Multiresolution Intensity Estimation of Piecewise Linear Poisson Processes

R. M. Willett

Abstract

Given observations of a one-dimensional piecewise linear, length-M Poisson intensity function, our goal is to estimate both the partition points and the parameters of each segment. In order to determine where the breaks lie, we develop a maximum penalized likelihood estimator based on information-theoretic complexity penalization. We construct a probabilistic model of the observations within a multiscale framework, and use this framework to devise a computationally ef£cient optimization algorithm, based on a tree-pruning approach, to compute the MPLE.

I. INTRODUCTION

Poisson processes are a useful statistical model for a wide variety of counting problems in a number of different £elds. Counting the photons being emitted from a radioactive source has applications in nuclear medicine, geochronology, and nuclear physics. Seismologists use Poisson processes to model earthquake occurrences, and failure analysts model failure rates. [**?**] Recently work by the astronomical community on Gamma Ray Bursts has also attracted the attention of Poisson statisticians. [**?**] In these and other applications, intensity estimation can be a daunting but necessary task for successful data analysis.

Recent work by Nowak and Kolaczyk provides a detailed look at multiscale Poisson intensity estimation. [**?**] They describe how a maximum likelihood intensity estimate will simply yield the original data points, and while accurate, this non-parsimonious description of the intensity would provide little insight into the true nature of the underlying process. Their solution to this problem is to work instead with maximum *a posteriori* (MAP) estimates. Nowak's work with Figueiredo [**?**] alternatively used complexity regularized MLEs via the minimum description length principle to obtain piecewise constant intensity estimates.

This paper extends the work of Nowak and Kolaczyk [**?**] to calculate multiscale complexity regularized intensity estimates of piecewise linear Poisson processes.

II. MULTISCALE FRAMEWORK

Let us consider independent observations of a one-dimensional piecewise linear, length-M Poisson intensity function λ_m such that

$$x_m \sim \mathcal{P}(\lambda_m),\ m = 0, \ldots, M-1 \tag{1}$$

where $\mathcal{P}(\lambda_m)$ denotes a Poisson distribution with intensity parameter λ_m. Our mission is to estimate $\boldsymbol{\lambda} = \{\lambda_m\}_{m=0}^{M-1}$ from the observed data $\mathbf{x} = \{x_m\}_{m=0}^{M-1}$. By performing a multiresolution analysis of the data, we can exploit signal structure at various scales and temporal locations. This technique is well-suited to our task of estimating a piecewise linear intensity function.

Through a recursive summation of the data, we de£ne the following multiscale framework:

$$\begin{aligned} x_{J,m} &\equiv x_m,\ m = 0, \ldots, 2^J - 1, \\ x_{j,m} &= x_{j+1,2m} + x_{j+1,2m+1},\ m = 0, \ldots, 2^j - 1,\ 0 \le j \le J-1 \end{aligned} \tag{2}$$

where the index j refers to the resolution of the analysis and the index m refers to the temporal location. Recall that the Poisson distribution reproduces itself under unweighted summation, with the distribution of the sum having an intensity equal to the sum of the intensities of the distributions being summed. (See Appendix A for proof.)

Work done in collaboration with Robert Nowak at Rice University, Dept. of Electrical and Computer Engineering, Houston, TX 77251, USA; willett@rice.edu, nowak@rice.edu

We thus note that each $x_{j,m} \sim \mathcal{P}(\lambda_{j,m})$, where $\lambda_{j,m}$ is de£ned by the analogous multiscale framework for the intensity function $\boldsymbol{\lambda}$:

$$\begin{aligned} \lambda_{J,m} &\equiv \lambda_m,\ m = 0, \ldots, 2^J - 1, \\ \lambda_{j,m} &= \lambda_{j+1,2m} + \lambda_{j+1,2m+1},\ m = 0, \ldots, 2^j - 1,\ 0 \le j \le J-1. \end{aligned} \quad (3)$$

This framework is equivalent to a representation of the data with a Haar multiresolution analysis.

In order to calculate the MPLE, we will need to maximize a penalized function of $p(\boldsymbol{x}|\boldsymbol{\lambda})$. Let us £rst examine $p(\boldsymbol{x}|\boldsymbol{\lambda})$ from within our multiscale framework. From [**?**] we have that

$$\begin{aligned} p(\boldsymbol{x}|\boldsymbol{\lambda}) &= \prod_{m=0}^{M-1} p(x_n|\lambda_n) \\ &= p(x_{0,0}|\lambda_{0,0}) \prod_{j=0}^{J-1} \prod_{m=0}^{2^j-1} p(x_{j+1,2m}|x_{j,m}, \rho_{j,m}) \\ &= \mathcal{P}(x_{0,0}|\lambda_{0,0}) \prod_{j=0}^{J-1} \prod_{m=0}^{2^j-1} \mathcal{B}(x_{j+1,2m}|x_{j,m}, \rho_{j,m}) \end{aligned}$$

where $\rho_{j,m} \equiv \frac{\lambda_{j+1,2m}}{\lambda_{j,m}}$, $\mathcal{P}(x|\theta) = \frac{e^{-\theta}\theta^x}{x!}$ denotes the Poisson distribution with parameter θ, and $\mathcal{B}(x|n,\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x}$ denotes the binomial distribution with parameters n and θ. We verify that the probability mass function of a Poisson random variable conditioned on its sum with another Poisson random variable follows a binomial distribution in Appendix B.

