Inductive Conformal Martingales for Change-Point Detection

Denis Volkhonskiy, Evgeny Burnaev, Ilia Nouretdinov, Alexander Gammerman, Vladimir Vovk

Yandex School of Data Analysis
Skolkovo Institute of Science and Technology
Royal Holloway, University of London

dvolkhonskiy@gmail.com

June 19, 2017
Quickest Change-Point Detection
The **change-point detection problem** is formulated as following:

- $z_1, z_2, \ldots, z_n, \ldots$ are independent random variables;
- $z_1, z_2, \ldots, z_{\theta-1}$ are each distributed according to a distribution $f_0(z)$;
- $z_{\theta}, z_{\theta+1}, \ldots$ are each distributed according to a distribution $f_1(z)$;
- Change-Point (CP) $2 \leq \theta \leq \infty$ is unknown;

The task is to find the **stopping time** $\tau$, such that

- **Probability of False Alarm:** $\mathbb{P}(\tau \leq \theta) \leq \alpha$ for all $\theta$
- **Mean Delay:** $\mathbb{E}(\tau - \theta \mid \tau > \theta) \rightarrow \min \tau$
The **change-point detection problem** is formulated as following:

- \( z_1, z_2, \ldots, z_n, \ldots \) are independent random variables;
- \( z_1, z_2, \ldots, z_{\theta - 1} \) are each distributed according to a distribution \( f_0(z) \);
- \( z_\theta, z_{\theta + 1}, \ldots \) are each distributed according to a distribution \( f_1(z) \);
- Change-Point (CP) \( 2 \leq \theta \leq \infty \) is unknown;

The task is to find the **stopping time** \( \tau \), such that

- **Probability of False Alarm**: \( P(\tau \leq \theta) \leq \alpha \) for all \( \theta \)
- **Mean Delay**: \( E(\tau - \theta \mid \tau > \theta) \rightarrow \min_\tau \)
Important applications: healthcare, security, production, equipment maintenance, Internet of things, traffic routing, etc.

Challenges:

- Large volumes of time-series data;
- Complex, intractable or unknown dynamic models;
- Apparent non-stationary and quasi-periodicity.

Automatic Change-Point detection is critical in today’s world where the sheer volume of data makes it impossible to tag change-point manually.
**Figure:** EEG for a human in a room, where the light is turned off at time 387
Figure: Seismological signals. Earthquake starts at time 600
Standard Approaches

**Standard algorithms** for quickest Change-Point detection: Cumulative Sum (CUSUM), Shiryaev-Roberts, Posterior Probability statistics.

- Based on **likelihood ratio** $\frac{L_\theta^n}{L_n}$, where

\[
L_\theta^n = \prod_{i=1}^{\theta-1} f_0(z_i) \prod_{i=\theta}^{n} f_1(z_i)
\] (1)

the likelihood of observations $z_1, \ldots, z_n$ when $\theta \in [1, n]$, and by

\[
L_n = \prod_{i=1}^{n} f_0(z_i)
\] (2)

the likelihood of observations $z_1, \ldots, z_n$ without Change-Point.

- **Our goal** is to provide distribution-free algorithm for Change-Point detection, that is based on Conformal Martingales.
Conformal Martingales
Conformal Predictors

Conformal martingales:

- **Non-Conformity measure** (the measure of strangeness)
  \[ \alpha_i = A(z_i, \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\}) \]
  - Example: distance to the nearest neighbor.

- **P-values**

  \[ p_n = p(z_n, z_{n-1}, \ldots, z_1) = \frac{\#\{i : \alpha_i > \alpha_n\} + U \#\{i : \alpha_i = \alpha_n\}}{n} \]

  where \( U \sim Uniform[0, 1] \) independent of \( z_1, z_2, \ldots \)

  - Small p-values ⇒ strange objects (other distribution)
Conformal Predictors

Conformal martingales:

- **Non-Conformity measure** (the measure of strangeness)
  \[ \alpha_i = A(z_i, \{z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n\}) \]
  - Example: distance to the nearest neighbor.

