Compound Decision for Parallel Sequential Change Detection

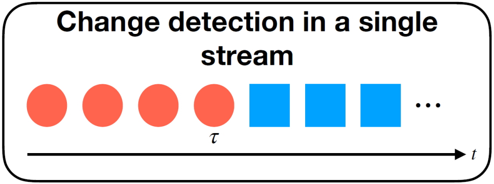
Yunxiao Chen

Department of Statistics, LSE

LSE Statistics Research Showcase

Single-Stream Sequential Change Detection

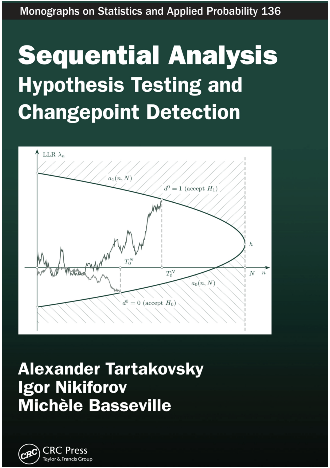
- Change time: τ
- Detection time: A stopping time T.
- Performance metric: Detection delay, false alarm, etc.



Single-Stream Sequential Change Detection (con'd)

Rich literature on this topic!

- Shewhart's control chart (Shewhart, 1931)
- CUSUM algorithm (Page, 1954)
- Shiryaev procedure (Shiryaev, 1963)
- ...



Statistica Sinica **11**(2001), 303-408

Celebrating the New Millennium: Editors' Invited Article - I

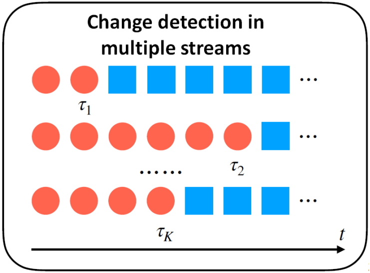
SEQUENTIAL ANALYSIS: SOME CLASSICAL PROBLEMS AND NEW CHALLENGES

Tze Leung Lai

Stanford University

Parallel Sequential Change Detection

- Data $X_{k,t}$: Data on stream k at time t.
- Detection time: Stopping times $\mathbb{T} = (T_1, ..., T_K)$.
- Stochastic control: Intervention on detected streams, which also changes the information filtration (e.g., deactivate the detected streams).



Parallel Sequential Change Detection (Con'd)

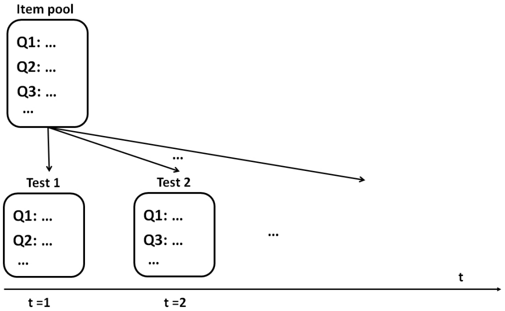
- Why do we want to study this problem? Can't we simply apply a single-stream change detection procedure to each stream individually?
- How do we formulate the problem and design methods?
- How do we assess their performance?

Parallel Sequential Change Detection (Con'd)

- Chen, Y. and Li, X. (2021). Compound Online Changepoint Detection in Parallel Data Streams. Statistica Sinica. Accepted.
- Chen, Y., Lee, Y-H, and Li, X. (2021). Item Quality Control in Educational Testing: Change Point Model, Compound Risk, and Sequential Detection. Journal of Educational and Behavioral Statistics. Accepted.
- Lu, Z., Chen, Y. and Li, X. (2022). Optimal Parallel Sequential Change Detection under Generalized Performance Measures. Submitted to IEEE Transactions on Signal Processing. Under review.

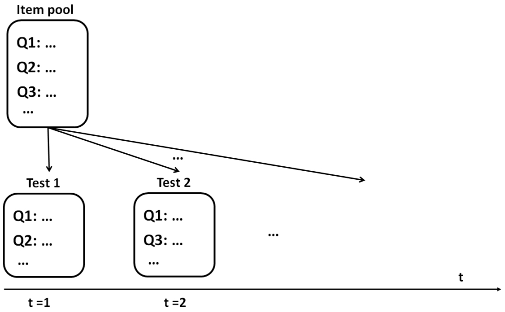
Why Parallel Sequential Change Detection?

