Unsupervised Segmentation of Poisson Data

Robert D. Nowak
Dept. of Electr. and Comp. Eng.,
Rice University
Houston, TX, USA
E-mail: nowak@rice.edu

Mário A. T. Figueiredo
Instituto de Telecomunicações,
Instituto Superior Técnico
1049-001 Lisboa, PORTUGAL
E-mail: mtf@lx.it.pt

Abstract

This paper describes a new approach to the analysis of Poisson point processes, in time (1D) or space (2D), which is based on the minimum description length *(MDL) framework. Specifically, we describe a fully unsupervised recursive segmentation algorithm for 1D and 2D observations. Experiments illustrate the good performance of the proposed methods.*

1. Introduction

Data modelled by Poisson statistics arises in many areas [1], such as bio-medical imaging (*e.g.*, nuclear imaging, electron microscopy), and particle and astronomical physics. Specifically, photon-limited data is acquired by detecting and counting individual photons. In this paper, we address a basic and important analysis problem: from an observed realization of a Poisson process (1D or 2D) we wish to *parse*, or *segment*, the observation space into regions that are well described as having homogeneous intensity.

To deal with this problem, we develop a method based on Rissanen's *minimum description length* (MDL) principle [2]. One interesting aspects of our development is that we are able to derive MDL criteria without recourse to the usual asymptotic approximations. Hence, our application of MDL here is especially simple and well motivated. Finally, we point out that our work can be seen as a coding-theoretic (unsupervised) alternative to related Bayesian methods presented in [3], [4], and [5]. Namely, our recursive method is related to the *Bayesian blocks* procedure in [5]; however, the selection rule in [5] differs considerably from our MDL criterion, and only 1D data is considered there.

2. The Mininum Description Length Principle

The MDL criterion addresses the following question: given a set of generation models, which one best explains the observed data? To formalize the notion of "best explanation," Rissanen introduced the following coding-theoretic thought-experiment [2]. Suppose that we wish to transmit the observed data x to a hypothetical receiver. If we are in possession of a (probabilistic) generation model for the data, say $p(x|\theta)$, the Shannon-optimal code length is well known to be $-\log p(x|\theta)$. Of course, the receiver needs to know the model parameters θ in order to build a decoder. If θ is *a priori* unknown, we also need to estimate it, code it, and transmit it. Now, consider a set of K competing model classes $\{p_i(x|\theta_i)\}_{i=1}^{K}$. In each class i, the "best" model is the one that gives the minimum code length,

$$\widehat{\theta}_i = \arg\min_{\theta_i} \{-\log p_i(x|\theta_i)\} = \arg\max_{\theta_i} p_i(x|\theta_i);$$

this is simply the *maximum likelihood* (ML) estimate. However, if the class is unknown, the "best" overall model, according to the MDL criterion, is the one leading to the minimum *description length*: the sum of $-\log p_i(x|\theta_i)$ with the length of the code for θ_i itself. The key aspect of MDL is that it performs model selection (which ML alone does not) by penalizing more complex model classes (requiring longer parameter code lengths).

The delicate issue in applying MDL is in how to encode the parameter θ_i; problems arise because real-valued parameters have to be truncated in order to be encoded by finite code-words. Usually, parameter code lengths are based on asymptotic approximations; *e.g.*, the well known $(1/2)\log N$, where N is the amount of data, is an asymptotically optimal parameter code length [2]. In this paper, we are able to work with (non-asymptotic) exact code lengths.

3. The Basic Criterion

The simplest instance of our approach can be described as follows. Let x_1 and x_2 be two counts which are samples of two Poisson variables of intensities λ_1 and λ_2, *i.e.*,

$$p(x_1|\lambda_1) = e^{-\lambda_1}\frac{\lambda_1^{x_1}}{x_1!}, \qquad p(x_2|\lambda_2) = e^{-\lambda_2}\frac{\lambda_2^{x_2}}{x_2!};$$

the model selection problem we wish to address is: are λ_1 and λ_2 equal or different? To attack this question with MDL tools, imagine we wish to transmit x_1 and x_2. To do so, we start by sending the sum $x_t = (x_1 + x_2)$, which can be done, for example, by using Elias' technique for arbitrary integers [2] (as we shall see, this code length for x_t is irrelevant for the resulting model selection criterion). Then, we send one of the counts, say x_1, from which the receiver can easily infer the other, $x_2 = x_t - x_1$. Now consider two model classes leading to two possible description lengths.