This can be extended to form the following statement: Let $X_1, X_2, \ldots X_N$ be N independent Poisson random variables with intensities $\lambda_1, \lambda_2, \ldots \lambda_N$, respectively, and let $Z \equiv \sum_{k=1}^N X_k$ and $\lambda_Z \equiv \sum_{k=1}^N \lambda_k$. Then

$$X_1, X_2, \ldots, X_N \mid Z \sim \mathcal{M}\left(x_1, x_2, \ldots, x_N \;\middle|\; z, \frac{\lambda_1}{\lambda_Z}, \frac{\lambda_2}{\lambda_Z}, \ldots \frac{\lambda_N}{\lambda_Z}\right)$$

where $\mathcal{M}(\mathbf{y}|n,\boldsymbol{\mu}) = \frac{n!}{\prod_{i=0}^{N-1} y_i!}\prod_{i=0}^{N-1}\mu_i^{y_i}$ denotes the multinomial distribution with parameters n and $\boldsymbol{\mu}$, with $n = \sum_{i=0}^{N-1} y_i$ and $\sum_{i=0}^{N-1}\mu_i = 1$. This relationship is veri£ed in Appendix C.

The parent-child relationship in (2) and (3) is intuitively captured by a binary tree, in which the leaf nodes represent the observed data and the underlying intensities, and the internal nodes represent recursive summations of the data and intensities. The edges between nodes represent "splitting factors" and determine the relative intensities of the two children. Figure 1 displays the full binary tree for a length-16 intensity function.

Fig. 1. Full Multiscale Tree Structure

Our goal is to estimate the nature of the piecewise linear structure of the underlying intensity function. We accomplish this by using MPL estimation to prune the tree until each leaf node represents either a homogeneous or linear segment. Consider the special case in which the intensity function is piecewise constant. Then we can prune the homogeneous intensity child nodes and store only the terminal nodes and the homogeneous intensity level for that node, as in Figure 2. For a piecewise linear intensity function, we could prune the tree in a similar manner, storing the intensity level for

Fig. 2. Pruned Tree for Piecewise Constant Intensities

homogeneous regions and storing both the average intensity level and the intensity variation (or slope), as if Figure 3.

III. MLE FOR POISSON PROCESSES

Before approaching our piecewise linear intensity problem, let us £rst consider the task of £nding a maximum likelihood estimate of the intensities of a linear Poisson process. The technique described in this section can be applied

Fig. 3. Pruned Tree for Piecewise Linear Intensities

to the multinomial distribution (which, as stated above, is applicable in our multiscale framework) with only minor modi£cations, as described in Section IV.

Unser and Eden [**?**] describe a maximum likelihood estimation of linear signal parameters for Poisson processes. They derive the maximum likelihood estimate of the mean vector of a Poisson process subject to the following linear constraint:

$$\mathbb{E}(\mathbf{x}) = \boldsymbol{\lambda} = \mathbf{T} \cdot \mathbf{a}, \tag{4}$$

where $\lambda_k = \sum_{i=1}^{M} a_i t_{k,i}$. Here $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_N]^\top$ is the mean vector of the measurement vector $\mathbf{x} = [x_1, \ldots, x_N]^\top$, $\mathbf{T} = [\mathbf{t_1}, \ldots, \mathbf{t_M}]$ is a known $N \times M$ matrix, and $\mathbf{a} = [a_1, \ldots, a_M]^\top$ is the vector of expansion coef£cients to be estimated.

They derive the log likelihood function

$$\begin{aligned} L(\mathbf{x} \mid \boldsymbol{\lambda}) &= \log\left(\prod_{k=1}^{N} \frac{e^{-\lambda_k} \cdot \lambda_k^{x_k}}{x_k!}\right) \\ &= \sum_{k=1}^{N} \left(-\lambda_k + x_k \log(\lambda_k) - \log(x_k!)\right). \end{aligned} \tag{5}$$

Because no closed-form solution to this problem exists, a numerical solution must be calculated using a gradient or steepest descent algorithm. Unser and Eden developed this algorithm, but did not investigate its convergence. We £ll that gap here. In order to demonstrate that their algorithm will correctly identify a unique global maxima, we prove that the likelihood function is strictly concave.