Application I: Item Pool Quality Control (CL, 2021, CLL, 2021)



- Standardised testing and item pool.
- Each item is a data stream.

Application I: Item Pool Quality Control (con'd)



- Change point τ_k: The time point at which item k is leaked (e.g., by a test preparation company).

- Detection time T_k (stochastic control): Time to remove (review and revise) item k.

Application I: Item Pool Quality Control (con'd)

Goals:

- Control the proportion of leaked items (false non-discoveries) in the remaining item pool at **any time point**.
- Avoid making too many false alarms, i.e., make full use of the pre-change items.

There is a trade-off between the quality of the test and the financial cost for operating the test. This is a compound decision problem as the performance metrics are based on all the data streams.

Application II: Market Basket Analysis

- Customers purchase grocery online sequentially (Instacart, Amazon Fresh, Tesco ...).
- Each customer corresponds to a data stream.
- We want to detect dramatic changes to customers' shopping lists and make interventions (e.g., coupons).
- Goal: control and optimise some compound risk functions (to maximise the profit).

Other Applications

- Detect chip defects in cloud computing.
- Multi-channel spectrum sensing for cognitive radios (Chen, Zhang, and Poor, 2020).


The New York Times

Tiny Chips, Big Headaches

As the largest computer networks continue to grow, some engineers fear that their smallest components could prove to be an Achilles' heel.

Figure from New York Times, February 7, 2022

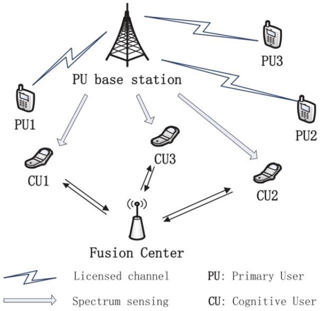


Figure from Chen, Zhang, and Poor (2020)

Why Parallel Sequential Change Detection?

- Multi-stream data are everywhere.
- Often, we need to make individualised decisions based on compound risks (which cannot be controlled using single-stream change detection methods).
- Methodologically interesting (compound decision, change detection, stochastic adaptive control, computation)!

Problem Formulation

A General Multi-stream Bayesian Change Point Model (CL, 2021)

- $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_K)$ follows a known prior distribution.
- Given $\boldsymbol{\tau}$, $X_{k,t}$'s are independent for different k and t.
- $$X_{k,t} \sim \begin{cases} p_{k,t} & \text{if } t \leq \tau_k \text{ (pre-change distribution)} \\ q_{k,t} & \text{if } t > \tau_k \text{ (post-change distribution)} \end{cases},$$

  where the pre- and post-distributions are known. (Still possible with partial information about these distributions; see CLL, 2021)

Example: An i.i.d. Model

- $\tau_1, \cdots, \tau_K \overset{i.i.d.}{\sim} Geo(\theta)$.
- Pre-change distribution (density): p.
- Post-change distribution (density): q.
- This model is a multi-dimensional extension of the classic Bayesian change model for a single data stream (Shiryaev, 1963).

Compound Sequential Change Detection Procedure (CL, 2021)

- At each time, we decide which streams to deactivate for the next time point based on information currently available.
  - This is a compound decision, because whether a stream will be deactivated or not depends on currently available information from all the streams.
- Once a stream is deactivated, it will not be reactivated and its data are no longer collected (other types of controls are possible, but we focus on deactivation here).
  - This procedure can be described by an index set process, denoted by S_t, where S_t is the set of active streams at time t.

