FPOP with labels

Toby Dylan Hocking

January 4, 2021

We have a sequence of d data points to segment, $\mathbf{z} \in \mathbb{R}^d$. We also have l labeled regions $R = \{(\underline{p}_1, \overline{p}_1, c_1), \dots, (\underline{p}_l, \overline{p}_l, c_l)\}$:

- $\underline{p}_j$ is the start of a labeled region,
- $\overline{p}_j$ is the end of a labeled region,
- $c_j \in \{0, 1\}$ is the number of changes in the region $[\underline{p}_j, \overline{p}_j]$.

For example $R = \{(\underline{p}_1 = 1, \overline{p}_1 = 2, c_1 = 0), (\underline{p}_2 = 2, \overline{p}_2 = 5, c_2 = 1)\}$ means that there is no change after the first data point, and there must be exactly one change somewhere between data points 2 and 5 (three possibilities). Assume the labeled regions are ordered:

$$1 \leq \underline{p}_1 < \overline{p}_1 \leq \underline{p}_2 < \overline{p}_2 \leq \dots \leq \underline{p}_l < \overline{p}_l \leq d \tag{1}$$

Thus the number of possible labels is $l \in \{0, 1, \dots, d-1\}$.

1 Optimization problem

We would like to find the best mean vector $\mathbf{m} \in \mathbb{R}^d$. Best is defined using the Optimal Partitioning objective function – a loss ℓ term for each data point, and a per-change penalty of $\lambda \geq 0$ [Maidstone et al., 2016]. We also add the constraints that the labels must be obeyed.

$$\begin{aligned}
\hat{C}_t = \min_{\mathbf{m} \in \mathbb{R}^t} \quad & \lambda \sum_{i=1}^{t-1} I(m_i \neq m_{i+1}) + \sum_{i=1}^{d} \ell(z_i, m_i) \qquad (2) \\
\text{subject to} \quad & \sum_{i=\underline{p}_j}^{\overline{p}_j - 1} I(m_i \neq m_{i+1}) = c_j \text{ for all } j \in \{1, \dots, l\}.
\end{aligned}$$

Note that I is the indicator function (1 if true, 0 otherwise).

SegAnnot is one algorithm that can be used in this context, but it is limited in that it can not detect any changes that are not labeled [Hocking and Rigaill, 2012]. In fact SegAnnot computes a model with exactly $N_1(R) = \sum_{(\underline{p}_j, \overline{p}_j, c_j) \in R} I(c_j = 1)$ changes.

The labels cover $S(R) = \sum_{(\underline{p}_j, \overline{p}_j, c_j) \in R} \overline{p}_j - \underline{p}_j$ possible changes, in which there are exactly $N_1(R)$ changes. There are thus $d - S(R)$ positions which are not labeled, which may or may not have changes. Thus the number of possible changes is in $[N_1(R), d - S(R) + N_1(R)]$.

2 Algorithm

We propose a new algorithm, LabeledFPOP, which solves (2). We define $C_t(\mu)$ as the optimal cost up to data point t if there is a mean of μ on the last data point t. The initialization is $C_1(\mu) = \ell(z_1, \mu)$.

2.1 Usual FPOP

The usual FPOP algorithm does not have label constraints [Maidstone et al., 2016]. The dynamic programming update rule for $t > 1$ is

$$C_t(\mu) = \ell(z_t, \mu) + \min \begin{cases} \hat{C}_{t-1} + \lambda & \text{if there is a change between } t-1 \text{ and } t, \\ C_{t-1}(\mu) & \text{if there is no change.} \end{cases} \tag{3}$$

Note that $\hat{C}_{t-1} = \min_\mu C_{t-1}(\mu)$ is the optimal cost up to data point $t-1$.

For LabeledFPOP the update rule (3) is used if there is no label between data points $t-1$ and t. Indeed, if $R = \{\}$ then there are no labels at all, so the LabeledFPOP problem is the same as the usual FPOP.