Note that for this work, we wish to consider the vector of intensities $\boldsymbol{\lambda}$ as varying linearly with the index, *i.e.* $u_k = a_1 + a_2(k - \frac{N-1}{2})$, so that

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ -\frac{N}{2} + \frac{1}{2} & -\frac{N}{2} + \frac{3}{2} & -\frac{N}{2} + \frac{5}{2} & \cdots & \frac{N}{2} - \frac{1}{2} \end{bmatrix}^\top$$

Note that the index k is shifted in this formulation. This is to ensure that a_0 is the average intensity of the intensity vector. In this manner, we reduce a two-dimensional search for the optimal parameter vector to a one-dimensional search for the optimal slope of the intensities. This formulation implies that $M = 2$, $t_{k,1} = 1$, and $t_{k,2} = k - \frac{N+1}{2}$. For notational convenience, let $k' \equiv k - \frac{N-1}{2}$. Thus

$$\begin{aligned} L(\mathbf{x} \mid \boldsymbol{\lambda}) &= L(\mathbf{x} \mid \mathbf{a}) \\ &= \sum_{k=1}^{N} \left(-a_1 - a_2 k' + x_k \log(a_1 + a_2 k') - \log(x_k!)\right). \end{aligned} \tag{6}$$

Theorem 1: $L(\mathbf{x} \mid \mathbf{a})$ is strictly concave.

Proof of Theorem 1: In order to determine whether $L(\mathbf{x} \mid \mathbf{a})$ is strictly concave, we may determine whether $-L(\mathbf{x} \mid \mathbf{a})$ is strictly convex. From [**?**], we know that $-L(\mathbf{x} \mid \mathbf{a})$ is strictly convex if and only if its Hessian matrix is positive de£-nite. The Hessian matrix of $-L(\mathbf{x} \mid \mathbf{a})$ is

$$\mathbf{Q} \equiv \begin{bmatrix} \frac{\partial^2(-L(\mathbf{x}|\mathbf{a}))}{\partial a_1^2} & \frac{\partial^2(-L(\mathbf{x}|\mathbf{a}))}{\partial a_1 \partial a_2} \\ \frac{\partial^2(-L(\mathbf{x}|\mathbf{a}))}{\partial a_1 \partial a_2} & \frac{\partial^2(-L(\mathbf{x}|\mathbf{a}))}{\partial a_2^2} \end{bmatrix}.$$

Let

$$c_k \equiv \frac{x_k}{(a_1 + a_2 k')^2}.$$

Then

$$\mathbf{Q} = \left[\begin{array}{cc} \sum_{k=1}^{N} c_k & \sum_{k=1}^{N} c_k k' \\ \sum_{k=1}^{N} c_k k' & \sum_{k=1}^{N} c_k k'^2 \end{array} \right]. \tag{7}$$

For notational simplicity, let

$$\begin{aligned} w &\equiv \sum_{k=1}^{N} c_k \\ y &\equiv \sum_{k=1}^{N} c_k k' \\ z &\equiv \sum_{k=1}^{N} c_k k'^2 \end{aligned}$$

so that

$$\mathbf{Q} = \left[\begin{array}{cc} w & y \\ y & z \end{array} \right].$$

Note that for $k' = 1, \cdots, N$, $x_k \geq 0$ because it is an observation of a Poisson random variable, and $(a_1 + a_2 k')^2 \geq 0$ because a_1, a_2, and k' are all real numbers. Let us assume, without loss of generality, that $\mathbf{x}$ is not a zero vector, and thus $\mathbf{Q}$ is not a zero matrix. Thus w, y, and z are all real numbers, and w and z are positive.

To show that $\mathbf{Q}$ is positive de£nite, we must show that $\mathbf{y}^\top \mathbf{Q} \mathbf{y} > 0$ for all $\mathbf{y} \in \mathbb{R}^2$ such that $\mathbf{y} \neq \mathbf{0}$. Let $\mathbf{v}$ be an eigenvector for $\mathbf{Q}$, such that $\mathbf{Q}\mathbf{v} = \lambda \mathbf{v}$. If $\mathbf{Q}$ is positive de£nite, then $\mathbf{v}^\top \mathbf{Q} \mathbf{v} = \lambda \mathbf{v}^\top \mathbf{v} > 0$. Since $\mathbf{v}^\top \mathbf{v} > 0$, λ must also be positive. Hence, in order for $\mathbf{Q}$ to be positive de£nite, its eigenvalues must be positive.