Compound Sequential Change Detection Procedure (con'd)

Key components:

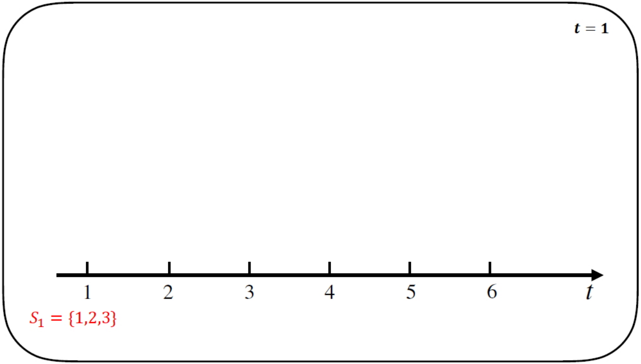
- $\mathcal{F}_t$: Information σ-field at time t.
- S_{t+1}: The set of active streams at time $t+1$, satisfying that S_{t+1} is $\mathcal{F}_t$ measurable, $S_{t+1} \subset S_t$, and $S_1 = \{1, ..., K\}$

  $\Rightarrow$ Detection time of the kth stream: $T_k = \sup\{t : k \in S_t\}$.

  $\Rightarrow \mathcal{F}_{t+1} = \sigma(\mathcal{F}_t, S_{t+1}, X_{k,t+1}, k \in S_{t+1})$.

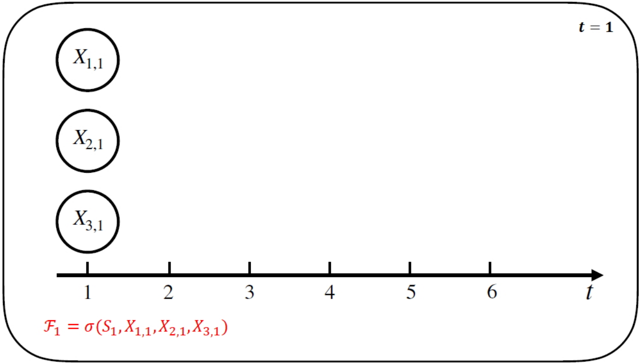
Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



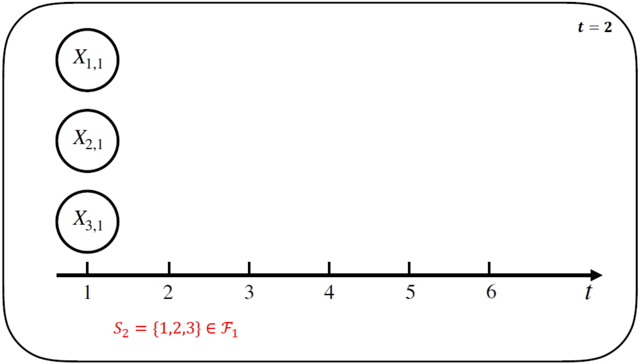
Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



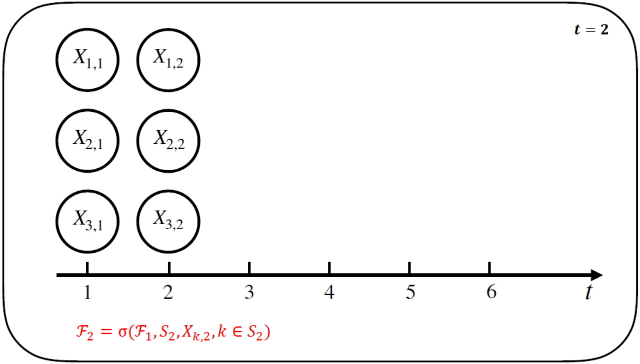
Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



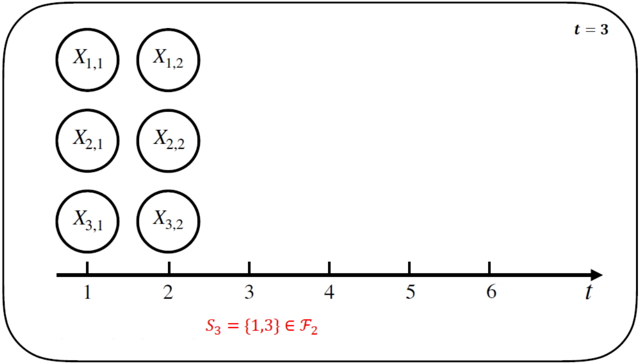
Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



