi}_{ib}$ for $\theta_i$ such that $\hat{\phi}_{ib} = \hat{\phi}_{ia} + \delta_i$ and $F(\hat{\phi}_{ia}) = F(\hat{\phi}_{ib})$;
- $\theta_i$’s true value $\phi_i$ lies within the interval $(\hat{\phi}_{ia}, \hat{\phi}_{ib})$, denoted by $\hat{\Phi}_i$.

Intuitively, we can narrow down the range of $\phi_i$ from its original range $\Phi_i$ to $\hat{\Phi}_i$; this also allows parameter tracking in time-varying environments. An iterative procedure can be used to find $\hat{\phi}_{ia}$ and $\hat{\phi}_{ib}$. Given $\delta_i$, one begins by arbitrarily choosing $\hat{\phi}_{ia}$ and $\hat{\phi}_{ib}$ within $\Phi_i$, say $\hat{\phi}_0^{ia}$ and $\hat{\phi}_0^{ib}$. And then the succeeding values of $\hat{\phi}_{ia}$ and $\hat{\phi}_{ib}$ are obtained according to the recursion:

\[
\hat{\phi}_{ia}^{k+1} = \hat{\phi}_{ia}^k + \xi D_i^k, \quad (9)
\]
\[
\hat{\phi}_{ib}^{k+1} = \hat{\phi}_{ia}^k + \delta_i, \quad (10)
\]

where $\hat{\phi}_i^k$ and $\hat{\phi}_i^k$ represent the values of $\hat{\phi}_{ia}$ and $\hat{\phi}_{ib}$ at the $k$th iteration, $D_i^k$ is the difference between $F(\hat{\phi}_{ib})$ and $F(\hat{\phi}_{ia})$ at the $k$th iteration, given by:

\[
D_i^k = F(\hat{\phi}_{ib}^k) - F(\hat{\phi}_{ia}^k) = E \left[ \log \frac{p^{\hat{\phi}_{ib}^k}(x(k))}{p^{\hat{\phi}_{ia}^k}(x(k))} \right],
\]

and $\xi$ is a step size controlling the rate of adjustment.

In practice, it is not easy to calculate $D_i^k$ due to the expectation operation. Two possible variations can be considered to overcome the difficulty.

- Continuous Updating

\[
\hat{\phi}_{ia}^{k+1} = \hat{\phi}_{ia}^k + \xi \hat{D}_i^k, \quad (11)
\]

where $\hat{D}_i^k = l_{\hat{\phi}_{ib}^k}(k) - l_{\hat{\phi}_{ia}^k}(k) = \log \frac{p^{\hat{\phi}_{ib}^k}(x(k))}{p^{\hat{\phi}_{ia}^k}(x(k))}$. It is a common approach to replace the ensemble average by the time average, accomplished by means of the recursive first-order difference equation in (11).
• Block Updating

Another variation is to average the log likelihood ratio of $p_{\hat{\varphi}_{ib}}$ and $p_{\hat{\varphi}_{ia}}$ over several iterations prior to making adjustment.

\[ \hat{\varphi}_{ia}^{(k+1)N} = \hat{\varphi}_{ia}^{kN} + \xi \hat{L}_{kN}, \]

(12)

where

\[ \hat{L}_{kN} = \frac{1}{N} \sum_{n=0}^{N-1} \log \frac{p_{\hat{\varphi}_{ib}}(x(kN + n))}{p_{\hat{\varphi}_{ia}}(x(kN + n))}. \]

Convergence analysis is as follows. If $D_{i}^{k} > 0$, $\varphi_{ia}^{k+1}$ and $\varphi_{ib}^{k+1}$ will grow according to (9) and (10), so that $D_{i}^{k+1}$ will decrease due to the concavity of the function. Similarly, if $D_{i}^{k} < 0$, $\varphi_{ia}^{k+1}$ and $\varphi_{ib}^{k+1}$ will become smaller so that $D_{i}^{k+1}$ will increase. In both cases, $D_{i}^{k}$ converges to zero surely. $\delta_{i}$ is a key factor in our algorithm. On the one hand larger $\delta_{i}$ leads to faster convergence. On the other hand, $\delta_{i}$ is the range for the unknown parameter, which is desired to be small. Therefore, choosing $\delta_{i}$ involves certain tradeoffs which depends on applications in practice.