The characteristic function of $\mathbf{Q}$ is

$$\begin{aligned} p(\lambda) &= (w - \lambda)(z - \lambda) - y^2 \\ &= \lambda^2 + \lambda(-w - z) + wz - y^2. \end{aligned} \tag{8}$$

To £nd the eigenvalues, we set the characteristic function equal to zero and solve for λ to get

$$\begin{aligned} \lambda &= \frac{z + w \pm \sqrt{(z + w)^2 - 4wz + 4y^2}}{2} \\ &= \frac{z + w \pm \sqrt{(z - w)^2 + 4y^2}}{2}. \end{aligned} \tag{9}$$

All terms within the square root are positive, resulting in a positive real square root. Thus $\lambda > 0$ if

$$z + w > \sqrt{(z - w)^2 + 4y^2}$$

which would imply $(z + w)^2 > (z - w)^2 + 4y^2$, which in turn implies $zw > y^2$. Returning to earlier notation, we express this condition as

$$\sum_{k=1}^{N} c_k k'^2 \cdot \sum_{k=1}^{N} c_k > \left(\sum_{k=1}^{N} c_k k' \right)^2.$$

This is true with strict inequality by the Cauchy-Schwarz inequality. We thus conclude that $\lambda > 0$, and therefore $\mathbf{Q}$ is positive de£nite, and so $L(\mathbf{x} \mid \mathbf{a})$ is strictly concave.

■

IV. MLE FOR MULTINOMIAL PROCESSES

This section discusses the calculation of $\hat{\mathbf{a}}_{\mathbf{ML}}$ as described in the algorithm of Section V. Consider the problem of estimating $\boldsymbol{\mu}$ from observations $\mathbf{y}$ following the distribution

$$p(\boldsymbol{y}|\boldsymbol{\mu}) = \mathcal{M}\left(\mathbf{y} \mid n, \boldsymbol{\mu}\right) \tag{10}$$

where $\mathcal{M}(\mathbf{y}|n,\boldsymbol{\mu}) = \frac{n!}{\prod_{i=0}^{N-1} y_i!}\prod_{i=0}^{N-1}\mu_i^{y_i}$ denotes the multinomial distribution with parameters n and $\boldsymbol{\mu}$, with $\sum_{i=0}^{N-1} y_i = n$ and $\sum_{i=0}^{N-1}\mu_i = 1$. We consider the additional constraints that

$$\boldsymbol{\mu} = \mathbf{T}\cdot\mathbf{a} \tag{11}$$

(as in Equation 4) and each $\mu \geq 0$.

The log-likelihood function is thus

$$\begin{aligned} L(\mathbf{y} \mid \boldsymbol{\mu}) &= \log\left(\frac{n!}{\prod_{i=0}^{N-1} y_i!}\right) + \sum_{i=0}^{N-1} y_i \log\mu_i \\ &= \log(n!) + \sum_{i=0}^{N-1}\left(y_i\log\mu_i - \log(y_i!)\right) \\ L(\mathbf{y} \mid \mathbf{a}) &= \log(n!) + \sum_{i=0}^{N-1}\left(y_i\log(a_0 + i\cdot a_1) - \log(y_i!)\right). \end{aligned}$$

Setting the derivative to zero we obtain

$$\begin{aligned} \frac{\partial L(\mathbf{y}\mid\mathbf{a})}{\partial \boldsymbol{a}} &= \mathbf{T}^{\top}\frac{\partial L(\mathbf{y}\mid\boldsymbol{\mu})}{\partial\boldsymbol{\mu}} \\ &= \mathbf{T}^{\top}\cdot\left[\left(\frac{y_1}{\mu_1}\right),\left(\frac{y_2}{\mu_2}\right),\ldots,\left(\frac{y_N}{\mu_N}\right)\right]^{\top} \\ &= \mathbf{0} \end{aligned}$$

or

$$\begin{bmatrix} \sum_{i=0}^{N-1}\frac{y_i}{a_0+i\cdot a_1} \\ \sum_{i=0}^{N-1}\frac{y_i\cdot i}{a_0+i\cdot a_1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Because no closed-form solution to this problem exists, again a numerical solution must be calculated using a gradient or steepest descent algorithm. In order to demonstrate that such an algorithm will correctly identify a unique global maxima, we prove that the likelihood function is strictly concave. Noting that the Hessian matrix of $-L(\mathbf{y}\mid\boldsymbol{\mu})$ is equivalent to that of $-L(\mathbf{x}\mid\boldsymbol{\lambda})$ in Section III, we conclude that the likelihood function is strictly concave.

V. Multiresolution MPLE Algorithm

We wish to estimate the intensity $\boldsymbol{\lambda}$ of a Poisson process $\mathbf{x}$ subject to the following criterion:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda}}\left\{-\log\left(p\left(\mathbf{x}\mid\boldsymbol{\lambda}\right)\right) + penalty(M)\right\} \tag{12}$$

where M is proportional to the complexity of the model. (The penalization will be discussed in detail below.)

In order to perform this estimation, we consider each dyadic segments of the observation vector and perform a triple hypothesis test. Our hypotheses are as follows:

- $\mathbf{H_0}$: λ's in this segment are constant with respect to time or location.
- $\mathbf{H_1}$: λ's in this segment vary linearly with time or location.
- $\mathbf{H_2}$: λ's in this segment are not homogeneous and do not vary linearly with time or location, but are otherwise unknown; *i.e.* the segment contains one or more partition points.