Model Class 0: Here, $\lambda_1 = \lambda_2$. In this case, the probability function of x_1, conditioned on x_t, is simply binomial with parameter $1/2$, *i.e.*, (for $x_1 \in \{0, 1, ..., x_t\}$)

$$p_0(x_1|x_t) = \binom{x_t}{x_1} (1/2)^{x_t} \equiv \mathcal{B}i(x_1 \,|\, x_t, 1/2).$$

Since there is no parameter to encode (in this class it is fixed at 1/2), the total description length is simply

$$L_0 = -\log \mathcal{B}i(x_1 \,|\, x_t, 1/2) = -\log \binom{x_t}{x_1} + x_t \log 2. \tag{1}$$

Model Class 1: In this case we let $\lambda_1 \neq \lambda_2$. The corresponding probability of x_1, given x_t, is still binomial but its parameter is no longer 1/2; specifically,

$$p_1(x_1|x_t) = \binom{x_t}{x_1} \rho^{x_1} (1-\rho)^{x_t - x_1} \equiv \mathcal{B}i(x_1 \,|\, x_t, \rho),$$

where $\rho = \lambda_1/(\lambda_1 + \lambda_2)$. In this case, the first step consists in estimating, coding, and transmitting ρ; its ML estimate is $\widehat{\rho} = x_1/x_t$. Because x_t was already transmitted, it suffices to encode and transmit x_1; this requires $\log(x_t+1)$ bits, since $x_1 \in \{0, 1, ..., x_t\}$. Surprisingly, we find that while encoding $\widehat{\rho}$, we have encoded x_1 itself, and so no additional coding is needed. The resulting total description length is simply

$$L_1 = \log(x_t + 1) = -\log \frac{1}{x_t + 1}. \tag{2}$$

The fact that, while encoding the parameter we have also encoded the data itself, is an instance of the *incompleteness* issue [6]. If a subset of code-words of a given code has zero probability of being used, this code is called (maybe somewhat counter-intuitively) *incomplete*. The MDL approach reviewed in Section 2 uses two-part codes: we first encode and send a parameter estimate, then the data itself, coded according to that parameter estimate. However, if we build a code for all possible data out-comes, this code is *incomplete*. In fact, once the receiver has the parameter estimate, it knows that only data out-comes that could have led to this estimate are possible. In our particular case, since the code-word for the parameter estimate coincides with the data itself, we do not need to send the data again at all. This is an extreme case of incompleteness removal.

The resulting model selection criterion is: $\lambda_1 = \lambda_2$, if $L_0 < L_1$, and $\lambda_1 \neq \lambda_2$, otherwise. As mentioned above, the code length associated with the total count x_t is a common constant added to both code lengths, thus irrelevant for model selection purposes. In practice, we would also need an extra bit to indicate which of the two model classes was chosen, which is also irrelevant in terms of model selection.

Finally, we show that the same criterion results from a Bayesian model selection approach [7]. Let x_1 denote a sample of a binomial random variable with probability $\mathcal{B}i(y \,|\, x_t, \rho)$ and consider the problem of deciding between two hypotheses: H_0: $\rho = 1/2$, or H_1: $\rho \neq 1/2$ (totally unknown). Assume that, *a priori*, $p(H_0) = p(H_1) = 1/2$. The models for ρ under the two hypotheses are

$$p(\rho|H_0) = \delta(\rho - 1/2), \qquad p(\rho|H_1) = U(\rho \,|\, 0, 1), \tag{3}$$

where $\delta(\rho - a)$ is a Dirac delta function at a and $U(\rho \,|\, b, c)$ stands for a uniform probability density function between b and c. Naturally, we decide for H_1 if $p(H_1|y) \geq p(H_0|y)$; this condition is equivalent to $p(y|H_1) \geq p(y|H_0)$ because $p(H_0) = p(H_1)$. The so-called *marginal likelihoods* $p(y|H_1)$ and $p(y|H_0)$ are particular cases of the *binomial-Beta* distribution (see [7], pp. 117)

