

Bayesian block decomposition.

Fyodor Sviridov

1 Introduction

Implementation of Scargle’s Bayesian Blocks algorithm is presented (Scargle et al. 2013). This algorithm is used for optimal segmentation of data into blocks that characterize statistically significant variations. This algorithm has a certain elegance, as it is a great example of utilizing dynamic programming for a problem that cannot be solved in a brute straightforward way. The data may originate from different fields, but we used this method for time series analysis in astrophysics to determine the ranges of signal and background in light curves of gamma-ray bursts. Here, we will consider binned data, but the algorithm is not constrained to it. For further generalizations, we refer the reader to the original paper.

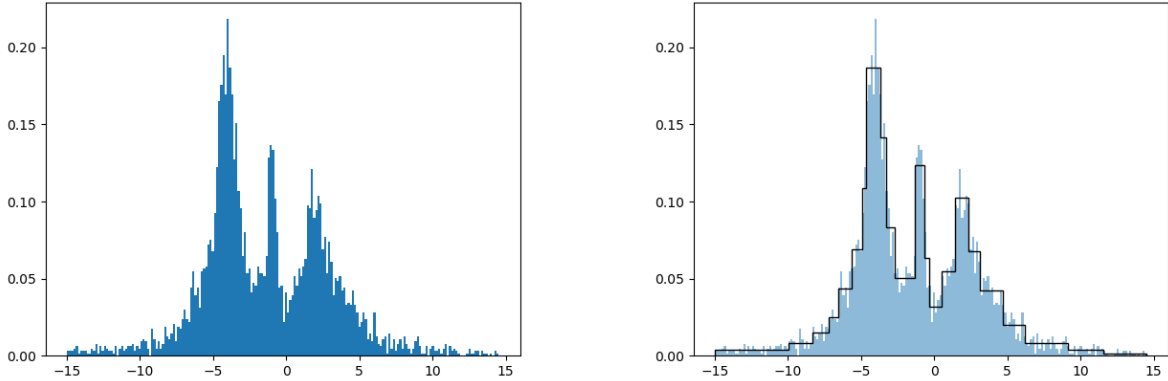


Figure 1: Bayesian block segmentation of an example dataset.

The task is to find some optimal segmentation by maximizing a certain criterion, referred to as a *fitness function*. As we will see below, it’s important that the total fitness function of a partition is block-additive. It must be the sum of the fitness functions of blocks, each evaluated only on the data within its respective block.

$$F[P] = \sum_{k=1}^{N_{\text{blocks}}} f(B_k) \quad (1)$$

where F is the total fitness of the partition P and $f(B_k)$ is the fitness of block k .

If there are N cells (bins), then the number of all possible partitions of the data is 2^N . For $N = 100$ the optimization task becomes infeasible. However, due to the block-additivity of the fitness function, we can employ a dynamic programming algorithm, that is very similar to a mathematical proof by induction. We incrementally add one bin, and to solve the task for $n + 1$ bins, we use saved information derived during previous steps ($\leq n$).

2 Outline of the algorithm

Let $P^{\text{opt}}(R)$ denote the optimal partition of the first R bins. In the initial case $R = 1$, the only possible partition is trivially optimal - a single block consisting of the first bin by itself. Once we have completed step R , having found the optimal partition $P^{\text{opt}}(R)$, we store the value of optimal fitness function in an array **best** and the location of the last change point of the optimal partition is stored

in an array **last** (change point is the bin index where a block starts). To obtain the optimal partition $P^{\text{opt}}(R+1)$, we consider partitions where the last block starts from r bin ($1 \leq r \leq R+1$). Let fitness function of this last block be $F(r)$. Under the additivity assumption, the fitness of $P^{\text{opt}}(R+1)$ is the sum of $F(r)$ and the fitness of $P^{\text{opt}}(r-1)$, saved from previous steps in the array **best**.

$$A(r) = F(r) + \begin{cases} 0, & r = 1 \\ \text{best}(r-1), & 2 \leq r \leq R+1 \end{cases} \quad (2)$$

Optimal value maximizing $A(r)$ is computed:

$$r^{\text{opt}} = \text{argmax}[A(r)]$$

After performing of all N steps, we can obtain final optimal partition $P^{\text{opt}}(R+1)$, using the information contained in the array **last**, where we have stored the index r^{opt} at each step. Indexes of change points in reverse order are

$$i_1 = \text{last}(N), \quad i_2 = \text{last}(i_1 - 1), \quad i_3 = \text{last}(i_2 - 1), \quad \dots \quad (3)$$

The beauty of this algorithm is that it performs a complete search of the optimum in the space 2^N partitions in time of order N^2