The algorithm recursively extracts the dyadic partition points according to our maximized penalized likelihood criterion de£ned in (12). As described here, our algorithm is reminiscent of dynamic programming, Coifman and Wickerhauser's "Best-Ortho-Basis" algorithm [**?**], or the CART algorithm [**?**]. It begins one level above the leaf nodes in our dyadic tree and traverses upwards, performing a tree-pruning operation at each stage. For each node at a particular level, the vector $\mathbf{y}$ contains the leaf nodes descending from the current node. We then calculate the penalized likelihoods $p(\mathbf{y}|H_0)$, $p(\mathbf{y}|H_1)$, and $p(\mathbf{y}|H_2)$, and the penalty associated with each likelihood: $penalty(1)$ for $\mathbf{H_0}$, $penalty(1)$ for

$\mathbf{H_1}$, and $penalty(0)$ for $\mathbf{H_2}$. We store the maximum penalized likelihood of the current node in our "tree", the hypothesis yielding that penalized likelihood, and either the homogeneous level or the average intensity and linear slope if the homogeneous or linear model, respectively, is most likely. Upon completion of this loop, we start at the root of the tree and perform a depth £rst search until we reach terminal nodes (leaf nodes or nodes representing homogeneous or linear regions), recording the intensity estimate at each time.

Before delving into the algorithm details, let us consider the simple example of a length-8 piecewise linear intensity function as shown in Figure 4. We would £rst compute the likelihoods that $\lambda_{3,0}$ and $\lambda_{3,1}$ are inhomogeneous and non-linear, homogeneous (constant), or varying linearly, and then repeat for $\lambda_{3,2}$ and $\lambda_{3,3}$,$\lambda_{3,4}$ and $\lambda_{3,5}$, and $\lambda_{3,6}$ and $\lambda_{3,7}$. On the next level we could compute the likelihood that $\lambda_{2,0} = \lambda_{2,1}$, and multiply it by the likelihoods (computed on the previous level) that $\lambda_{3,0} = \lambda_{3,1}$, and $\lambda_{3,2} = \lambda_{3,3}$. We then assume that $\lambda_{3,0}$ through $\lambda_{3,3}$ vary linearly with time or position, and calculate the maximum likelihood estimate of the slope. We calculate the likelihood that $\lambda_{3,0}$ through $\lambda_{3,3}$ lie along the line speci£ed by our estimate of the slope and the mean value of the data. Next we calculate the likelihood that $\lambda_{2,0}$ and $\lambda_{2,1}$ are unrelated, and multiply it by the maximum of the likelihoods calculated on the previous level. In this example, we would note that it is most likely that the nodes are homogeneous. These steps are repeated for $\lambda_{3,4}$ through $\lambda_{3,7}$, and we note that it is most likely that the nodes vary linearly. Finally, we examine likelihoods for $\lambda_{1,0}$ and $\lambda_{1,1}$ and see that the two subtrees are most likely not connected.

Fig. 4. Algorithm Example

- `Initialize:` $j = J-1$; $L_{min}(n_{J,:}) = L_1(n_{J,:}) = 0$.
- `Loop:` for each node $n_{j,m}$ at level j
- `Calculate:`

$$\begin{aligned}
\hat{\rho}_0 &= \frac{1}{2} \\
\hat{\mathbf{a}}_{\mathbf{ML}} &= \arg\max_{\mathbf{a}} \; p\left(\left[x_{J,m\cdot 2^{J-j}} \ldots x_{J,(m+1)\cdot 2^{J-j}-1}\right]^\top \mid x_{j,m}, \mathbf{T}\cdot\mathbf{a}\right) \\
\hat{\boldsymbol{\rho}}_1 &= \frac{\mathbf{T}\cdot\hat{\mathbf{a}}_{\mathbf{ML}}}{\lambda_{j,m}}; \quad \mathbf{T} = \begin{bmatrix} 1\,1\,1\cdots & 1 \\ 1\,2\,3\cdots & 2^{J-j} \end{bmatrix}^\top \quad (\textit{see section III}) \\
\hat{\rho}_2 &= \frac{x_{j+1,2m}}{x_{j,m}}
\end{aligned}$$

- `Evaluate:`

$$\begin{aligned}
L_0(n_{j,m}) &= -\log\left(p\left(x_{j+1,2m} \mid x_{j,m}, \hat{\rho}_0, H_0\right)\right) + L_0(n_{j+1,2m}) + L_0(n_{j+1,2m+1}) + penalty(1) \\
L_1(n_{j,m}) &= -\log\left(p\left(\left[x_{J,m\cdot 2^{J-j}} \ldots x_{J,(m+1)\cdot 2^{J-j}-1}\right]^\top \mid x_{j,m}, \hat{\boldsymbol{\rho}}_1, H_1\right)\right) + penalty(1) \\
L_2(n_{j,m}) &= -\log\left(p\left(x_{j+1,2m} \mid x_{j,m}, \hat{\rho}_2, H_2\right)\right) + L_{min}(n_{j+1,2m}) + L_{min}(n_{j+1,2m+1}) + penalty(0)
\end{aligned}$$

- `Save:`

$$L_{min}(n_{j,m}) = \min_{i\in\{0,1,2\}} L_i(n_{j,m})$$

and

$$model_{min}(n_{j,m}) = \arg\min_{i\in\{0,1,2\}} L_i(n_{j,m})$$

- `j=j-1`

• `Goto Loop`.
• `Estimate` Perform a depth £rst search for terminal nodes. When a terminal node is found, record the MPLE for each λ originally descending from the current node.