$$p(x_1|H_0) = \int_0^1 p(x_1|\rho)\, p(\rho|H_0)\, d\rho = \mathcal{B}i(x_1 \,|\, x_t, 1/2)$$
$$p(x_1|H_1) = \int_0^1 p(x_1|\rho)\, p(\rho|H_1)\, d\rho = \frac{1}{x_t + 1}.$$

Then, comparing $p(y|H_0)$ versus $p(y|H_1)$ is the same as comparing L_0 versus L_1, as given by (1) and (2).

4. Adaptive Recursive Segmentation

4.1. Splitting a Sequence

Suppose now that we have a length-N sequence of Poisson observations (or counts) $x = \{x_k\}_{k=0}^{N-1}$. Let us consider the following model classes, competing to explain this data. Under Model Class 0, x is a sequence of Poisson samples with common intensity λ. Alternatively, consider $N-1$ other model classes: Model Class i, for $i = 1, .., N-1$. Under Model Class i, $\{x_k\}_{k=0}^{i-1}$ is modeled as a sequence of i Poisson samples of intensity λ_a, while $\{x_k\}_{k=i}^{N-1}$ is a set of Poisson samples of intensity λ_b, with $\lambda_a \neq \lambda_b$. We thus have a total of N candidate classes. If these classes are *a priori* equiprobable, index i is encoded with constant code-length $\log N$, and dropped from any comparisons.

Assume that the total count $s_N = \sum_{k=0}^{N-1} x_k$ is known to the receiver and need not be encoded (as we shall see, this

will be a natural assumption in the complete segmentation method). The description lengths achieved are:

Model Class 0: With a constant intensity model, and given the total s_N, the individual counts follow a multinomial distribution with all parameters equal to $1/N$, *i.e.*,

$$p_0(x_1, ..., x_N | s_N) = \left(\begin{array}{c} s_N \\ x_1, \ldots, x_N \end{array} \right) \left(\frac{1}{N} \right)^{s_N},$$

where the multimonial coefficients are given by

$$\left(\begin{array}{c} s_N \\ x_1, \ldots, x_N \end{array} \right) = \frac{s_N!}{x_1!\, x_2! \cdots x_N!}.$$

In this case, there is no parameter to estimate and the resulting total description length is simply

$$L_0 = -\log \left(\begin{array}{c} s_N \\ x_1, \ldots, x_N \end{array} \right) + s_N \log N. \qquad (4)$$

Note that (1) is a particular case of (4), for $N = 2$.

Model Classes 1, ..., $N-1$: Model class i assumes that $\{x_k\}_{k=0}^{i-1}$ and $\{x_k\}_{k=i}^{N-1}$ are sets of Poisson samples of different intensities, respectively λ_a and λ_b. Given s_N, the individual counts are still multinomially distributed; however, the first i parameters are now equal to $\rho = \lambda_a/(i\lambda_a + (N-i)\lambda_b)$, and the $N-i$ last ones equal to $(1-i\rho)/(N-i) = \lambda_b/(i\lambda_a + (N-i)\lambda_b)$. Notice that with $\lambda_a = \lambda_b$, we get $\rho = 1/N$ and we recover Model Class 0. Then,

$$p_i(x_1, ..., x_N | s_N, \rho) = \left(\begin{array}{c} s_N \\ x_1, \ldots, x_N \end{array} \right) \times \rho^{s_i} \left(\frac{1-i\rho}{N-i} \right)^{s_N - s_i}. \qquad (5)$$

