Multiple changepoints Offline: Essential Statistical and Computational Concepts

Arnaud Liehrmann & Guillem Rigaill

Sorbonne Université & INRAE

February 27, 2025

Plan

Introduction

- Focus on detecting multiple changepoints.
- Penalized maximum likelihood approach with constant penalty.
- Implementation in Gaussian case using $O(n^2)$ dynamic programming.
- References: [Bellman, 1961, Auger and Lawrence, 1989, Jackson et al., 2005].

Likelihood Optimization

- Optimizing likelihood is a natural first approach for statisticians.
- Empirically competitive when properly tuned.
- Good statistical performance in applications and simulations.
- Example: see FPOP results with Yao's penalty [Maidstone et al., 2017, Fearnhead and Rigaill, 2020].

Complexity Considerations

- Quadratic complexity: $O(n^2)$
- Runtime (when implemented in C or some other low-level programming language)
  - $n = 10^4$: tens of minutes.
  - $n = 10^5$: 1-2 hours (varies with penalty).

- Quadratic complexity is a concern for larger datasets.

Complexity Considerations

Exact Computational Pruning

- Using inequality-based (PELT) or functional pruning (FPOP)
- PELT's pruning is generic [Killick et al., 2012]
  - Complexity: $\approx O(n^2/\hat{K})$ where $\hat{K}$ is estimated changepoints.
  - Theoretically $O(n)$ if $O(n)$ changepoints
- FPOP's pruning is much less generic (low dimensional models) [Maidstone et al., 2017, Pishchagina et al., 2024]
  - Complexity: $O \approx (n \log(n))$ even when there is no changepoints ($n = 10^8$ in 1 minute)
  - Theoretically $O(n)$ is $O(n)$ changepoints

Complexity Considerations

Local search and Isolation: Statistical Pruning

- For changepoints Max. Likelihood inference is not always possible or too slow
- Finding a solution that is
  - Easy to implement (to handle various models and penalization schemes)
  - Computationally fast
  - Good statistical guarantees
- Local Search and Isolation is an elegant solution [Fryzlewicz, 2014, Fryzlewicz, 2020, Cho and Kirch, 2019, Kovács et al., 2023, Verzelen et al., 2023]
  - Apply a LRT/Cusum-like strategy on sufficiently many chunks and aggregate

Conclusion

- Maximum likelihood (or more generally DP-based) approaches with constant penalty are a good baseline.
- Computational limitations (see previous slides)
- Statistical limitations:
  - Does not detect spurious changes.
  - Misses low-energy changepoints.
- Multi-scale penalty approaches may improve detection [Pein et al., 2017, Cho and Kirch, 2019, Verzelen et al., 2023].

Plan

Outline

Model and Likelihood

- Multiple changepoint detection extends single changepoint detection.
- Consider a time series $Y_{1:n}$ where each Y_t follows a distribution f_{Y_t}.
- Null hypothesis $\mathbf{H_0}$:
  - No changepoint in $Y_{1:n}$:

$$f_{Y_1} = f_{Y_2} = \cdots = f_{Y_n}.$$

- Alternative hypothesis $\mathbf{H_1}$:
  - K changepoints at unknown positions:

$$f_{Y_1} = \cdots = f_{Y_{\tau_1}} \neq \cdots \neq f_{Y_{\tau_K+1}} = \cdots = f_{Y_n}.$$

Objective of Changepoint Detection

- Estimate the number of changepoints K and their locations

$$\boldsymbol{\tau} = (\tau_1, \dots, \tau_K)$$

.

- Goal: Detect high-energy changepoints while avoiding spurious ones.

Notation and Definitions

- For any segmentation $\boldsymbol{\tau}$:
  - Define $\tau_0 = 0$
  - Define $\tau_{K+1} = n$.
- Define sets of segmentations:
  - $\mathcal{M}^K_{1:n}$: Set of all segmentations with K changes of n data points.
  - $\mathcal{M}_{1:n}$: Set of all segmentations of n data points.

$$\mathcal{M}_{1:n} = \bigcup_k \mathcal{M}^k_{1:n}$$

Exercise

Exercise

- Count the number of segmentations with K changes: $\#\mathcal{M}_n^K$.
- Count the total number of segmentations: $\#\mathcal{M}_n$.