- **P-values**
  \[ p_n = p(z_n, z_{n-1}, \ldots, z_1) = \frac{\#\{i : \alpha_i > \alpha_n\} + U\#\{i : \alpha_i = \alpha_n\}}{n} \]
  where \( U \sim Uniform[0, 1] \) independent of \( z_1, z_2, \ldots \)
  - Small p-values \( \Rightarrow \) strange objects (other distribution)
Conformal Martingales

Theorem

If p-values are not independent and uniformly distributed in \([0, 1]\), then \(z_1, \ldots, z_n, \ldots\) don’t satisfy the i.i.d. assumption. [Vovk et al., 2003]

- Conformal Martingale: \(S_n = \prod_{i=1}^{n} g_i(p_i), n = 1, 2, \ldots\), where \(g_i(p_i) = g_i(p_i | p_1, \ldots, p_{i-1})\) is what we call a betting function.

1. Strange objects (not i.i.d., Change-Point);
2. p-values are not Uniform;
3. penalize with betting function;
4. Martingale grows.

Betting Function

0.0 0.2 0.4 0.6 0.8 1.0
0.9 1.0 1.1 1.2 1.3 1.4

Denis Volkhonskiy (YSDA)  ICM for Change-Point Detection  June 19, 2017  10 / 26
Betting Functions

- **Constant** Betting function;
  \[ g(p) = \begin{cases} 
  1.5, & \text{if } p \in [0, 0.5), \\
  0.5, & \text{if } p \in [0.5, 1]. 
\end{cases} \]

- **Mixture** Betting Function;
  \[ g(p) = \int_0^1 \varepsilon p^{\varepsilon-1} d\varepsilon. \]

- **Kernel** Density Betting Functions;
  \[ g_n(p_n) = K_{p_{n-L}, \ldots, p_{n-1}}(p_n). \]
Contribution: Inductive Conformal Martingales for Change-Point detection

Conformal Martingale:

+ Theoretically valid;
− Computationally very inefficient;
− Wasn’t designed for quickest change-point detection.

Inductive Conformal Martingales:

▶ Computation of non-conformity scores with fixed training set:

\[ \alpha_i = A(z_i, \{z^*_{-(m-1)}, \ldots, z^*_0\}); \]

▶ Increase efficiency a lot: no need in Leave-One-Out-like method;
▶ Save validity.
Contribution: Inductive Conformal Martingales for Change-Point detection

**Conformal Martingale:**

- Theoretically valid;
- Computationally very inefficient;
- Wasn’t designed for quickest change-point detection.

**Inductive Conformal Martingales:**

- Computation of non-conformity scores with fixed training set:
  \[ \alpha_i = A(z_i, \{z^*_0, \ldots, z^*_{m-1}\}); \]
- Increase efficiency a lot: no need in Leave-One-Out-like method;
- Save validity.
Validity of Inductive Conformal Martingales

Figure: Validity Test of ICM: case of small train sets
Contribution: adaptation of ICMs for Change-Point detection

The **stopping rule** is \( \tau_{CM}^{S_n} = \inf \{ n : S_n \geq h \} \) for some constant \( h \).

**Problems:**

- It takes a lot of time to reach the change-point;
- Martingale can decrease to \(-\infty\).

**We propose:**

- ”Cut” the martingale:

\[
C_n = \max\{0, \, C_{n-1} + \log(g_n(p_n))\},
\]

(3)

where \( p_n \) is a p-value, and \( g_n \) is a betting function

- Stopping rule: \( \tau_{CM}^{C_n} = \inf \{ n : C_n \geq h \} \)

+ This modification performs better in terms of the *mean delay* until Change-Point detection;
Contribution: adaptation of ICMs for Change-Point detection

The **stopping rule** is \( \tau_{CM}^{S_n} = \inf \{ n : S_n \geq h \} \) for some constant \( h \).