To use this model to encode the data, we compute the ML estimate of ρ, $\widehat{\rho} = s_i/(i\, s_N)$, where $s_i = \sum_{k=0}^{i} x_k$. Since s_N is known to the receiver, all that needs to be encoded is s_i involving a $\log(1 + s_N)$ code-length (since $s_i \in \{0, 1, ..., s_N\}$). After transmitting s_i, the best code for the data has to take into account the fact that the receiver already knows that $\sum_{k=0}^{i-1} x_k = s_i$ and $\sum_{k=i}^{N-1} x_k = s_N - s_i$ (see comment about incompleteness in the previous section). Specifically, each set of counts is itself multinomially distributed, leading to a total code length

$$L_i = \log(1 + s_N) - \log \left(\begin{array}{c} s_i \\ x_1 \ldots x_{i-1} \end{array} \right) + s_i \log i - \log \left(\begin{array}{c} s_N - s_i \\ x_i \ldots x_{N-1} \end{array} \right) + (s_N - s_i) \log(N-i).$$

Notice that this code-length has three components: $\log(1 + s_N)$ bits needed to encode the partial sum s_i, plus the two "-log(multinomial)" terms corresponding to the two segments (compare with (4)).

4.2. Recursive Segmentation of a Sequence

Our progressive/recursive parsing (or transmission) scheme, proceeds as follows. As above, we start by encoding the total count s_N by using, *e.g.*, Elias' technique for arbitrary integers [2]. Then, from the full data set, we compute all the L_i's. If $L_0 < L_{i^*} \equiv \min\{L_1, ..., L_{N-1}\}$, our criterion states that the data is best encoded as a single piece, and the procedure stops. Otherwise, there is one best partition of the data, $\{x_k\}_{k=0}^{i^*-1}$ and $\{x_k\}_{k=i^*}^{N-1}$. We then transmit i^* and s_{i^*} and apply the criterion to the two segments $\{x_k\}_{k=0}^{i^*-1}$ and $\{x_k\}_{k=i^*}^{N-1}$. The receiver can compute the second partial count from s_{i^*} and s_N (which it already has): $s_N - s_{i^*}$; *i.e.*, when the procedure is applied to each of the subsegments, the respective lengths and totals were already transmitted. By recursively repeating this procedure independently to the resulting sub-blocks of data, we obtain a very efficient recursive scheme of refinement. The process stops when no further splits are indicated by the criterion (*i.e.*, we keep splitting until L_0 is selected for each sub-block). The underlying intensity field estimate is piecewise constant, with the segments defined by the obtained parsing and the corresponding intensities as the ML estimates inside each segment. Of course, this is a suboptimal scheme, because at each level we are ignoring that each segment may be further subdivided into even smaller pieces, thus achieving a shorter code length. It is then clear that our scheme can only *under-segment*, never *over-segment* the sequence. An optimal scheme would be computationally extremely heavy. We conclude this section with an illustrative example in Figure 1.

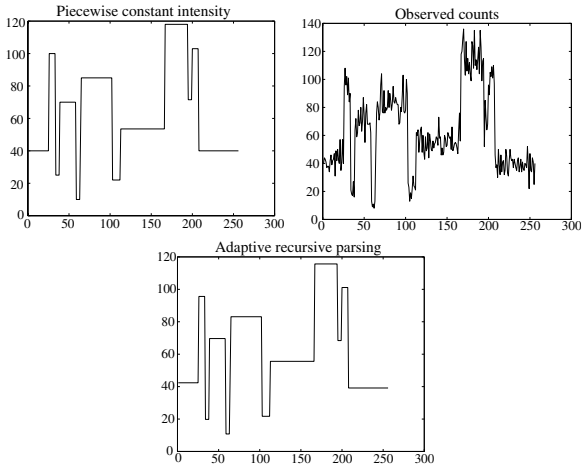
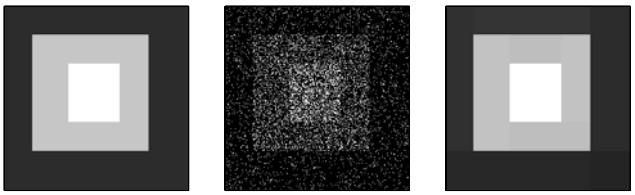


Figure 1. Segmenting a piece-wise constant intensity function from observed counts.