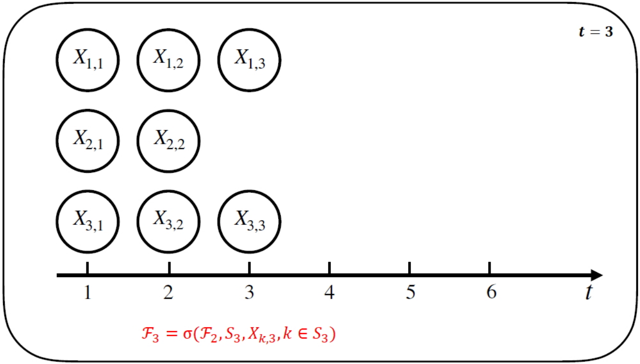
Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



Compound Sequential Change Detection Procedure (con'd)

A toy example with three streams ($K = 3$).



Compound Performance Metrics (LCL, 2022)

- In hypothesis testing, we control Type I error and try to maximise power.
- When designing a parallel change detection procedure, we also consider two compound performance metrics. At each time point, we control one metric, and try to optimise the other.
- The choice of performance metrics should depend on the application. In what follows, we give some examples. See LCL (2022) for other performance metrics.

Compound Performance Metrics (con'd)

Standardised educational testing:

- A test company would like to control the quality of the item pool at each time point, which may be measured by the false non-discovery proportion

$$\mathrm{FNP}_t = \frac{\sum_{k \in S_{t+1}} 1(\tau_k < t)}{|S_{t+1} \vee 1|} = \frac{\text{No. of active post-change streams}}{\text{No. of active streams}}$$

- As FNP is not observable, we control the local false non-discovery rate (LFNR)

$$\mathrm{LFNR}_t = \mathbb{E}[\mathrm{FNP}_t | \mathcal{F}_t].$$

Compound Performance Metrics (con'd)

Standardised educational testing:

- In the meantime, we may want to maximise the incremental run length (IRL), defined as

$$\mathsf{IRL}_t = \sum_{k \in S_{t+1}} \mathbb{1}_{\{\tau_k > t\}},$$

which indicates the total number of pre-change streams being used at each time.

- Again, as IRL is not observable, we hope to maximise its posterior mean, called the incremental average run length (IARL)

$$\mathsf{IARL}_t = \mathbb{E}[\mathsf{IRL}_t | \mathcal{F}_t].$$

Compound Performance Metrics (con'd)

Standardised educational testing:

- Trade-off between IARL and LFNR: More detections $\Rightarrow$ smaller LFNR & smaller IARL.
- Goal: maximize IARL while controlling LFNR to a prespecified level α (e.g., $\alpha = 0.05$) at any time.

Compound Performance Metrics (con'd)

Spectrum sensing for cognitive radios:

- Following the arguments in Chen, Zhang, and Poor (2020), it is of interest to control the false discovery proportion (FDP):

$$\mathrm{FDP}_t = \frac{\sum_{k \in S_t \setminus S_{t+1}} \mathbb{1}_{\{\tau_k \geq t\}}}{|S_t \setminus S_{t+1}| \vee 1},$$

  which concerns the quality of the detection set.

- Again, we can only control its posterior mean

$$\mathrm{LFDR}_t = \mathbb{E}[\mathrm{FDP}_t | \mathcal{F}_t].$$

Compound Performance Metrics (con'd)

Spectrum sensing for cognitive radios:

- In the meantime, we may want to minimise the incremental detection delay (IDD)

$$\mathrm{IDD}_t = \sum_{k \in S_{t+1}} \mathbb{1}_{\{\tau_k < t\}},$$

which indicates the total number of post-change streams being used at each time.

- Again, we need to consider its estimate

$$\mathrm{IADD}_t = \mathbb{E}[\mathrm{IDD}_t | \mathcal{F}_t].$$

Methods and Optimality

Proposed One-step Update Rule (control LFNR)

- **Input**: α, index set S_t, and posterior probability of active streams: $W_{k,t} = \mathbb{P}(\tau_k < t | \mathcal{F}_t)$ for $k \in S_t$.
- Sort $W_{k,t}$ for $k \in S_t$.
- Deactivate streams from the largest $W_{k,t}$ to the smallest one, until the average of the remainning ones is no larger than α.
- **Output**: Index set S_{t+1}.