Bonus

Count the number of segmentations in $K+1$ segments with length at least r.

Outline

Likelihood for a known number of changes

- Given a vector of changes $\boldsymbol{\tau}$ of size K, and $K+1$ parameters $\theta_1, \ldots, \theta_{K+1}$:

$$\prod_{k=1}^{K+1} \prod_{t=\tau_{k-1}+1}^{\tau_k} f_{Y_t}(\theta_k)$$

- Minus the log-likelihood:

$$\sum_{k=1}^{K+1} \sum_{t=\tau_{k-1}+1}^{\tau_k} -\log(f_{Y_t}(\theta_k))$$

Optimizing the Likelihood

- If the number of changepoints K is known:

$$\mathcal{L}_{1:n}^{K} = \min_{\boldsymbol{\tau} \in \mathcal{M}_n^K} \left\{ \sum_{k=1}^{K+1} \min_{\theta_k} \left(\sum_{t=\tau_{k-1}+1}^{\tau_k} -\log(f_{Y_t}(\theta_k)) \right) \right\}$$

If K is unknown

- That is replacing $\mathcal{M}_n^K$ by $\mathcal{M}_n$ in the previous equation
- Optimizing over all segmentations leads to a trivial segmentation with $n-1$ changes.

Balancing the Complexity with a penalty

- Mathematically deriving an appropriate/optimal penalty for some generic assumptions on the distribution is an open question (to the best of our knowledge)
- A fixed penalty $\beta > 0$ for each new change is a simple approach with some good computational and statistical properties
- Penalized maximum likelihood problem:

$$\mathcal{L}_{1:n} = \min_{\boldsymbol{\tau} \in \mathcal{M}_n} \left\{ \sum_{k=1}^{\#\boldsymbol{\tau}+1} \min_{\theta_k} \left(\sum_{t=\tau_{k-1}+1}^{\tau_k} -\log(f_{Y_t}(\theta_k)) + \beta \right) \right\}.$$

Key Questions

- How to set β?
  - Should be large enough to avoid spurious changes, but small enough to detect significant changepoints.
  - This is a statistical problem
- Given β, how to recover the segmentation optimizing the penalized likelihood?
  - This is an algorithmic problem.

Three Key Ingredients

For any multiple changepoint approach

[Truong et al., 2020]

1. **Model or Loss Function:** Measures likelihood fit to the data or homogeneity.
2. **Model Selection Criteria:** Balances goodness-of-fit with segmentation complexity (penalty based on number of changes, segment length, variance).
3. **Algorithm:** Explores segmentation space to output candidate segmentations with a good trade-off between likelihood and complexity.

Plan

Archetypical model and approach

Model: Univariate i.i.d. Gaussian errors.

Penalty: $c_1 \log(n) + c_2$ for some $c_1 \leq 2$ and $c_2 \leq 0$.

Algorithm: $O(n^2)$ with optimal partitioning algorithm.

Univariate Change in Mean Model

- Recall the i.i.d. Gaussian model in mean for a single changepoint (with known variance).
- In the multiple changepoint setting, under $\mathbf{H_1}$, there exist K changepoints at unknown positions $\boldsymbol{\tau}$:

$$\theta_1 = \cdots = \theta_{\tau_1} \neq \cdots \neq \theta_{\tau_k+1} = \cdots$$
$$\cdots = \theta_{\tau_{k+1}} \neq \cdots \neq \theta_{\tau_K+1} = \cdots = \theta_n.$$

Penalized Maximum Likelihood Problem

- The penalized maximum likelihood problem simplifies to minimizing the mean-squared error:

$$\mathcal{L}_{1:n} = \min_{\boldsymbol{\tau} \in \mathcal{M}} \left\{ \sum_{k=1}^{\#\boldsymbol{\tau}+1} \left(\sum_{t=\tau_{k-1}+1}^{\tau_k} (Y_t - \bar{Y}_{\tau_{k-1}:\tau_k})^2 + \beta \right) \right\}.$$

Exercises

Exercise

Derive the previous simplification.