5. Segmenting in Two Dimensions

The 1D strategy described above can be extended to 2D. The main difference is that in 2D we have more freedom in how we split the data. To maintain a manageable algorithm, we restrict the splitting to rectangular tesselations. In our recursive scheme, the MDL criterion is applied to rectangular blocks to select one of the following possibilities: **(a)** no splitting (the rectangle is considered homogeneous); **(b)** the rectangle is split into four (or two[1]) sub-rectangles defined by a common vertex (the best possible such splitting is chosen). As in the 1D case, the code lengths for these options are derived from the multinomial probabilities. As in 1D, we start by applying the criterion to the full image. Every time one rectangular block (the image itself, to start) is split (into 2 or 4 sub-rectangles), the criterion is again applied to the resulting sub-regions. The parsing process stops when no further splits are indicated by the MDL criterion. The final estimate of the intensity field is piece-wise flat, with the rectangular regions defined by the parsing; the corresponding intensities are the ML estimates based on the data inside each region. Figures 2 shows an example based on a piecewise-constant intensity image. The sequence of segmentations obtained along the recursive scheme is shown in Fig. 3. Finally, Fig. 4 shows an example on a natural image.



(a) (b) (c)

Figure 2. (a) Piecewise-constant intensity (intensities 0.05, 0.2, and 0.4). (b) Observed photon events. (c) Intensity field parsing.

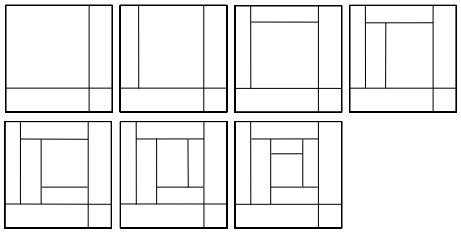
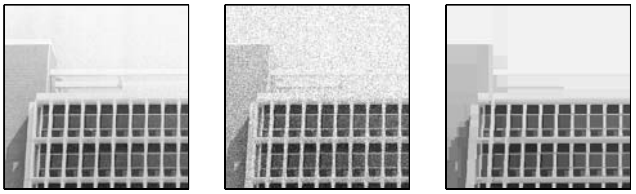


Figure 3. Segmentation sequence for the data shown in Fig. 2.

[1] Notice that with the vertex at one of the edges of the rectangle, we can also perform a horizontal or vertical split into two sub-rectangles.



(a) (b) (c)

Figure 4. Parsing a natural image. (a) Intensity. (b) Counts, (normalized) MSE = 1.00. (c) Adaptive recursive estimate, MSE = 0.54.

6. Conclusions and Future Work

Our MDL parsing scheme is a fully unsupervised alternative to the Bayesian methods of [3, 4]. We have shown that our MDL criterion is, in fact, a special case of a Bayesian approach. However, MDL has no free parameters; it is fully data-driven. Due to the predictive (coarse-to-fine) nature of the method, we were able to write exact (non-asymptotic) expressions for the parameter code-lengths.

The 2D method described here is based on rectangular tesselations, thus showing a clear preference for vertical and horizontal edges. We could use more general refinement schemes based on polygonal region splitting. For example, in the recursive scheme, at each step we could search for the optimal (in MDL sense) line(s) partitioning a given polygon into to smaller polygons.

References

[1] D. Snyder and M. Miller, *Random Point Processes in Time and Space*. New York: Springer Verlag, 1991.

[2] J. Rissanen, *Stochastic Complexity in Stastistical Inquiry*. Singapore: World Scientific, 1989.

[3] K. Timmermann and R. Nowak, "Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging," *IEEE Trans. on Info. Theory*, vol. 45, pp. 846–862, 1999.

[4] E. Kolaczyk, "Bayesian multi-scale models for Poisson processes," *J. Amer. Statist. Assoc.*, vol. 94, pp. 920-933, 1999.

[5] J. Scargle, "Studies in astronomical time series analysis. Bayesian blocks, a new method to analyze structure in photon counting data," *Astrophysical Jour.*, vol. 504, pp. 405-418, 1998.

[6] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. on Information Theory*, vol. 42, pp. 40-47, 1996.

[7] J. Bernardo and A. Smith, *Bayesian Theory*. Chichester, UK: J. Wiley & Sons, 1994.