# Modelling Recurrent Events for Improving Online Change Detection

Alexandr Maslov*‡ Mykola Pechenizkiy* Indrė Žliobaitė† Tommi Kärkkäinen‡

**Abstract**

The task of online change point detection in sensor data streams is often complicated due to presence of noise that can be mistaken for real changes and therefore affecting performance of change detectors. Most of the existing change detection methods assume that changes are independent from each other and occur at random in time. In this paper we study how performance of detectors can be improved in case of recurrent changes. We analytically demonstrate under which conditions and for how long recurrence information is useful for improving the detection accuracy. We propose a simple computationally efficient message passing procedure for calculating a predictive probability distribution of change occurrence in the future. We demonstrate two straightforward ways to apply the proposed procedure to existing change detection algorithms. Our experimental analysis illustrates the effectiveness of these approaches in improving the performance of a baseline online change detector by incorporating recurrence information.

## 1 Introduction

Increasing volumes of data are being generated by sensors when monitoring industrial processes, traffic, or infrastructure. Such streaming data typically needs to be analyzed continuously as it arrives, using only information observed so far. Predictive models, once learned on historical data, may become obsolete due to changes in physical process generating data, as well as external changes in the environment. In machine learning and data mining changes in the underlying data distribution over time are referred as concept drift [1].

A popular approach for handling concept drift is to monitor data or model performance for changes and to adapt model using most recent data collected after the last detected change [2]. Even without automated model adaptation, online change detection is practically relevant in many domains, such as medicine, energy production, industrial processes monitoring [3], for signalling human experts that something has changed in the process.

Change points are defined as moments in time, when the statistical properties of a data stream change significantly according to predefined criteria. The process of online change detection is often challenging due to presence of noise that, when operating online, may be mistaken for real changes. A good change detector is expected to detect changes within an acceptable time lag, and be robust to noise by not raising false alarms. A balance between these two requirements is determined by the parameter settings for the detector. Typically, the optimal settings are learned during an offline training phase, and then applied on newly arriving data. Our idea is to adjust the sensitivity of the detector online during operation, in relation to the current probability of change reoccurrence.

While many change detection methods have been developed [3, 4] for offline and online settings, they typically assume that changes occur at random in time, and are independent from each other. In practice, however, in many industrial applications changes occur with some regularity (e.g. seasonality). Our approach is to capture this information from data, and utilize it for improving the accuracy of online change detection.

Consider the following example. A continuously operating power production plant needs to be regularly refueled. Change detection software is deployed for detecting refueling events from sensor data, and providing this information to the operation and control system. If, for instance, historical data suggests that refueling usually happens between 7pm and 10pm every day, this information can be learned from data, and the sensitivity of the detector can be increased within this time range expecting to improve detection speed, and lowered otherwise expecting to reduce the rate of false alarms.

In this study we analyze how and under which conditions the performance of online change detectors can be improved for detecting recurrent changes:

- We define a predictive change confidence function (PCCF) for modelling recurrent changes and predicting times of changes in the future. We derive the exact analytical expression for PCCF under Gaussian data distribution.

*TU Eindhoven, Department of Mathematics and Computer Science, The Netherlands

†Aalto University, Department of Computer Science; Helsinki Institute for Information Technology HIIT; University of Helsinki, Department of Geosciences and Geography, Finland

‡University of Jyväskylä, Department of Mathematical Information Technology, Finland

- We demonstrate under which conditions taking into account recurrence information improves detection accuracy by analytically relating PCCF and time passed from the most recent confirmed change.
- We demonstrate how to improve the accuracy of an online change detector of user's choice by utilizing recurrence information with two simple yet generic approaches: (1) a post-filtering of change detection output can substantially reduce the number of false alarms while preserving the same detection rate; (2) an adaptation mechanism can be used to adjust the sensitivity of the detector online based on the change recurrence expectations.
- Our experimental study provides evidence that it is indeed feasible to improve performance of online change detection by utilizing recurrence information even with such simple approaches.

Several lines of work relate to our approach via attention to recurrent concept drift [5–7], predictability of concept drift [8], or change detection with delayed labeling [9]. These approaches are specific to handling concept drift, while our focus is on generic online change detection and its accuracy. In the Bayesian online change detector proposed in [10] and extended in [11] authors model time intervals between change points (run lengths) using the hazard rate. This approach allows to take into account recurrence by tuning single parameter, but it does not allow to distinguish outliers from changes which may appear between them. In [12] data stream volatility, defined as the rate of detected changes, is used to make detector more reactive. We concentrate on the problem of improving change detection by predicting time locations of the changes in the future in order to better distinguish outliers from real changes. To the best of our knowledge this is the first work that explicitly models change recurrence as PCCF and uses it to enhance online change detection performance.

*Outline.* The rest of the paper is organized as follows. In Section 2 we describe an individual change detection procedure and give definitions for recurrent and periodic changes. In Section 3 we describe our approach for computing PCCF. In Section 4 we describe integration of PCCF with a base-level change detector. In Section 5 we summarize the results of our experimental study. Section 6 concludes the paper with the summary of the contributions and future work.