**Problems:**
- It takes a lot of time to reach the change-point;
- Martingale can decrease to \(-\infty\).

**We propose:**
- "Cut" the martingale:

\[
C_n = \max\{0, C_{n-1} + \log(g_n(p_n))\},
\]

where \( p_n \) is a p-value, and \( g_n \) is a betting function

- Stopping rule: \( \tau_{CM}^{C_n} = \inf \{ n : C_n \geq h \} \)

+ This modification performs better in terms of the **mean delay** until Change-Point detection;
Figure: Example of the ICM in case of data with Change-Point (at $\theta = 500$) and without Change-Point
Contribution: Kernel Betting Function

Kernel Density Betting Function ([Fedorova et al., 2012], estimated from p-values seen before):

- Computationally inefficient;
- Need some time to update from uniform distribution after change-point;
- Theoretically has the best possible growing rate [Fedorova et al., 2012].

We propose to precompute it on p-values for data with an example of a change-point:

- Computationally efficient (no need to recompute);
- Faster Change-Point detection.
Contribution: Kernel Betting Function

Kernel Density Betting Function ([Fedorova et al., 2012], estimated from p-values seen before):

- Computationally inefficient;
- Need some time to update from uniform distribution after change-point;
+ Theoretically has the best possible growing rate [Fedorova et al., 2012].

We propose to precompute it on p-values for data with an example of a change-point:

+ Computationally efficient (no need to recompute);
+ Faster Change-Point detection.
Experiments
Oracles for Change-Point detection

**Classical approach** (CUSUM, Shiryaev-Roberst):
- **Optimal** in terms of Mean Delay;
- Need to know the data model.

**Conformal Martingales:**
- Only i.i.d assumption;
- We expect them to be worse.

**Oracles**, based on standard Change-Point detection algorithms:
- We assume the distribution class to be known;
- We don’t know the parameters;
- Likelihood:

\[
\bar{L}_n = \int \prod_{i=1}^{n} f(z_i \mid c_0) q(c_0) dc_0.
\]  

(4)
Oracles for Change-Point detection

**Classical approach** (CUSUM, Shiryaev-Roberst):
- **Optimal** in terms of Mean Delay;
- Need to know the data model.

**Conformal Martingales:**
- Only i.i.d assumption;
- We expect them to be worse.

**Oracles**, based on standard Change-Point detection algorithms:
- We assume the distribution class to be known;
- We don’t know the parameters;
- Likelihood:

\[
\bar{L}_n = \int \prod_{i=1}^{n} f(z_i \mid c_0) q(c_0) dc_0. \tag{4}
\]
Non-Conformity Measures:

- **k Nearest Neighbors (kNN NCM)** — average distance to \( k \) nearest neighbors.

- **Likelihood Ratio (LR NCM)** — the value of likelihood ratio \( \frac{f_1(x)}{f_0(x)} \).

As a **performance characteristics** we use:

- **Mean delay** until Change-Point detection \( \mathbb{E}_1(\tau - \theta \mid \tau > \theta) \),

- **Probability of False Alarm (FA)** \( \mathbb{P}_0(\tau \leq \theta) \).

Parameters:

- \( f_0 \sim \mathcal{N}(0, 1) \);
- \( f_1 \sim \mathcal{N}(m_1, 1), \quad m_1 \in \{1, 1.5, 2\} \);
- \( \theta \in [100, 200] \).
Mixture Betting Function. Comparison with Oracles

$\theta = 100, \mu_1 = 1$

$\theta = 200, \mu_1 = 1$

$\theta = 100, \mu_1 = 1.5$

$\theta = 200, \mu_1 = 1.5$

$\theta = 100, \mu_1 = 2$

$\theta = 200, \mu_1 = 2$
Kernel Density Betting Function. Comparison with Oracles

\[
\log_{10}(1 + E_{\tau \mid \tau > \theta})
\]

\(P_0(\tau \leq \theta)\)

\(\theta = 100, \mu_1 = 1\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR

\(\theta = 200, \mu_1 = 1\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR

\(\theta = 100, \mu_1 = 1.5\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR

\(\theta = 200, \mu_1 = 1.5\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR

\(\theta = 100, \mu_1 = 2\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR

\(\theta = 200, \mu_1 = 2\)