2.2 Negative label

If there is a $c_j = 0$ label such that $t \in \{\underline{p}_j + 1, \dots, \overline{p}_j\}$, then the update rule is simpler: the only possibility is no change.

$$C_t(\mu) = \ell(z_t, \mu) + C_{t-1}(\mu). \tag{4}$$

2.3 Positive label at the start

For a $c_j = 1$ label the update rule is a bit more complicated. Take a simple example to start. For a $(\underline{p}_j = 1, \overline{p}_j = 3, c_j = 1)$ label, the cost up to the end of the label is

$$\min_{\mu_1, \mu_2} \begin{cases} \lambda + \ell(z_1, \mu_1) + \ell(z_2, \mu_2) + \ell(z_3, \mu_2) & \text{for a change after 1} \\ \lambda + \ell(z_1, \mu_1) + \ell(z_2, \mu_1) + \ell(z_3, \mu_2) & \text{for a change after 2} \end{cases} \tag{5}$$

In general for a $(\underline{p}_j = 1, \overline{p}_j, c_j = 1)$ label, the cost up to the end of the label is

$$\min_{\mu_1, \mu_2} \min_{t \in [1, \overline{p}_j - 1]} \lambda + \underbrace{\sum_{j=1}^{t} \ell(z_j, \mu_1)}_{\mathcal{L}_{1,t}(\mu_1)} + \underbrace{\sum_{j=t+1}^{\overline{p}_j} \ell(z_j, \mu_2)}_{\mathcal{L}_{t+1, \overline{p}_j}(\mu_2)}. \tag{6}$$

We can use the recursive update rule for $t \in \{2, \dots, \overline{p}_j - 1\}$:

$$\mathcal{L}_{1,t}(\mu) = \mathcal{L}_{1,t-1}(\mu) + \ell(z_t, \mu). \tag{7}$$

And then for the optimal cost we initialize $t = 2$ to the cost of a change between 1 and 2:

$$C_2(\mu) = \hat{\mathcal{L}}_{1,1} + \lambda + \ell(z_2, \mu). \tag{8}$$

And then the update rule for $t \in \{3, \dots \overline{p}_j\}$ is:

$$C_t(\mu) = \ell(z_t, \mu) + \min \begin{cases} \hat{\mathcal{L}}_{1,t-1} + \lambda & \text{if there is a change between } t-1 \text{ and } t, \\ C_{t-1}(\mu) & \text{if the labeled change was before that.} \end{cases} \tag{9}$$

2.4 Positive label general case

In general for a $(\underline{p}_j, \overline{p}_j, c_j = 1)$ label, we need to compute the following quantities. First at $t = \underline{p}_j + 1$ we only need to consider one possibility (a change between $t - 1 = \underline{p}_j$ and $t = \underline{p}_j + 1$) to compute the optimal cost function,

$$C_t(\mu) = \ell(z_t, \mu) + \hat{C}_{t-1} + \lambda. \tag{10}$$

If the label spans more than one possible changepoint ($\underline{p}_j + 1 < \overline{p}_j$) then for all $t \in \{\underline{p}_j + 1, \dots, \overline{p}_j - 1\}$ we also need to compute the cost of no change since the start of the label, and store it in a separate function,

$$V_t(\mu) = \ell(z_t, \mu) + \begin{cases} C_{t-1}(\mu) & \text{if } t = \underline{p}_j + 1, \\ V_{t-1}(\mu) & \text{if } t \in \{\underline{p}_j + 2, \dots, \overline{p}_j - 1\}. \end{cases} \tag{11}$$

And for all $t \in \{\underline{p}_j + 2, \dots, \overline{p}_j\}$ the optimal cost is computed by taking the minimum of two functions,

$$C_t(\mu) = \ell(z_t, \mu) + \min \begin{cases} \hat{V}_{t-1} + \lambda & \text{if there is a change between } t-1 \text{ and } t, \\ C_{t-1}(\mu) & \text{if the change was before that.} \end{cases} \tag{12}$$

2.5 Pseudocode

Algorithm 1 Labeled Functional Pruning Optimal Partitioning Algorithm.