## 2 Recurrent and periodic changes

The purpose of the change detection process is to detect changes in the input data stream of observations $x_t, t \in \mathcal{T}$ to be obtained sequentially at time moments $\mathcal{T} = \langle t_1, t_2, \ldots, t_T \rangle \equiv \langle t_q \rangle_{q=1}^{T}$ with a constant sampling rate. Change is identified by the moment of time when it happened. Let us denote the sequence of changes to be detected as $\langle c_i \rangle_{i=1}^{k} \in \mathcal{T}$ and an individual change from this sequence as $c_i$. After a change occurs, we need to collect a new portion of observations during the time $\delta$ to detect the change and to assess a new probability distribution of the data.

Change detector can be considered as a binary classifier whose output for each $x_t$ is the label $(+)$ if change is alarmed, which we call a change detection event (CDE) and denote it $\mathbf{e}_t^+$, and label $(-)$ otherwise, which we denote as $\mathbf{e}_t^-$.

To assess detectors' performance we define True Positive (TP), False Positives (FP), True Negatives (TN) and False Negatives (FN) as follows:

- $\mathbf{e}_t^+$ is TP if $\exists c_i : t - c_i < \delta$, and FP if $\nexists c_i : t - c_i < \delta$
- $\mathbf{e}_t^-$ is FN if $\exists c_i : t - c_i < \delta$, and TN if $\nexists c_i : t - c_i < \delta$

Recall the real world example of a power production plant, where the task is to detect changes in the fuel mass flow signal generated by monitoring sensors. One subtask is to detect online changes in the signal that correspond to the refueling process [13]. Consider two scenarios. Scenario **A**: historical data indicates that refueling typically happens every day in between 7pm and 10pm (90% of all the observed cases). Hence, if a detector alarms a change at 11am we may conclude that most likely it is an outlier and take actions to prevent FP. Scenario **B**: we have information that refueling is performed only when the fuel is about to run out, but we have no information that it is performed on a daily basis. We can observe from historical data that the average time between refueling events is between 32 and 36 hours (90% of all observed cases).

The two scenarios differ in possibilities to anticipate the next change point. In scenario **A** we assume that changes are associated with time moments, which can be learned and thus known in advance (e.g. end of the day), while in scenario **B** we can relate future changes to the recently detected event(s). Nevertheless, in both cases our intuition is to estimate the average time between changes from historical data, and use this information for determining the time periods where changes are more likely to happen in the future.

Let us denote by $p(c|\theta)$ a probability mass function (Pmf) defined on a discrete set of time moments $\mathcal{T}$ parameterized by vector $\theta$ holding information about expected distance between consecutive changes (e.g. the average and standard deviation). By $p(c_i = t|\theta)$ we denote Pmf for a change $c_i$ to happen at time $t$. Recurrent and periodic changes are defined as follows.

DEFINITION 1. *Changes* $\langle c_i \rangle_{i=1}^{k}$ *are periodic if*

$$p(c_{i+1} = t|\theta) = p(c_1 = t - iw \mid \theta), i \in \mathbb{Z}, \tag{2.1}$$

*where* $c_1$ *is the time of the first change and* $w$ *is a time range within which we expect the next change.*

Definition 1 corresponds to the case **A** and states that Pmf of a periodic change $c_{i+1}$ is a Pmf of the change $c_1$ shifted along the time axis by $iw$. In the power plant example, refueling happens every day in the evening, i.e. $iw$ is the beginning of the $i^{th}$ day and $w = 24$ hours.

DEFINITION 2. *Changes* $\langle c_i \rangle_{i=1}^{k}$ *are recurrent if*

$$p(c_{i+1} = t \mid \theta) = p(c_1 = t - c_i \mid \theta), \tag{2.2}$$

*where* $c_1$ *is the time of the* $1^{st}$ *change and* $c_i$ *is the time of the* $i^{th}$ *change.*

Definition 2 corresponds to the case **B** and states that Pmf of the change $c_{i+1}$ is a Pmf of the change $c_1$ shifted along the time axis by time of the $i^{th}$ change $c_i$.

## 3 Online detection of recurrent changes

DEFINITION 3. *PCCF is the probability to observe any a recurrent change out of sequence of all possible changes* $c \in \langle c_i \rangle_{i=1}^{k}$ *at any given time moment* $t$*:*

$$\mathcal{P}(t \mid \theta) = \sum_{i=1}^{k} p(c_i = t \mid \theta). \tag{3.3}$$

Next we demonstrate how to compute PCCF for recurrent (Scenario **B**) and periodic (Scenario **A**) changes, and update it using time stamps of changes $t_s$ confirmed by an online confirmation mechanism. We show under which conditions recurrence information in a form of the average time between changes and its standard deviation $\theta = (\mu, \sigma)$ is beneficial for improving the detection accuracy.

**3.1 PCCF for periodic changes.** The number of periodic changes $k$, which have happened until now (time $t$), can be calculated using the integer division operation:

$$k = \mathbf{div}(t, w). \tag{3.4}$$

Using Definition 1 and 3 and omitting $\theta$ we can write

$$\mathcal{P}(t) = \sum_{j=1}^{k=\mathbf{div}(t,w)} p(c_1 = t - (j-1)w). \tag{3.5}$$

If each $p(c_j)$ is defined only within the time interval $(jw - w, jw]$ (e.g. $w$ stands for the duration of the day, and we expect every change in a sequence to happen only within the next day) then PCCF will be given only by the last member of the sum in Eq.(3.5). An example of such PCCF is illustrated in Figure 1.