VI. PENALTIES

The likelihood penalization allows the estimation algorithm to perform model selection, as it penalizes different intensity models according to their complexity. There are a number of valid techniques for assigning penalties. Our technique was to assign each constant region a single penalty and each linear region a double penalty, but not assign any penalties for splitting.

The task of choosing a penalization scheme offers a variety of options and no clear best choice. We overlook this source of ambiguity for the present, however. Our goal is, in future work, to derive performance bounds for the estimator for a given penalization scheme. That is, for a given penalization scheme and the desired value of some performance metric, we hope to be able to set the penalization factor to ensure that our estimate lies within the said performance metric. (The penalization factor is the number by which each penalty is multiplied in a penalization scheme.)

VII. EXTENSIONS

A. *Piecewise Polynomial Intensity*

It may be of interest to the reader to note that the ideas developed thus far in the paper may easily be extended to piecewise-polynomial intensity functions. Unser's work remains applicable, and the likelihood function remains concave. While the computation time necessarily increases, polynomial £tting is clearly a better choice than line £tting for a number of signals.

B. *Piecewise Exponential Intensity*

The extension of multiresolution intensity estimation to piecewise exponential intensities is somewhat more complex. Consider, for example, the model

$$\lambda(k) = \exp\{a_1 k + a_0\},$$

and set

$$\boldsymbol{\eta} \equiv \log(\boldsymbol{\lambda})$$

so that, in the same notation as above, $\boldsymbol{\eta} = \mathbf{T} \cdot \mathbf{a}$. Then

$$\begin{aligned} L(\mathbf{x} \mid \boldsymbol{\eta}) &= L(\mathbf{x} \mid \mathbf{a}) \\ &= \sum_{k=1}^{N} \left(-\exp\{\mathbf{T} \cdot \mathbf{a}\} + x_k \mathbf{T} \cdot \mathbf{a} - \log(x_k!)\right), \end{aligned}$$

which leads to the MLE formulation

$$\begin{aligned} \frac{\partial L(\mathbf{x} \mid \mathbf{a})}{\partial \boldsymbol{a}} &= \mathbf{T}^\top \left[\mathbf{x} - \exp\{\mathbf{T} \cdot \mathbf{a}\}\right] \\ &= \mathbf{0}. \end{aligned}$$

Note further that the likelihood function is concave. The proof closely follows that for the linear case; now we would have

$$c_k \equiv \exp\{a_1 k + a_0\}$$

which by inspection is always positive, and the proof then follows.

C. *Non-Dyadic Partition Points*

All the algorithm details above utilize the binary tree representation of the data and intensities, and hence can only identify breakpoints that lie on dyadic partition points. Hence if we have a breakpoint which lies on a non-dyadic partition point, the above algorithm would continue to halve partitions of the data until it reached a breakpoint. (See Figure 5 for illustration.) While this still yields a quite reasonable intensity estimate, this technique is clearly unsuitable

Fig. 5. Effect of Non-dyadic Breakpoint on Dyadic Algorithm

for someone primarily interested in the change points. At this point we refer to the work done by Nowak and Kolaczyk in [?], and their discussion of libraries of recursive partitions which are not restricted to dyadic partitions. They outline an $O(n^3)$ algorithm for likelihood parameter estimation with non-dyadic partitions. A variation of this was implemented, and the results are discussed below. (The reader should note that there are many approaches to non-dyadic estimation, neither discussed nor implemented here, several of which are highlighted in the Nowak and Kolaczyk paper.)

VIII. Results and Applications

A. Simulated Data

The above algorithms were tested on MATLAB generated data in order to assess their performance. MSE values displayed on the plots compare the estimate with the true intensity function, not the data. (The MSE is not a useful tool in judging how well the estimate £ts the data, as the MMSE estimator would simply return the data points as the estimate, a model we strive to avoid by using penalized likelihoods.)

Figure 6 demonstrates the algorithm's ability to estimate piecewise constant intensity functions quite accurately. The dashed vertical lines show the dyadic partition points identi£ed by the algorithm. Figure 7 show the decision made by the algorithm for each dyadic segment on each level. This is a graphical representation of the "tree" which is pruned to form the estimate. The blue line at the top indicates that all the data cannot be £tted on one line or constant, so the algorithm proceeds to the next level. Here it sees that the £rst half of the data can be represented by one linear segment, while the second half of the data would be better £t with two separate estimates. Thus the algorithm proceeds to the next level for the second half of the data only. Here it sees that the third and fourth quarters of the data can each be represented by a (different) constant. Since the algorithm has now parsed every branch of the tree until it reached a terminal node, it halts and returns the estimate seen in Figure 6.