Exercise

Implement in Python a function that simulates i.i.d. Gaussian data with several changes in the mean.

- It should take as parameters the positions of the changes and the mean of each segment.

Exercise

(AT HOME) Do the same for a change in the scale parameter of an Exponential distribution.

Outline

The Basic Idea

- The number of segmentations is large; for $n = 1000$ and $K = 3$:
$$\binom{1000-1}{3} > 1.66 \times 10^8.$$
- A naive search for the best segmentation is infeasible for small n and K.
- Key property: Knowing one change position t would give two simpler sub-problems

Formalizing the Idea

- Define $\ell_{i:j}$ as the optimal minus log-likelihood of segment $i:j$:

$$\ell_{i:j} = \min_{\theta} \left\{ \sum_{t=i}^{j} -\log(f_{Y_t}(\theta)) \right\}.$$

- In the Gaussian case:

$$\ell_{i:j} = \sum_{t=i}^{j} (Y_t - \bar{Y}_{i:j})^2.$$

- Minus the log-likelihood of a segmentation $\boldsymbol{\tau}$:

$$\sum_{k=1}^{\#\boldsymbol{\tau}+1} \left(\ell_{(\tau_{k-1}+1):\tau_k} + \beta \right).$$

Fundamental Recursion/Update Rule

- Update rule:

$$\mathcal{L}_{1:n} = \min_{\tau < n}\{\mathcal{L}_{1:\tau} + \ell_{\tau+1:n} + \beta\}. \quad \text{(this is a recursion)}$$

- This recursion is the basis for the dynamic programming algorithm.
- Found in various forms in literature [Bellman, 1961, Auger and Lawrence, 1989, Jackson et al., 2005].

Exercise

Exercise

Prove the update equation by contradiction.

Exercise

How many times do you need to apply the update to get $\mathcal{L}_{1:n}$?
What is the complexity of computing $\mathcal{L}_{1:n}$?

Keeping it Low in Memory

- The update rule leads to $O(n^2)$ time complexity assuming we have access to all $\ell_{i:j}$
- To apply this recursion, one might first think we need to store all $\ell_{i:j}$ for $1 \leq i < j \leq n$.
- Storing all these values scales as $\mathcal{O}(n^2)$, causing memory issues and slow empirical runtimes.
- Often, this can be avoided.

Efficient Computation of the segment Log-Likelihood

- For low-dimensional models, $\ell_{i:j}$ can often be computed efficiently using summary statistics.
- Example: Change in mean model with i.i.d. Gaussian errors:

$$\ell_{i:j} = \sum_{t=i}^{j}(Y_t - \bar{Y}_{i:j})^2 = \sum_{t=i}^{j} Y_t^2 - \frac{1}{j-i+1}\left(\sum_{t=i}^{j} Y_t\right)^2.$$

Some pre-computation for Efficiency

- Pre-compute in $O(n)$ time:
  - $S_j^{(1)} = \sum_{t=1}^{j} Y_t$.
  - $S_j^{(2)} = \sum Y_t^2$.
- Compute $\ell_{i:j}$ on the fly in $O(1)$ time:

$$\ell_{i:j} = G(i,j) = (S_j^{(2)} - S_{i-1}^{(2)}) - \frac{1}{j-i+1}(S_j^{(1)} - S_{i-1}^{(1)})^2.$$

General Applicability of this cumulative sum trick

- We have seen this already for LRT/Cusum statistics
- Works for many models, including:
  - Changes in parameters of distributions in the exponential family.
  - Changes in regression coefficients.
- For some models where this trick does not apply
  - Still possible to align the DP recursion with the calculation of $\ell_{i:j}$ to avoid storing more than n values [Celisse et al., 2018].