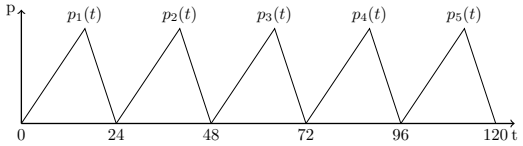


Figure 1: An example of PCCF function for periodic events with $w = 24$h, $iw = (24, 48, 72, \dots)$.

**3.2 PCCF for recurrent changes.** According to Definition 2 Pmf of the change $c_{i+1}$ is conditioned on the time of the $i^{th}$ change $c_i$. Following the sum rule for probability[1] in order to compute Pmf of $c_{i+1}$ we need to consider all possible time locations of $c_i$.

$$p(c_{i+1} = t) = \sum_{\tau=i}^{t-1} p(c_{i+1} = t \mid c_i = \tau) p(c_i = \tau). \tag{3.6}$$

PCCF is a sum over the total number of possible changes by the time moment $t$:

$$\begin{aligned} \mathcal{P}(t) &= \sum_{i=1}^{t} \sum_{\tau=i}^{t-1} p(c_{i+1} = t | c_i = \tau) p(c_i = \tau) \\ &= \sum_{i=1}^{t} \sum_{\tau=i}^{t-1} p(c_1 = t - c_i) p(c_i = \tau). \end{aligned} \tag{3.7}$$

**3.3 Numerical PCCF computation procedure.** Figure 2 shows a toy example of PCCF calculation for every moment of the recurrent change detection process of time length $T = 5$, i.e. $\mathcal{T} = \langle 1, 2, 3, 4, 5 \rangle$. Pmf for the first change $p(c_1) = [p_1, p_2, p_3, p_4, p_5]$ is depicted by the first column of the nodes which are all black since the first change can occur at any moment $t \in [1, 5]$. According to Eq.(3.6) Pmf for the second change is

$$\begin{aligned} p(c_2) &= \sum_{\tau=1}^{4} p(c_2 = t | c_1 = \tau) p(c_1 = \tau) \\ &= \sum_{\tau=1}^{4} p(c_1 = t - \tau) p(c_1 = \tau). \end{aligned} \tag{3.8}$$

This sum can be equivalently written in a matrix form

$$p(c_2) = \begin{bmatrix} p_1 & 0 & 0 & 0 \\ p_2 & p_1 & 0 & 0 \\ p_3 & p_2 & p_1 & 0 \\ p_4 & p_3 & p_2 & p_1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}. \tag{3.9}$$

[1] $P(x) = \sum_y P(x|y)p(y)$

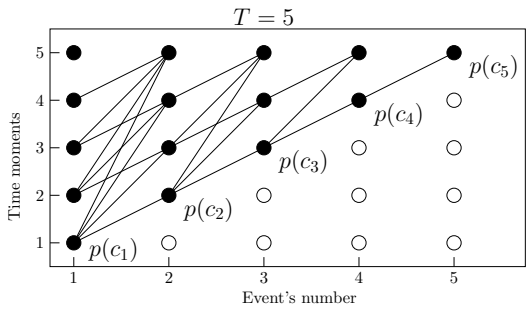


Figure 2: Possible times of recurrent changes. Black dots indicate the times at which the $i^{th}$ change may have appeared with non-zero probability.

Thus, Eq.(3.9) gives Pmf for $c_2$ (second column in Figure 2). Pmf for $c_3, c_4, c_5$ is calculated using the same procedure. Finally $\mathcal{P} = p(c_1) + p(c_2) + p(c_3) + p(c_4) + p(c_5)$. The online computation procedure for an arbitrary initial Pmf $p(c_1)$ is described in Algorithm 1.

**3.4 The exact PCCF for Gaussian distribution.** Consider Gaussain distribution for the settings when $p(x|\theta = (\mu, \sigma)) \sim 0$ for all $x \leq 0$.

Eq.(3.6) is a convolution of the Pmf $p(c_1)$ of the $1^{st}$ recurrent change, which is given as a boundary condition, and of the Pmf of the change $c_i$ calculated in the previous step:

$$\begin{aligned} p(c_{i+1}) &= (p(c_1) * p(c_i))[\tau] \\ &= \sum_{\tau=1}^{t-1} p(c_1 = t - \tau) p(c_i = \tau). \end{aligned} \tag{3.10}$$

The convolution of two Gaussian distributions (please see the proof in [14]) is

$$(p(x|\mu_1, \sigma_1) * p(x|\mu_2, \sigma_2)) = p(x|\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2}). \tag{3.11}$$

PCCF for recurrent changes can be written as a t-fold convolution and computed analytically

$$\mathcal{P}(t) = \underbrace{(p(c_1) * p(c_1) * \cdots * p(c_1))}_{\text{t}} = p(c_1)^{*t}. \tag{3.12}$$

PCCF for the moment $t$ is a sum

$$\mathcal{P}(t) = \sum_{l=1}^{t} \frac{1}{\sigma\sqrt{2\pi l}} \exp\left(\frac{-(t - l\mu)^2}{2l\sigma^2}\right). \tag{3.13}$$

We can calculate the limit $L$ of $\mathcal{P}(t)$ when $t \to \infty$

$$L = \lim_{t\to\infty} \sum_{l=1}^{\infty} \frac{1}{\sigma\sqrt{2\pi l}} \exp\left(-\frac{(t - \mu l)^2}{2l\sigma^2}\right). \tag{3.14}$$