- CUSUM Oracle
- S-R Oracle
- Posterior Oracle
- ICM 7 NN
- ICM LR
Precomputed Kernel Density Betting Function.
Comparison with Oracles

\[ P_0(\tau \leq \theta) \]

\[ \theta = 100, \mu_1 = 1 \]

\[ \theta = 200, \mu_1 = 1 \]

\[ \theta = 100, \mu_1 = 1.5 \]

\[ \theta = 200, \mu_1 = 1.5 \]

\[ \theta = 100, \mu_1 = 2 \]

\[ \theta = 200, \mu_1 = 2 \]

\[ \log_{10}(1 + E_1[\tau - \theta | \tau > \theta]) \]

\[ \theta = 100, \mu_1 = 1 \]

\[ \theta = 200, \mu_1 = 1 \]

\[ \theta = 100, \mu_1 = 1.5 \]

\[ \theta = 200, \mu_1 = 1.5 \]

\[ \theta = 100, \mu_1 = 2 \]

\[ \theta = 200, \mu_1 = 2 \]
### Table: Mean Delay. Kernel Density Betting Function

<table>
<thead>
<tr>
<th>Param. \ Probab. of FA</th>
<th>( \theta = 200, \mu_1 = 1 )</th>
<th>( \theta = 200, \mu_1 = 1.5 )</th>
<th>( \theta = 200, \mu_1 = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>10%</td>
<td>5%</td>
</tr>
<tr>
<td>ICM kNN</td>
<td>54.14</td>
<td>28.70</td>
<td>28.70</td>
</tr>
<tr>
<td>CUSUM Oracle</td>
<td>37.78</td>
<td>37.80</td>
<td>37.80</td>
</tr>
<tr>
<td>S-R Oracle</td>
<td>37.80</td>
<td>15.16</td>
<td>15.16</td>
</tr>
<tr>
<td>Posterior Oracle</td>
<td>38.73</td>
<td>7.83</td>
<td>8.30</td>
</tr>
</tbody>
</table>

### Table: Mean Delay. Precomputed Kernel Density Betting Function

<table>
<thead>
<tr>
<th>Param. \ Probab. of FA</th>
<th>( \theta = 200, \mu_1 = 1 )</th>
<th>( \theta = 200, \mu_1 = 1.5 )</th>
<th>( \theta = 200, \mu_1 = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5%</td>
<td>10%</td>
<td>5%</td>
</tr>
<tr>
<td>ICM LR</td>
<td>14.14</td>
<td>7.24</td>
<td>4.90</td>
</tr>
<tr>
<td>ICM kNN</td>
<td>28.70</td>
<td>10.80</td>
<td>6.15</td>
</tr>
<tr>
<td>CUSUM Oracle</td>
<td>37.78</td>
<td>14.62</td>
<td>8.02</td>
</tr>
<tr>
<td>S-R Oracle</td>
<td>37.80</td>
<td>14.52</td>
<td>7.98</td>
</tr>
<tr>
<td>Posterior Oracle</td>
<td>38.73</td>
<td>15.16</td>
<td>8.30</td>
</tr>
</tbody>
</table>
Comparison with Optimal detectors

\[ P_0(\tau \leq \theta) \]

\[ \log_{10}(1 + E_1[\tau - \theta | \tau > \theta]) \]

\[ \theta = 100, \mu_1 = 1 \]

\[ \theta = 200, \mu_1 = 1 \]

\[ \theta = 100, \mu_1 = 1.5 \]

\[ \theta = 200, \mu_1 = 1.5 \]

\[ \theta = 100, \mu_1 = 2 \]

\[ \theta = 200, \mu_1 = 2 \]

CUSUM, S-R, PP, ICM 7 NN, ICM LR

Figure: Comparison with Optimal detectors. Precomputed Kernel Density Betting Function

Denis Volkhonskiy (YSDA)

ICM for Change-Point Detection

June 19, 2017 24 / 26
Conclusion

▶ An adaptation of Conformal Martingales for change-point detection problem was proposed;
▶ We demonstrated the efficiency of this approach by comparing it with natural oracles, which are likelihood-based change-point detectors;
▶ We proposed and compared several approaches to calculating a betting function;
▶ We also compared Inductive Conformal Martingales with methods that are optimal for known pre- and post-CP distributions, such as CUSUM, Shiryaev-Roberts and Posterior Probability statistics.
▶ Paper: [Volkhonskiy et al., 2017]