The Optimal Partitioning Algorithm

- Due to [Jackson et al., 2005] and very similar to [Bellman, 1961, Auger and Lawrence, 1989] with constraints on the number of segments.
- Need to recover both the optimal likelihood and the optimal set of changepoints.
- Store the arg min at each step: $\mathcal{T}_t$ for back-tracking after the OP recursion.

Generic Optimal Partitioning Algorithm

Algorithm 2 The generic Optimal Partitioning algorithm

Require: $\ell_{i:j}$ and β
Ensure: Optimal minus log-likelihood L_t and Argmin $\mathcal{T}_t$

1: $\mathcal{L}_{1:1} \leftarrow 0$
2: **for** $t \in \{2, \ldots, n\}$ **do**
3: $\quad \mathcal{L}_{1:t} \leftarrow \min_{\tau<t}(\mathcal{L}_{1:\tau} + \ell_{\tau+1:t} + \beta)$
4: $\quad \mathcal{T}_t \leftarrow \arg\min_{\tau<t}(\mathcal{L}_{1:\tau} + \ell_{\tau+1:t} + \beta)$
5: **end for**

Generic Optimal Partitioning Algorithm

Algorithm 3 The Optimal Partitioning algorithm in the Gaussian case

Require: Cumulative sum and sum of square of the data $S_t^{(1)}$, $S_t^{(2)}$
Ensure: Optimal minus log-likelihood L_t and Argmin $\mathcal{T}_t$

$\mathcal{L}_{1:1} \leftarrow 0$
for $t \in \{2, \ldots, n\}$ **do**
  $\mathcal{L}_{1:t} \leftarrow \min_{\tau<t}(\mathcal{L}_{1:\tau} + G(\tau+1, t) + \beta)$
  $\mathcal{T}_t \leftarrow \arg\min_{\tau<t}(\mathcal{L}_{1:\tau} + +G(\tau+1, t) + \beta)$
end for

Exercises

Exercise

Backtracking: Given the vector $\mathcal{T}_t$ for all t in $1:n$, how would you recover the optimal segmentation?

Exercise

Implement the OP algorithm for i.i.d. Gaussian errors in Python and the corresponding back-tracking algorithm.

exercise

Test your algorithm on simulated data and check its runtime complexity for various values of β.

Outline

Importance of Penalty β

- Choosing the value of β is crucial in changepoint detection.
- A practical approach involves simulation to minimize false positives.
- For i.i.d. Gaussian errors, a SIC-like penalty of $2\sigma^2 \log(n)$ is consistent (Yao, 1989).
  - The proof is mathematically involved
  - Some intuition of why it works using a larger penalty

Local Log-Likelihood Ratio

- Consider a chunk of data $i : j$
- Consider the statistic:

$$\sum_{t=i}^{j}(Y_t - \bar{Y}_{i:j})^2 - \sum_{t=i}^{\tau}(Y_t - \bar{Y}_{i:\tau})^2 - (Y_t - \bar{Y}_{\tau+1:j})^2$$

- A good penalty β should ensure this statistic is small with high probability under the null hypothesis.

$$\sum_{t=i}^{j}(Y_t - \bar{Y}_{i:j})^2 \leq \sum_{t=i}^{\tau}(Y_t - \bar{Y}_{i:\tau})^2 + (Y_t - \bar{Y}_{\tau+1:j})^2 + \beta$$

CUSUM Statistic (again)

- The statistic can be rewritten as the square of a Cusum

$$C_{i,\tau,j} = \sqrt{\frac{(\tau - i + 1)(j - \tau)}{j - i + 1}} (\bar{Y}_{\tau+1:j} - \bar{Y}_{i:\tau}).$$

- Controlled using a sub-Gaussian bound:

$$P(|C_{i,\tau,j}| > x) \leq e^{-1/2x^2}.$$

Controlling Probability

- There are $\binom{n}{3} \leq \frac{n^3}{6}$ triplet choices.
- Setting $x^2 = 6\log(n) - 2\log(\alpha)$:

$$P\left(\exists\ i < \tau < j \ \text{ such that } |C_{i,\tau,j}| \geq x\right) \leq \alpha.$$

A conservative bound

- Our union bound is conservative and ignores dependencies among $C_{i,\tau,j}$.
- The constant 2 from Yao (1989) is optimal and no smaller constant suffices (Wainwright, 2019).
- Still, similar bounds are often key to the proof of many papers

Plan

Exercise: Upper Bound on the number of changes

Exercise

Show that with $\beta = 6\log(n) - 2\log(\alpha)$, the segmentation $\boldsymbol{\tau}$ from the OP algorithm returns at most 2 changes between two true changepoints τ_k^* and τ_{k+1}^*.