We can replace $l$ by $t/\mu$ in the denominators since we are interested in the terms for which $l \sim t/\mu$ and when $t \to \infty$, $\lim_{t\to\infty}\left(\frac{1}{t} - \frac{1}{t/\mu}\right) = 0$, we are making approximation error of order $(1 + o(1))$,

$$L = \lim_{t\to\infty} \sum_{l=1}^{\infty} \frac{\sqrt{\mu}}{\sigma\sqrt{2\pi t}} \exp\left(-\frac{\mu^3(t/\mu - l)^2}{2t\sigma^2}\right). \tag{3.15}$$

Exponential terms corresponding to $l$, for which $|t/\mu - l| \gg \sqrt{t}$, will have small values which we can ignored. Therefore, we need to estimate the sum consisting of the terms for $l \in [t/\mu \pm \sqrt{t}]$. It is convenient to consider a wider interval of width $t^{3/5} > \sqrt{t}$. Let us consider three intervals for $l$ (1) $[0, t/\mu - t^{3/5})$, (2) $[t/\mu - t^{3/5}, t/\mu + t^{3/5}]$, (3) $(t/\mu + t^{3/5}, \infty)$. The components in the sum in Eq. 3.15 are very small within the intervals (1) and (3) since both are bounded by $\exp(-\frac{\mu^2}{2\sigma^2}N^{1/5})$. Therefore, we can find the limit by estimating the sum only within the interval (2):

$$L = \lim_{t\to\infty} \frac{\sqrt{\mu}}{\sigma\sqrt{2\pi t}} \sum_{l=-t^{3/5}}^{t^{3/5}} \exp\left(-\frac{\mu^3(t/\mu - l)^2}{2t\sigma^2}\right). \tag{3.16}$$

Eq.(3.16) is the Riemann sum for the Gaussian integral $\int_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\frac{\pi}{a}}$, thus

$$L = \frac{\sqrt{\mu}}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{\mu^3 x^2}{2\sigma^2}} dx = \frac{1}{\mu}. \tag{3.17}$$

Figure 3 illustrates two Gaussian PCCF functions (Eq. 3.13) for two cases ($\mu = 10, \sigma = 2$) and ($\mu = 15, \sigma = 3$). Local extrema of the PCCF function correspond to the time moments $l\mu, l \in \mathbb{Z}$. PCCF values converge to limits as defined in Eq.(3.17) $L = 1/10$ and $L = 1/15$.

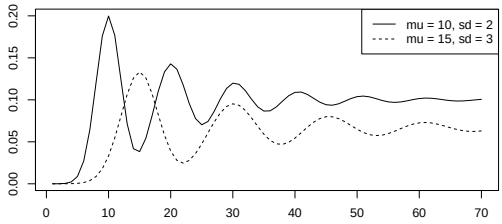


Figure 3: An example of two Gaussian PCCF functions.

**3.5 PCCF update after a confirmed change.** Typically, a detector would have no internal means to know for sure whether $\mathbf{e}_t^+$ is TP or FP. If we have a feedback mechanism indicating with some time delay $D$ when a change $c_i$ did happen, we can update PCCF

to make our detector more confident about the future change points. Given a known $c_i$, we recompute PCCF starting from that point. An updated PCCF is depicted by the dotted line in Figure 4. In Figure 4 we can see an example of how PCCF oscillates having local maximums at moments $t = k\mu$ until converging to the limit $L = 0.1$. An event at the moment $t = 200$ was confirmed as a change and new PCCF is calculated (dashed line).

If we do not reestimate $\theta$ then update procedure is very fast as we simply shift already computed PCCF along the time axis.

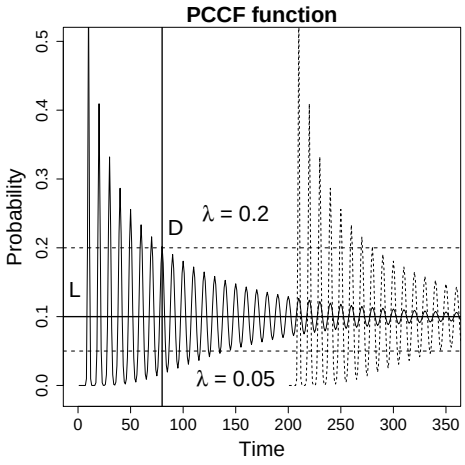


Figure 4: PCCF converges to the limit $L = 0.1$ depicted by the horizontal bold line. Two horizontal lines $\lambda$ illustrate possible thresholds above and below the limit. Vertical line $D$ depicts delay of change confirmation.

## 4 Integration of PCCF into a change detector

There are two straightforward approaches for using PCCF can be used to potentially improve the accuracy of existing change detectors: (1) post-processing – a detector with fixed settings is used; its output (CDEs) are adjusted based on PCCFs values, and (2) pre-processing – the sensitivity of the detector is dynamically adjusted according to the current PCCFs values.

At the beginning of the change detection process initial probabilities of changes in the foreseeable future are computed using PCCF. After that PCCF is recalculated every time the last confirmed $c_i$ is known.