Fig. 6. A Simple Piecewise Linear Example

Fig. 7. Model Selection at Each Scale

Figure 8 demonstrates the ability of the piecewise polynomial algorithm. For the sake of comparison, we present Figure 9, in which the data (the same data as in Figure 8) was generated from a polynomial intensity function, and the estimate of the intensity using the piecewise linear model. MSE in this case is somewhat higher than that achieved using the piecewise linear model, but one may observe that the linear estimate is not too visually assaulting.

Fig. 8. A Simple Piecewise Polynomial Example (Degree = 3)

Next observe Figure 11, in which the piecewise exponential algorithm was used to estimate data observed from a piecewise exponential true intensity. For the sake of comparison, observe Figure 12, in which the exponential intensity was estimated using the piecewise polynomial algorithm. Again, while this estimate is not quite as accurate as the one achieved using exponential algorithm, it still provides a very reasonable estimate of the true intensity. This result is encouraging, for we frequently would not know whether the true underlying intensity is polynomial, exponential, or some combination when analyzing data from the real world.

Finally, the results of the non-dyadic partition algorithm are displayed in Figure 10. In this implementation, the signal was divided into four dyadic regions, and the non-dyadic partitioning algorithm was applied to each quarter. This reduced the the complexity from $O(n^3)$ to $O(4 * (\frac{n}{4})^3) = O(n^3/16)$. Note that the large break in the data at the

Fig. 9. Polynomial Signal Estimated with Linear Model

$2/3$ point is much better represented by this partitioning, although some undesirable partition points were added in the far-right region of the signal.

Fig. 10. Non-Dyadic Partitioning

I shall now discuss two such "real world" applications for this estimation technique.

Fig. 11. A Simple Exponential Example

Fig. 12. Exponential Signal Estimated with Polynomial Model

B. Network Traf£c Data

Another area of active research is the analysis of network traf£c. Our piecewise linear estimation technique we performed on trace 20000125-143640-0 of the Auckland-II data set of long GPS-synchronized IP header traces, which contained the number of packets arriving in 2 ms intervals. [**?**]

Figures 13 and 14 show the application of the piecewise linear and polynomial algorithms, respectively, to a section of the aforementioned network data, binned to re¤ect the number of packets received in 100 ms intervals and truncated so that we can focus on just a segment of time.

For contrast, we offer Figure 15, in which linear and constant regions received the same penalty, which results in a linear £t always being the most likely for any data region. Note while this estimation appears to be a better representation of the data, it is also less parsimonious than the result obtained by giving linear regions a higher penalty than constant regions.

C. GRB Data

One of the most fascinating problems in astrophysics today is the nature and origin of Gamma Ray Bursts, quick, extremely intense (stronger even than those from the sun) bursts of gamma rays commonly associated with star formation and supernovae. [**?**] Kolaczyk has done preliminary work in this £eld using Haar wavelets, an approach similar to the one described in this paper with the restriction that the intensity be piecewise constant. [**?**]

Figures 16 and 17 show the application of the piecewise linear and polynomial algorithms, respectively, to Gamma Ray Burst data detected by the Burst and Transient Source Experiment (BATSE) trigger 845, separated into 256 bins.

IX. Future Plans

As mentioned in Section VI, one immediate goal is to derive a relationship between the penalization factor and estimation error. We would also like to analyze the rate at which our estimator converges to the true intensity (assuming that the true intensity is piecewise linear) as the number of samples, N, increases.

An obvious extension of this work is to develop the algorithm for two-dimensional signals. While this current work is a solid foundation upon which to build, image processing presents several challenges not applicable to the problem in one dimension. A linear region might vary in the x or y direction, or at any angle between the axes. We anticipate that the performance bounds will need to be carefully reviewed and adapted. Finally, we hope to incorporate Donoho's wedgelets [**?**] into the 2D estimation in order to achieve faster convergence.

Fig. 13. Network Traf£c Linear Estimate

Fig. 14. Network Traf£c Polynomial Estimate

APPENDICES

I. REPRODUCIBILITY OF POISSON DISTRIBUTION

Theorem 2: Let X and Y be two independent Poisson random variables, with intensities λ_X and λ_Y, respectively. If $Z = X + Y$, then Z is also a Poisson random variable, and its intensity is $\lambda_Z = \lambda_X + \lambda_Y$.