B. change detection

In the process of tracking the unknown parameters, change detection is conducted through an appropriate CUSUM test simultaneously. Basically, after the parameter tracking in time $t$, a value is specified in each parameter’s new range and substituted into the CUSUM test, i.e., equ (5) with $s(t) = \max (s(t-1) + l_\hat{\varphi}(t), 0)$, where $\hat{\varphi}$ is the parameter vector composed of the specified values of all the unknown parameters. As we discussed at the end of last section, we expect the range size $\delta_{i}$ as small as possible and the algorithm converges as fast as possible. However, they can not be achieved simultaneously. In the following, we’ll discuss how to exploit the non-optimality coefficient to calculate $\delta_{i}$ to give a best tradeoff between convergence and performance. We first propose an approach for one unknown parameter case and then discuss the problems extending the approach to multiple unknown parameters.
Define a non-optimality coefficient \( \varepsilon_\varphi \) and the maximum non-optimality coefficient \( \varepsilon_m \) as:

\[
\varepsilon_\varphi = 1 - \frac{T_{\text{opt}}}{\bar{T}_\varphi}, \quad (13)
\]

\[
\varepsilon_m = \sup_{\varphi \in [\varphi_{\text{min}}, \varphi_{\text{max}}]} \varepsilon_\varphi, \quad (14)
\]

where \( T_{\text{opt}} \) is the optimal detection delay given by the Page’s CUSUM Test when \( \varphi \) is known and \( \bar{T}_\varphi \) is the detection delay achieved by the Parallel CUSUM Test or our Adaptive CUSUM Test when the true value \( \varphi \) is unknown.

When there is one unknown parameter, given \( \varepsilon_m \), we can predetermine \( L \) candidates \( a_j \), according to the Parallel CUSUM Test [9], and confidence intervals \([\bar{a}_j, a_j] \) associated with these candidates. \(^3\) \( \delta \)\(^4\) can be determined as the minimal confidence interval, which is associated with the \( l \)th candidate. Specifically,

\[
\delta = \min_{1 \leq j \leq L} (\bar{a}_j - a_j); \quad l = \arg \min_{1 \leq j \leq L} (\bar{a}_j - a_j)
\]

After (9) and (10), the value used in the CUSUM test is updated as:

\[
\bar{\varphi}^k = \varphi_a^k + \alpha \delta,
\]

where \( \alpha = 1/2 \) when \( F(\hat{\varphi}) \) is symmetric with respect to \( \varphi \), or \( \alpha = \frac{a_i - a_a}{\delta} \) when \( F(\hat{\varphi}) \) is asymmetric. \( \bar{\varphi}^k \) converges to some value \( \bar{\varphi} \) when the parameter tracking procedure converges. If \( \bar{\varphi}^k \) is out of the range \([\varphi_{\text{min}}, \varphi_{\text{max}}] \), it is set to \( \varphi_{\text{min}} \) or \( \varphi_{\text{max}} \), whichever is closer. Our algorithm continues by substituting \( \bar{\varphi}^k \) as the true value of \( \varphi \) in the CUSUM test.

Under \( H_1 \), \( \bar{\varphi} = \varphi \) for the symmetric \( F(\hat{\varphi}) \) since when the algorithm converges, \( \varphi \) lies right in the middle of \( \varphi_a \) and \( \varphi_b \). In such a case this procedure provides an adaptive estimation for the unknown parameter. For the asymmetric \( F(\hat{\varphi}) \), \( \bar{\varphi} \neq \varphi \) but \( \varepsilon_{\bar{\varphi}} \leq \varepsilon_m \) since the true value is within \( \delta \) after convergence and \( \delta \) is the confidence interval of \( \bar{\varphi} \), which is equal to the minimal interval in the Parallel CUSUM Test. And \( \varepsilon_{\bar{\varphi}} \) is actually much less than \( \varepsilon_m \) shown by the simulation result in the next section.

\(^3\)These candidates guarantee that the non-optimality coefficient \( \varepsilon_\varphi \) is less than \( \varepsilon_m \) for any true value \( \varphi \). If the true value is within the confidence interval associated with a candidate, say \( a_1, \varepsilon_{a_1} \leq \varepsilon_m \).

\(^4\)we omit the subscript \( i \) here since there is only one unknown parameter.
Extending the above approach to multiple parameters is theoretically feasible, however in practice it could be quite difficult to do so. First finding multiple candidates, each of which is a vector, and their associated “confidence space” is quite challenging since the equ (13) and (14) do not provide enough information for the candidates. Secondly, based on these confidence space, specifying $\delta_i$ for parameter $i$ is nontrivial. Let’s take two unknown parameters for example, in which “confidence space” becomes “confidence area”. To find out $\delta_i$ we have to find out the largest square within each confidence area, which is time consuming. Therefore how to choose $\delta_i$ for multiple parameters needs further study.

The Adaptive CUSUM Test is summarized as follows:

1) Initialization: for parameter $i$, set an initial value $\varphi_{0i}$ and a new range size $\delta_i$.

2) In time $k$,
   - update the estimation of parameter $i$ $\varphi_{ki}$ according to the parameter tracking approach;
   - specify a proper value within $\hat{\Phi}_i$;
   - substitute the specified value of each parameter into the CUSUM test.

3) Stop: The algorithm continues until the accumulated statistic $s(k)$ in the CUSUM test is larger than the threshold $h$.

This approach is suboptimal in nature. However the simulations in the next section show that its performance is close to the optimal one by choosing a fairly small range size for each parameter.