In case of post-processing, a change detector with fixed settings, learned offline, is applied. When the detector alarms a change at time $t$, we check whether the probability of change given by PCCF is greater than a user defined threshold $\mathcal{P}(t) > p_h$. If it is, we count CDE as a change and output $\mathbf{e}_t^+$, otherwise we output $\mathbf{e}_t^-$.

In case of pre-processing, detector's settings are adjusted dynamically according to PCCF. The detector is made more sensitive when a change is expected with a higher probability $\mathcal{P}(t) > p_h$, and less sensitive when the change is expected with a lower probability. This strategy should be applied with caution in order not to make the detector too sensitive, which would result in an increase the FP rate. In this study we consider only scenario when the highest value of the dynamically adjusted sensitivity of the detector is equal to the optimal threshold value learned during the training phase. When the estimated probability of the change given by PCCF is low, the sensitivity threshold is lowered. This scenario is illustrated in Figure 5 where detector's sensitivity is a function of PCCF.

## 5 Experiments

To illustrate the utility of the PCCF we performed three experiments using artificially generated data and two experiments with real data. [2]. Our goal is to improve the accuracy of online change detection when recurrent changes are expected. The required information about recurrence is the time intervals between consecutive changes, given as the parameters of the normal distribution $N(\theta = \mu, \sigma)$.

**5.1 PCCF simulation.** In order to confirm the correctness of the PCCF Eq. (3.7), (3.13) expressing the probability to observe a recurrent change at every moment $t$, we performed a simulation by generating sequences of the recurrent changes multiple times and calculating frequencies of occurrence of generated time locations of the changes at each moment of time. The shape of the obtained curve of frequencies perfectly fits analytically computed PCCF function.

**5.2 Delay of change confirmation.** In this simulation we illustrate a procedure to estimate maximum delay of change confirmation $D$ during which the recurrence information in a form of parameters $\theta = (\mu, \sigma)$ is still useful and demonstrate the use of PCCF in post-processing settings.

From Figures 3 and 4 depicting behavior of the PCCF it can be seen that the probability of a change oscillates with decreasing amplitude and converges to the limit $L$. Assuming the probability of making a FP at any moment is constant and is equal to $\lambda$, there are three scenarios to consider:

1. The error rate $\lambda$ is higher than $\mathcal{P}(t)$ for all $t$. In this case the detection accuracy cannot be improved

[2] We have made the code for computing PCCF and running simulations and experiments publicly available at `https://github.com/av-maslov/Pccf`.

with recurrence information $\theta$;

2. $\lambda$ is higher than the limit $L$ but it is lower than the local maximums of the PCCF $\mathcal{P}(t)$. This case is depicted in Figure 4 by the horizontal dashed line with value 0.2.
3. $\lambda$ is lower than the limit $L$; this case is depicted with the horizontal dashed line with value 0.05.

We are not interested in the first case because the probability of making errors is so high that the baseline change detector should be optimized better first. In the second case peaks of $\mathcal{P}(t)$ at moments $t = i\mu$ are higher than $\lambda$ until some moment $D$ (vertical line), after which the probability of the error is always higher than the probability of change. The third case is equivalent to the first case till the moment when local minimums of $\mathcal{P}(t)$ become always larger than $\lambda$. These observations suggest that we can determine the maximum tolerable confirmation delay $D$ (Figure 4) by computing the PCCF function, estimating the error rate of detector, and finding the last peak of $\mathcal{P}(t)$ which is higher than $\lambda$. To confirm our hypothesis we run the following experiment.

*Input data description.* For each delay $D$ in the range from 5 to 300 we generate input signal 500 times with 50 recurrent change points with $\mu = 10$ and $\sigma = 0.7$. PCCF $\mathcal{P}(t|\mu, \sigma)$ is calculated and updated after the location of the last change is output with the current delay value $D$. Change point locations are generated according to Definition 2, i.e. given $c_1$, the second change is $c_2 = c_1 + \epsilon_1$, $c_3 = c_2 + \epsilon_3$ and so on, where $\epsilon_i$ is a sample from the Gaussian distribution. The error rate defining the expected number of FPs per $\mu$ is set to $\lambda = (0.05, 0.2)$. Time locations of outliers are sampled from the uniform distribution so that there are $1/\lambda$ outliers per time interval of width $\mu$ where $\lambda$ is the FP rate of the detector.

*Change detector.* In this and the next simulation for PCCF pre-processing we used artificially generated data streams, in which changes are abrupt step changes in the value of the signal. We use a simple base-level change detector, the output statistic for which is the first order difference of the signal $(x_2 - x_1, \ldots, x_n - x_{n-1})$. Its behaviour is illustrated in Figure 5. A CDE $\mathbf{e}_t^+$ is alarmed when

$$x_i - x_{i-1} > h \tag{5.18}$$

*Experimental results.* We measure the accuracy of the detector integrated with PCCF for post-processing settings by calculating its TP and FP rates (not the detector it is used to enhance). The TP rate is a fraction of changes in the input signal that are identified correctly. The FP rate is a fraction of CDEs that would be FPs if the base-level change detector alone was used without PCCF post-processing. The results are illustrated in Figure 6. (The corresponding PCCF can be seen in Figure 4.)