Proposition: This one-step update rule guarantees $\mathrm{LFNR}_t \leq \alpha$.

Proposed One-step Update Rule (control LFDR)

- **Input**: α, index set S_t, and posterior probability of active streams: $W_{k,t} = \mathbb{P}(\tau_k < t | \mathcal{F}_t)$ for $k \in S_t$.
- Sort $W_{k,t}$ for $k \in S_t$.
- Deactivate streams from the largest $W_{k,t}$ to the smallest one, until the average of the selected ones is no larger than $1 - \alpha$.
- **Output**: Index set S_{t+1}.

Proposition: This one-step update rule guarantees $\mathrm{LFDR}_t \leq \alpha$.

$W_{k,t}$ in the i.i.d. Model

- $W_{k,0} = 0$ for all k, and

$$W_{k,t+1} = \begin{cases} \frac{q(X_{k,t+1})/p(X_{k,t+1})}{(1-\theta)(1-W_{k,t})/(\theta+(1-\theta)W_{k,t})+q(X_{k,t+1})/p(X_{k,t+1})} & \text{if } k \in S_{t+1} \\ W_{k,t} & \text{if } k \notin S_{t+1} \end{cases}$$

- Recall in the i.i.d. model, $\tau_1, ..., \tau_K \overset{i.i.d.}{\sim} Geo(\theta)$, with pre-change density p and post-change density q.

- Calculating $W_{k,t}$ is almost the same as in the Shiryaev procedure.

Proposed One-step Update Rule: Local Optimality

Proposition: The one-step update rule for controlling LFNR is locally optimal with respect to IARL, in the sense that the IARL_t based on S_{t+1} is no smaller than that based on any other $S \in \mathcal{F}_t$.

Proposition: The one-step update rule for controlling LFDR is locally optimal with respect to IADD, in the sense that the IADD_t based on S_{t+1} is no larger than that based on any other $S \in \mathcal{F}_t$.

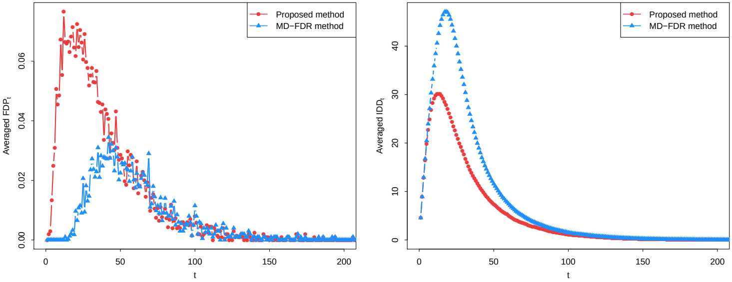
Proposed Compound Detection Method: Optimality

Theorem (CL, 2021, LCL, 2022): Under the i.i.d. model, if we keep running the proposed one-step update rule for controling LFNR (with threshold α), then this procedure (denoted by $\mathbb{T}^*$) is uniformly optimal with respect to IARL, in the sense that $\text{LFNR}_t(\mathbb{T}^*) \leq \alpha$ and for any $\mathbb{T}$ satisfying $\text{LFNR}_t(\mathbb{T}) \leq \alpha$ for all t, we have

$$\mathbb{E}(\text{IARL}_t(\mathbb{T})) \leq \mathbb{E}(\text{IARL}_t(\mathbb{T}^*)).$$

- We note that the procedure for controlling LFDR does not have uniform optimality, due to the lack of certain monotone properties.

Comparison with CZP (2020)



- FDP$_t$ (left) and IDD$_t$ (right) averaged over 1000 Monte Carlo simulations with $K = 100$.
- Goal: control FDR under $\alpha = 0.1$.

Extentions

- Partial information about the prior, pre- and post-distributions (CLL, 2021)
- Other types of controls
- Frequentist formulation and methods (currently considering a knock-off type procedure)
- Dependent streams
- Multi-stream point process data
- ...

Thank you!