C. performance analysis

The detection delay can be approximated by [7]

$$T_1 \approx \frac{h}{E_1[I_{\varphi}(t)]} \quad as \quad h \to \infty,$$

Under $H_0$, the mean time between false alarms admits [7]:

$$\bar{T}_0 \geq e^h,$$

which means there is no performance loss in term of $\bar{T}_0$, unlike the Parallel CUSUM Test in which $\bar{T}_0 \geq \frac{1}{L}e^h$. 
V. NUMERICAL RESULT

A. single parameter

1) Symmetric $F(\hat{A})$: In this section we take detecting a sinusoid wave with an unknown amplitude for an example to demonstrate the performance improvement of our Adaptive CUSUM Test over the Parallel CUSUM Test. The following hypotheses are assumed:

$$\begin{align*}
H_0 &: x(t) = n(t) \\
H_1 &: x(t) = A \sin(wT_s) + n(t),
\end{align*}$$

where $A$ is an unknown amplitude within the range $[2, 36]$; $w$ is the carrier frequency and $T_s$ is the sampling period, both of which are known. $n(t)$ is Gaussian noise with zero mean and unit variance.

Denote by $\hat{A}$ an estimate of amplitude $A$. It is easy to check $F(\hat{A}) = E[l_A(t)]$ is symmetric about $A$. We choose three candidates for the Parallel CUSUM Test so that its computation complexity is comparable with the Adaptive CUSUM Test. We set $\varepsilon_m$ to 0.25 and the corresponding three candidates of the Parallel CUSUM Test are 3, 6, 18. The thresholds for both tests are the same and set to 5000.

Fig. 1 compares non-optimality coefficients of our Adaptive CUSUM Test and the Parallel CUSUM Test for different amplitudes, staying fixed during the detection process. We can see non-optimality coefficients of the Parallel CUSUM Test are substantially larger than those of our Adaptive CUSUM Test at almost all possible amplitudes (except for the three candidates chosen by the Parallel CUSUM

Fig. 1: Performance comparison: fixed amplitude.
Fig. 2: Performance comparison: time-varying amplitude

Test). The average non-optimality coefficient of the Parallel CUSUM Test is 0.11, while for our Adaptive CUSUM Test, it is 0.01, indicating little degradation in optimality. In Fig. 2, it is assumed that the sinusoid wave goes through a block fading channel, where the amplitude changes every 600 time slots randomly within the range rather than fixed. We compare the average non-optimality coefficient of these two tests under different average number of variations, and for each point of Fig. 2, 250 independent experiments are conducted to guarantee its fidelity. We can see that the performance of our Adaptive CUSUM Test is stable and close to the optimal detection since it can track the amplitude change, while the Parallel CUSUM Test performs worse as more variations are involved.

2) Asymmetric $F(\hat{\phi})$: To demonstrate the performance of our algorithm in an asymmetric case, we detect the change of the mean in a poisson distribution. The hypothesis is given by (1), where $\lambda = 1$ and $\Phi = [2, 40]$. The non-optimality coefficient $\epsilon_m = 0.25$ for the Parallel CUSUM Test, which results in three candidates 2.6, 8.1 and 56.9.

Fig. 3 compares non-optimality coefficients of these two tests. Surprisingly, the Adaptive CUSUM performs much better than the Parallel CUSUM Test even the true value of the parameter has not been achieved.
Fig. 3: Performance comparison for the asymmetric case.

B. Multiple parameters

In this simulation, we assume a Gaussian variable changes its distribution from $\mathcal{N}(0, 1)$ to $\mathcal{N}(m, \sigma^2)$, where both $m$ and $\sigma$ are unknown to the detector and $m \in [3, 23], \sigma \in [2, 4]$. The simulation is composed of 50 rounds and in each round $m$ and $\sigma$ are randomly chosen from their ranges. Set $\delta_m = 6$ and $\delta_\sigma = 1$ in our adaptive CUSUM test and specify the values of $m$ and $\sigma$ in the middle of their ranges. As we discussed in the last section it’s difficult to choose candidates for the Parallel CUSUM Test in the two parameters case. We choose them by experiments. For simplicity, we apply the true variance in the parallel CUSUM Test and choose the candidates 3, 6, 12 for the mean to guarantee that its non-optimality coefficient is less than 0.1. Figure 4 shows the non-optimality coefficient of both algorithms. We can see from the figure that the non-optimality coefficient of the Adaptive CUSUM Test is below 0.05, far outperforming the Parallel CUSUM Test. In addition compared to the single parameter cases in Fig.1 and 3, multiple unknown parameters do not degrade the performance of our approach notably.

VI. CONCLUSIONS

In this report, we have studied quickest change detection with unknown parameters. An adaptive CUSUM algorithm is proposed to narrow down the ranges of the unknown parameters and track their possible change during the detection process. Analysis and numerical results show that the new algorithm achieves
better performance than the Parallel CUSUM Test. Interesting future directions include collaborative quickest detection in an ad-hoc network and the application in the spectrum sensing problem in Cognitive Radio.

REFERENCES