The left plot in Figure 6 illustrates the case when periodicity is beneficial for any delay, since the error rate $\lambda = 0.05$ is low and $L > \lambda$ almost all the time. By the time when PCCF converges to its limit all CDEs are considered as changes. That is why the fraction of correctly predicted changes (bold line) tends to 1 while the fraction of identified errors tends to 0 (dotted line).

The right plot illustrates the situation, when recurrence information is beneficial only in some cases, that is when the error rate is lower than the values of PCCF. The maximum tolerable delay $D$ is illustrated by the vertical line. Please observe that this line is located close to the moments when TP rate goes down *and* FP rate goes up meaning that improvement of the base detector is not possible any more. This confirms our hypothesis that the maximum change confirmation delay $D$ can be estimated by finding the closest local maximum of the PCCF which is higher than $\lambda$.

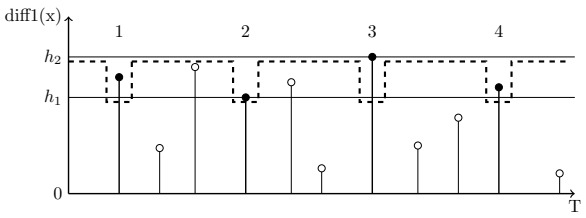


Figure 5: The base-level detector. The lower is the threshold the higher is the probability of error. Outliers between changes $c_1$, $c_2$ and $c_3$ are detected as changes when using constant value threshold $h_1$. The optimal dynamic threshold is depicted by the dashed line.

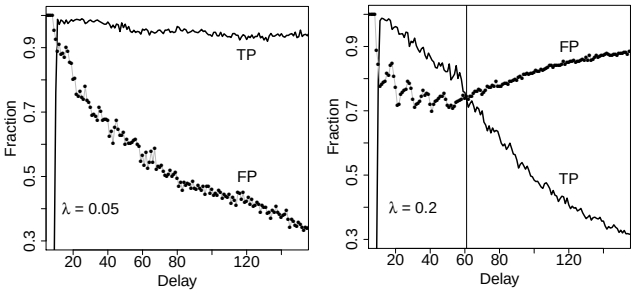


Figure 6: TP and FP rates of PCCF (not the base- or PCCF-enhanced detector) vs. the delay $D$ of change point confirmation.

**5.3 Dynamic adjustment of sensitivity.** This experiment demonstrates the pre-processing approach for integration of PCCF with a change detector, where detection sensitivity is adjusted online. We first pre-calculate PCCF with parameters $\mu$ and $\sigma$, and then dynamically change the parameters according to the most recent probability of recurrent change. Recall Figure 5 depicting a toy example of dynamically changing threshold $h$ (dashed line). When PCCF value is low, $h$ is set to a higher value $h_2$, when PCCF value is high, the detector threshold is set to $h_1$.

For this simulation we generate 200 times input signal with 15 recurrent changes and 15 uniformly distributed outliers. The average time interval between changes is $\mu = 10$ with the $\sigma = 0.5$. The error rate is set to $\lambda = 1$, i.e. 1 outlier per time interval of width $\mu$. To detect changes we used the same base-level detector as in the previous experiment.

The results are depicted in Figure 7, showing TP and FP rates. The performance of the detector without PCCF is depicted by the line with circles, obtained by varying the sensitivity threshold of the detector. The performance of the detector with PCCF is depicted by the line with red squares, obtained by changing delay of the $c_i$ confirmation $D$ for a fixed threshold of the base detector. We can see that performance of the detector can be improved using PCCF if $D < 2\mu$. This is in line with our theoretical results.

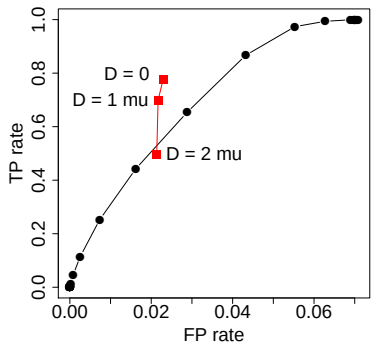


Figure 7: The trade-off between TPs and FPs. Red squares show performance with the dynamic threshold with delayed feedback $(0, \mu, 2\mu)$.

**5.4 Real datasets.** We experimented with two real datasets, the Boiler dataset that contains sensor readings containing recurrent refueling behavior, and the UK network traffic dataset with the typical recurrent aggregated traffic peaks.

**Boiler dataset.** Data comes from sensors installed on the scales measuring fuel mass in the container of Circulating Fluidized Bed boiler (CFB) [13]. The signal is shown in the top part of Figure 8, where the true changes are highlighted by vertical lines. The mass of fuel decreases continuously as it is being consumed. Abrupt changes in the signal happen at the start of the refueling process. The signal is fluctuating due to rotating parts of the system making it difficult to distinguish the true changes from noise.

We calculated a series of PCCF, updated after each provided confirmation with the delay $2\mu$ where $\mu$ is estimated from training data average distance between changes. The results are shown in Figure 8. PCCF functions are computed with confirmation delay $2\mu$ (bottom plot). Initial and updated PCCFs are marked by numbers. Two outliers are correctly identified by PCCF and do not cause FPs.