Exercise: Lower Bound on the number of Changepoints

Exercise

Show that the segmentation $\boldsymbol{\tau}$ has at least one changepoint in the interval:

$$\left(\frac{\tau^*_{k-1} + \tau^*_k}{2}, \frac{\tau^*_k + \tau^*_{k+1}}{2} \right)$$

Assuming the size of the change is sufficiently large:

$$|\delta| \sqrt{2 \frac{(\tau^*_{k-1} + \tau^*_k)(\tau^*_k + \tau^*_{k+1})}{(\tau^*_{k-1} + \tau^*_{k+1})}} \geq \sqrt{C\beta}.$$

Practical Exercise

Exercise

- Test the OP algorithm on simulated data.
- Measure the control of the null hypothesis (H0) and the power to detect changes as a function of their height.

Practical Exercise 2

Exercise

- Repeat the previous exercise, but assume the variance is unknown.
- Pre-estimate the variance using MAD (Median Absolute Deviation) or HALL.

Practical Exercise 3

Exercise

- Refine your implementation of the OP algorithm.
- Incorporate a minimum segment length constraint.
- Analyze the algorithm's performance in the presence of outliers.

Auger, I. E. and Lawrence, C. E. (1989).
Algorithms for the optimal identification of segment neighborhoods.
Bulletin of mathematical biology, 51(1):39–54.

Bellman, R. (1961).
On the approximation of curves by line segments using dynamic programming.
Communications of the ACM, 4(6):284.

Celisse, A., Marot, G., Pierre-Jean, M., and Rigaill, G. (2018).
New efficient algorithms for multiple change-point detection with reproducing kernels.
Computational Statistics & Data Analysis, 128:200–220.

Cho, H. and Kirch, C. (2019).
Localised pruning for data segmentation based on multiscale change point procedures.
arXiv preprint arXiv:1910.12486.

Fearnhead, P. and Rigaill, G. (2020).

Relating and comparing methods for detecting changes in mean.
Stat, 9(1):e291.

Fryzlewicz, P. (2014).
Wild binary segmentation for multiple change-point detection.

Fryzlewicz, P. (2020).
Detecting possibly frequent change-points: Wild binary segmentation 2 and steepest-drop model selection.
Journal of the Korean Statistical Society, 49(4):1027–1070.

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. (2005).
An algorithm for optimal partitioning of data on an interval.
IEEE Signal Processing Letters, 12(2):105–108.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012).
Optimal detection of changepoints with a linear computational cost.

Journal of the American Statistical Association, 107(500):1590–1598.

Kovács, S., Bühlmann, P., Li, H., and Munk, A. (2023).
Seeded binary segmentation: a general methodology for fast and optimal changepoint detection.
Biometrika, 110(1):249–256.

Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. (2017).
On optimal multiple changepoint algorithms for large data.
Statistics and computing, 27:519–533.

Pein, F., Sieling, H., and Munk, A. (2017).
Heterogeneous change point inference.
Journal of the Royal Statistical Society Series B: Statistical Methodology, 79(4):1207–1227.

Pishchagina, L., Rigaill, G., and Runge, V. (2024).
Geometric-based pruning rules for change point detection in multiple independent time series.
Computo.

Truong, C., Oudre, L., and Vayatis, N. (2020).
Selective review of offline change point detection methods.
Signal Processing, 167:107299.

Verzelen, N., Fromont, M., Lerasle, M., and Reynaud-Bouret, P. (2023).
Optimal change-point detection and localization.
The Annals of Statistics, 51(4):1586–1610.