3 Fitness function

3.1 Piecewise constant model

Now it is necessary to determine the fitness function of a certain model. For binned data, we will consider here *piecewise constant model*. In many cases in astrophysics, we deal with the Poisson distribution. Therefore, the likelihood function of the n -th bin equals:

$$L_n = \frac{(\lambda W_n)^{N_n} e^{-\lambda W_n}}{N_n!} \quad (4)$$

where N_n is the detected number of events in bin n , λ is the actual event rate in counts per unit time, and W_n is the bin width in time units. For a piecewise constant model, λ remains constant inside the block. So the likelihood function of the block is:

$$L^{(k)}(\lambda) = \prod_n L_n = \lambda^{N^{(k)}} e^{-\lambda W^{(k)}} \quad (5)$$

where $N^{(k)} = \sum_n N_n$ and $W^{(k)} = \sum_n W_n$ are the total number of counts and the block width, respectively. Here, we are omitting a constant term that depends only on the full sample, is irrelevant to the fitness when summed over all blocks. The log-likelihood is

$$\ln L^{(k)}(\lambda) = N^{(k)} \ln \lambda - \lambda W^{(k)} \quad (6)$$

The maximum is reached when $\lambda_{\text{max}} = N^{(k)}/W^{(k)}$, which has the meaning of the average count rate in the block. The maximum value of the log-likelihood is the fitness function of the block:

$$f^{(k)} = N^{(k)} \ln \frac{N^{(k)}}{W^{(k)}} - N^{(k)} \quad (7)$$

3.2 Number of blocks

But under the derivation above, we have not considered how many blocks there are in a partition. It remains to implicitly include a parameter in the fitness that corresponds to the number of blocks in a model. For this purpose, we need to multiply the likelihood by a prior distribution for the number of blocks:

$$P(N_{\text{blocks}}) = p_0 \gamma^{-N_{\text{blocks}}} \quad (8)$$

where p_0 is a normalization constant, $0 \leq N_{\text{blocks}} \leq N$ and $\gamma \geq 1$. By choosing this prior it is much more likely that $N_{\text{blocks}} \ll N$, as we assume in advance. Then, the posterior probability (the total fitness of a partition $F[P]$) equals:

$$F[P](\vec{\lambda}, N_{\text{blocks}}) = \prod_{k=1}^{N_{\text{blocks}}} [L^{(k)}(\lambda_k)] \cdot P(N_{\text{blocks}}) \quad (9)$$

And :

$$\begin{aligned} \ln F[P] &= \sum_{k=1}^{N_{\text{blocks}}} \ln L^{(k)} - N_{\text{blocks}} \ln \gamma = \\ &= \sum_{k=1}^{N_{\text{blocks}}} (\ln L^{(k)} - \ln \gamma) \end{aligned} \quad (10)$$

So maximizing it over $\vec{\lambda}$ leads us to the final fitness function of a partition:

$$\ln F[P] = \sum_{k=1}^{N_{\text{blocks}}} f^{(k)} \quad (11)$$

$$f^{(k)} = \left(N^{(k)} \ln \frac{N^{(k)}}{W^{(k)}} - N^{(k)} \right) - \text{ncp_prior} \quad (12)$$

where $f^{(k)}$ is the fitness function of k-th block, $N^{(k)} = \sum_n N_n$ and $W^{(k)} = \sum_n W_n$ are the total number of counts and the block width, respectively. $\text{ncp_prior} \equiv \log \gamma$ is the parameter from 0 to $+\infty$, that controls the number of blocks, with a zero value corresponding to $N_{\text{blocks}} = N$, while a large value corresponds to one or just few blocks covering all data.

3.3 Piecewise linear model

A derivation for the piecewise linear fitness function follows similar lines, we need only to introduce a new inclined parameter a . In this model, data can be changed inside a block:

$$\lambda \rightarrow \lambda(1 + a(n - n_i)) = \lambda(1 + a\Delta n) \quad (13)$$

where n is the bin number, n_i is the first bin number of some block. Log-likelihood function of the k-th block equals

$$\ln L^{(k)} = N^{(k)} \ln \lambda + \sum_{n=n_i}^{n_f} N_n \ln[(1 + a\Delta n)W_n] - \sum_{n=n_i}^{n_f} \lambda(1 + a\Delta n)W_n \quad (14)$$

For solving the piecewise constant model, we previously omitted the second term, but now it is necessary due to its dependency on the new parameter. From $d/d\lambda \ln L^{(k)} = 0$, we can find λ_{max}

$$\lambda_{\text{max}} = \frac{N^{(k)}}{\sum_n (1 + a\Delta n)W_n} \quad (15)$$

And thereby,

$$\ln L^{(k)}(\lambda_{\text{max}}, a) = N^{(k)} \left(\ln \frac{N^{(k)}}{\sum_n (1 + a\Delta n)W_n} - 1 \right) + \sum_n N_n \ln[(1 + a\Delta n)W_n] \quad (16)$$

The maximum value of this equation defines the fitness function of k-th block (omitting the subtraction of ncp_prior). If $a = 0$ we get the already well-known function for the piecewise constant model. There is a lower bound on the parameter a , arising from the requirement that expectation of $\lambda(1 + a\Delta n)$ events should be positive.

$$a > -1/N$$

4 References

[1] Scargle, J et al. (2013), ApJ, Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations, 764, 2, 167. [doi:10.1088/0004-637X/764/2/167](https://doi.org/10.1088/0004-637X/764/2/167)