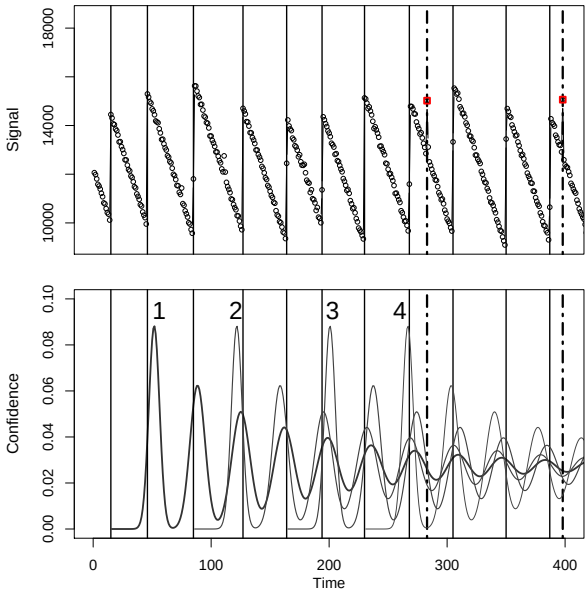


Figure 8: The mass flow signal with abrupt changes (top) and updated PCCFs (bottom). Outliers due to jammed particles (vertical dashed lines) in the mass flow do not result in FPs.

**The Internet traffic dataset.** Next, we test our approach on publicly available dataset[3] containing aggregated internet traffic data from Internet Service Provider in the UK academic network backbone. The data series is illustrate in Figure 9. Measurements were taken between 19 Nov 2004 and 27 Jan 2005 at five minute intervals. The task is to detect on-line maximum

[3] https://datamarket.com/data/list/?q=internet+traffic+data+price%3Afree

daily traffic by detecting changes in the trend of the signal. These changes are periodic; we do not use information of the daytime in order to consider them as recurrent.

Figure 10 shows TP and FP rates of the PCCF-based detector for post-processing settings. The results are analogues for the results obtained in the experiment on the artificial data illustrated in Figure 6. From the left plot we can see that max delay of confirmation is $D = 6\mu$.

Figure 11 illustrates improvement in the performance of the detector when using PCCF. The ROC curve (solid line) shows the trade-off between TPs and FPs of the detector. Each point on this curve indicates the performance of the detector with a fixed sensitivity threshold. The red triangles depict the performance of the PCCF-based detector with post-processing corresponding to the baseline 'Static' detector (denoted with a blue diamond) we chose to improve as one already having a high TP rate but poor performance wrt FPs. The red circles depict the performance of the PCCF-based detector with post-processing settings. Different triangles and circles correspond to different values of $D$. The triangles and circles concentrated in the top left corner with TP rate close to 1.0 and FP rate close to 0 correspond to setting when $D \leq 6\mu$. When $D > 6\mu$ the triangles are drifting down showing the deterioration in the TP rate. FP rate is not increasing because in both pre- and post-processing settings of PCCF-based detector we do not generate additions CDEs as we only reduced the sensitivity of a TP-optimal static detector we chose on the ROC.

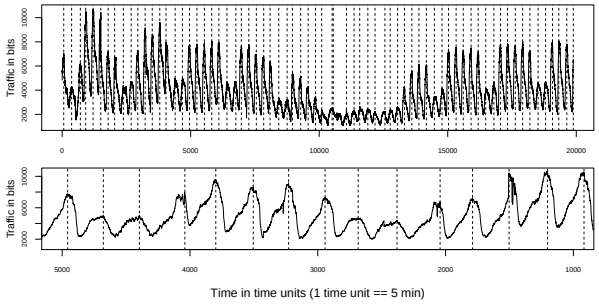


Figure 9: Aggregated traffic in the UK academic network. The bottom figure in a zoomed region. Dashed lines depict the true change points.

## 6 Conclusions

For handling recurrent changes we introduced the predictive change confidence function (PCCF), which is a probability mass function, conditioned on the time of

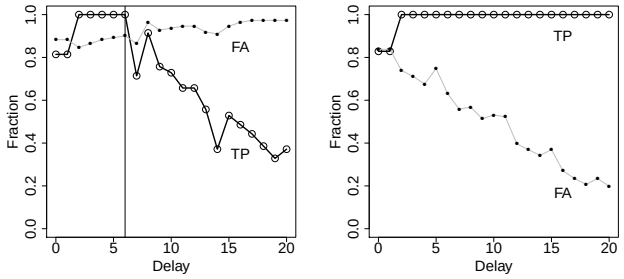


Figure 10: PCCF sensitivity and specificity for the internet traffic dataset. Threshold is higher (left) and lower (right) than PCCF limit. The maximum delay of confirmation is $6\mu$ (vertical line on the left plot).

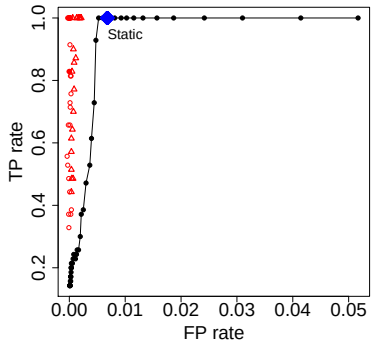


Figure 11: Black line - performance of the detector with static parameters. Red triangles - performance of PCCF based detector when applied to the static detector with settings corresponding to the point marked as 'Static'.

the last confirmed change. We derived the performance guarantees analytically, assuming Gaussian distribution of the time intervals between changes, and verified the analytical expression for PCCF experimentally on synthetic data. We proved that over time values of Gaussian PCCF converge to a limit value $L = \frac{1}{\mu}$ (Eq. 3.17). We demonstrated on both synthetic and real datasets the feasibility of utilizing PCCF for recurrent change detection with two simple approaches: post-processing of manifested changes and dynamic adjustment of the sensitivity of the base-level detector.