Input: data $\mathbf{z} \in \mathbb{R}^d$, non-negative penalty $\lambda \in \mathbb{R}_+$, sorted labels $(\underline{p}_j, \overline{p}_j, c_j) \in R$.
Output: optimal cost $\hat{\mathbf{C}} \in \mathbb{R}^d$, mean $\mathbf{m} \in \mathbb{R}^d$, previous segment end $\mathbf{T} \in \{1, \dots, d\}^d$.
$j \leftarrow 1$ // label counter
labelType $\leftarrow$ unlabeled
for t from 1 to d:
  if $t == 1$:
    cost $\leftarrow 0$
  else if labelType $== 0$:
    cost $\leftarrow$ prev_cost
  else if labelType $== 1$:
    change_cost $\leftarrow$ MinPiece(V) $+ \lambda$
    if $t == \underline{p}_j + 1$:
      cost $\leftarrow$ change_cost
    else:
      cost $\leftarrow$ MinOfTwo(change_cost, prev_cost)
    V += CostFun(z_t)
  else if labelType == unlabeled:
    change_cost $\leftarrow$ MinPiece(cost) $+ \lambda$
    cost $\leftarrow$ MinOfTwo(change_cost, prev_cost)
  cost += CostFun(z_t)
  if $t == \overline{p}_j$: // what update rule should we use next?
    labelType $\leftarrow$ unlabeled
    $j{+}{+}$
  if $t == \underline{p}_j$:
    labelType $\leftarrow c_j$
    if labelType $== 1$:
      $V \leftarrow$ cost
  $\hat{C}_t, m_t, T_t \leftarrow$ Minimize(cost)
  prev_cost $\leftarrow$ cost

2.6 Peak model with start/end labels

In the peak model of Hocking et al. [2020] there are two states: up/peak and down/background. We therefore need to compute two kinds of cost functions, $\overline{C}_t(\mu), \underline{C}_t(\mu)$ for all $t \in \{1, \dots, d\}$. We assume three different kinds of labels, $c_j \in \{0, 1, -1\}$, where 0 means no peaks in the region, 1 means exactly one change up, and -1 means exactly one change down.

For t in an unlabeled region the update rules are

$$\overline{C}_t(\mu) = \ell(z_t, \mu) + \min\{\overline{C}_{t-1}(\mu), \underline{C}^{\leq}_{t-1}(\mu) + \lambda\}, \tag{13}$$

$$\underline{C}_t(\mu) = \ell(z_t, \mu) + \min\{\underline{C}_{t-1}(\mu), \overline{C}^{\geq}_{t-1}(\mu)\}. \tag{14}$$

For t in a $c_j = 0$ (no peaks) label,

$$\overline{C}_t(\mu) = \infty, \tag{15}$$

$$\underline{C}_t(\mu) = \ell(z_t, \mu) + \min\{\underline{C}_{t-1}(\mu), \overline{C}^{\geq}_{t-1}(\mu)\}. \tag{16}$$

$$\tag{17}$$

(the ∞ should be treated as a special case in the min operator).

For t in a $c_j = 1$ (peak start) label, we simplify by assuming we need to be in down state at the start of the label and up state at the end of the label.

$$\overline{C}_t(\mu) = \begin{cases} \infty & \text{if } t = \underline{p}_j, \\ \ell(z_t, \mu) + \min\{\overline{C}_{t-1}(\mu), \underline{C}^{\leq}_{t-1}(\mu)\} & \text{otherwise.} \end{cases} \tag{18}$$

$$\underline{C}_t(\mu) = \ell(z_t, \mu) + \begin{cases} \min\{\underline{C}_{t-1}(\mu), \overline{C}^{\geq}_{t-1}(\mu)\} & \text{if } t = \underline{p}_j, \\ \infty & \text{if } t = \overline{p}_j, \\ \underline{C}_{t-1}(\mu) & \text{otherwise.} \end{cases} \tag{19}$$

The update rule for t in a $c_j = -1$ (peak end) label is analogous.

Labels could be inconsistent if $\overline{p}_j = \underline{p}_{j+1}$, e.g., two peakStart labels right next to each other. To deal with that, we can enforce $\overline{p}_j < \underline{p}_{j+1}$ for the peak model.

References

T. D. Hocking and G. J. Rigaill. SegAnnot: an R package for fast segmentation of annotated piecewise constant signals. HAL technical report 00759129, 2012.

T. D. Hocking, G. Rigaill, P. Fearnhead, and G. Bourque. Constrained Dynamic Programming and Supervised Penalty Learning Algorithms for Peak Detection in Genomic Data. *Journal of Machine Learning Research*, 21(87):1–40, 2020. URL `http://jmlr.org/papers/v21/18-843.html`.

R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, pages 1–15, 2016. ISSN 1573-1375.