The following conclusions can be made regarding the benefits of using recurrence information for improving detections' performance:

- Recurrence information $\theta = (\mu, \sigma)$ is long-term useful if limit of the PCCF $L = \frac{1}{\mu}$ when $t \to \infty$ is greater than detector error rate $\frac{1}{\mu} > \lambda$ where $\mu$ is average time between recurrent changes.

- Recurrence information $\theta$ is short-term useful with the max confirmation delay $D$ if local maximums of the PCCF at moments $l\mu \leq D$ after confirmed change has higher values than probability of error $\lambda$, but limit $L < \lambda$.
- The main factor defining usefulness of $\theta$ is the ratio $\frac{\mathcal{P}(t=l\mu)}{\lambda}$ within time $D$ from the most recently confirmed change and $\frac{L}{\lambda}$ in a longer term when we do not have confirmation or $D$ is high.

In the performed experiments parameters $\theta$ were assumed to be known a priory or estimated from the training data.

This work provide several straightforward opportunities for future work, in particular investigating how uncertainty of $\theta$ estimates affects the performance of recurrent change detection, and studying other approaches for integrating PCCF with different existing change detection mechanisms.

## 7 Acknowledgments

This research is partly supported by STW CAPA project and COMAS funding. We would like to thank Jaakko Hollmen and Alexander Shklyaev for useful comments on this work. Any remaining errors are our own.

## References

[1] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, no. 1, pp. 69–101, 1996.

[2] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 44:1–44:37, 2014.

[3] I. V. Nikiforov and M. Basseville, "Detection of Abrupt Changes," 1993.

[4] A. S. Polunchenko and A. G. Tartakovsky, "State-of-the-Art in Sequential Change-Point Detection," *Methodology and Computing in Applied Probability*, vol. 14, no. 3, pp. 649–684, Oct. 2011.

[5] J. Gama and P. Kosina, "Learning about the learning process," in *Proc. of IDA'11*, pp. 162–172.

[6] J. B. Gomes, M. M. Gaber, P. A. C. Sousa, and E. M. Ruiz, "Mining recurring concepts in a dynamic feature space," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 1, pp. 95–110, 2014.

[7] J. B. Gomes, P. A. C. Sousa, and E. M. Ruiz, "Tracking recurrent concepts using context," *Intell. Data Anal.*, vol. 16, no. 5, pp. 803–825.

[8] H. Ang, V. Gopalkrishnan, I. Zliobaite, M. Pechenizkiy, and S. Hoi, "Predictive handling of asynchronous concept drifts in distributed environments," *IEEE Trans. on Knowl. and Data Eng.*, vol. 25, pp. 2343–2355, 2013.

[9] I. Zliobaite, "Change with delayed labeling: When is it detectable?" in *ICDM 2010 Workshops*, pp. 843–850.

[10] R. P. Adams and D. J. MacKay, "Bayesian online changepoint detection," Cambridge, UK, 2007.

[11] R. C. Wilson, M. R. Nassar, and J. I. Gold, "Bayesian online learning of the hazard rate in change-point problems," *Neural computation*, vol. 22, no. 9, pp. 2452–2476, 2010.

[12] D. Huang, Y. S. Koh, G. Dobbie, and R. Pears, "Detecting volatility shift in data streams," in *ICDM'2014*, pp. 863–868.

[13] M. Pechenizkiy, J. Bakker, I. Zliobaite, A. Ivannikov, and T. Karkkainen, "Online mass flow prediction in CFB boilers with explicit detection of sudden concept drift," *SIGKDD Explorations*, vol. 11, no. 2, pp. 109–116, 2009.

[14] P. Bromiley, "Products and convolutions of gaussian probability density functions," *Tina-Vision Memo*, vol. 3, 2003.

## A PCCF pseudo-code

In Algorith 1, in line 11, a zero-matrix of size $T$ is initialized with the first column filled by initial Pmf values $p(c_1)$. In line 12 a square lower triangular matrix (Eq. 3.9) is generated using the function `WeightsMatrix()`. Next the probabilities for individual changes in a sequence $\langle c_i \rangle_{i=1}^{k}$ are updated within the loop (line 14). PCCF is calculated in line 15 by summing up probabilities in the columns of matrix $P$.

**Algorithm 1** PCCF function pseudo-code

```
1:  function WeightsMatrix(T,θ)
2:      M = zeros(T, T); M[1, :] = 1:T
3:      for i = 2:T do
4:          for j = i:T do
5:              M[i,j] = M[i-1, j-1]
6:      return Pmf(M^T |θ)
7:  end function
8:
9:  function PCCF(T, θ = (μ, σ))
10:     P = zeros(T,T)
11:     P[:,1] = Pmf(1:T |θ)    ▷ Pmf fo the first change
12:     W = WeightsMatrix(T,θ)
13:     for i = 1:T-1 do
14:         P[i+1:T,i+1]=W[1:T-i,1:T-i] * P[i:end-1, i]
15:     return sum(P,2)                ▷ Sum of columns
16: end function
```