Proof of Theorem 2: Consider the characteristic function of Z:

$$\begin{aligned}
\Phi_Z(\omega) &= \mathbb{E}[e^{\jmath\omega Z}] \\
&= \mathbb{E}[e^{\jmath\omega(X+Y)}] \\
&= \mathbb{E}[e^{\jmath\omega X}] \cdot \mathbb{E}[e^{\jmath\omega Y}] \\
&= \sum_{x=0}^{\infty} e^{\jmath\omega x} \frac{e^{-\lambda_X}\lambda_X^x}{x!} \cdot \mathbb{E}[e^{\jmath\omega Y}] \\
&= \sum_{x=0}^{\infty} (e^{\jmath\omega}\lambda_X)^x e^{-\lambda_X} e^{\lambda_X e^{\jmath\omega}} e^{-\lambda_X e^{\jmath\omega}} \frac{1}{x!} \cdot \mathbb{E}[e^{\jmath\omega Y}] \\
&= e^{\lambda_X(e^{\jmath\omega}-1)} \cdot e^{\lambda_Y(e^{\jmath\omega}-1)} \\
&= e^{(\lambda_X+\lambda_Y)(e^{\jmath\omega}-1)}.
\end{aligned}$$

This last is the characteristic function of a Poisson random variable with intensity $\lambda_X + \lambda_Y$. ∎

II. BINOMIAL DISTRIBUTION FOR POISSON R.V. CONDITIONED ON SUM

Theorem 3: Let X and Y be two Poisson random variables with intensities λ_x and λ_y, respectively. Then

$$p_{X|X+Y}(x \mid z) = \mathcal{B}\left(x \,\middle|\, z, \frac{\lambda_x}{\lambda_x+\lambda_y}\right)$$

Proof of Theorem 3:

$$\begin{aligned}
p_{X|X+Y}(x \mid z) &= \frac{\lambda_x^x e^{-\lambda_x}}{x!} \cdot \frac{\lambda_y^{z-x} e^{-\lambda_y}}{(z-x)!} \cdot \frac{z!}{(\lambda_x+\lambda_y)^z e^{-(\lambda_x+\lambda_y)}} \\
&= \binom{z}{x} \left(\frac{\lambda_x}{\lambda_x+\lambda_y}\right)^x \cdot \left(\frac{\lambda_y}{\lambda_x+\lambda_y}\right)^{z-x} \\
&= \binom{z}{x} \left(\frac{\lambda_x}{\lambda_x+\lambda_y}\right)^x \cdot \left(1 - \frac{\lambda_x}{\lambda_x+\lambda_y}\right)^{z-x} \\
&= \mathcal{B}\left(x \,\middle|\, z, \frac{\lambda_x}{\lambda_x+\lambda_y}\right)
\end{aligned}$$

∎

Fig. 15. Network Traf£c Strictly Linear Estimate

Fig. 16. Gamma Ray Burst Piecewise Linear Estimate

III. Multinomial Distribution for Poisson R.V.'s Conditioned on Sum

Theorem 4: Let $X_1 \dots X_N$ be N independent Poisson random variables with intensities $\lambda_1 \dots \lambda_N$, respectively. If $Z \equiv \sum_{n=1}^{N} X_n$, then

$$p_{X_1,X_2,\dots,X_N|Z}(x_1,x_2,\dots,x_N \mid z) = \mathcal{M}\left(x_1,x_2,\dots,x_N \,\middle|\, z, \frac{\lambda_{x_1}}{\lambda_z}, \frac{\lambda_{x_2}}{\lambda_z}, \dots, \frac{\lambda_{x_N}}{\lambda_z}\right).$$

Proof of Theorem 4: (by induction) For $N = 2$, we note that the theorem holds by Section B since

$$\begin{aligned}
\mathcal{B}\left(x \,\middle|\, z, \frac{\lambda_x}{\lambda_x+\lambda_y}\right) &= \binom{z}{x}\left(\frac{\lambda_x}{\lambda_x+\lambda_y}\right)^{x} \cdot \left(\frac{\lambda_y}{\lambda_x+\lambda_y}\right)^{z-x} \\
&= \frac{z!}{x!\cdot y!}\left(\frac{\lambda_x}{\lambda_x+\lambda_y}\right)^{x} \cdot \left(\frac{\lambda_y}{\lambda_x+\lambda_y}\right)^{z-x} \\
&= \mathcal{M}\left(x, y \,\middle|\, z, \frac{\lambda_x}{\lambda_x+\lambda_y}, \frac{\lambda_x}{\lambda_x+\lambda_y}\right).
\end{aligned}$$

For $N > 2$, we assume the theorem holds for $N-1$ and use this to prove that it holds for N:

$$\begin{aligned}
p_{X_1,X_2,\dots,X_N|Z}(x_1,x_2,\dots,x_N \mid z) &= p_{X_1,X_2,\dots,X_{N-1}|Z-X_N}(x_1,x_2,\dots,x_{N-1} \mid z-x_N) \cdot p_{X_N|Z}(x_N \mid z) \\
&= \mathcal{M}\left(x_1,\dots,x_{N-1} \,\middle|\, z-x_N, \frac{\lambda_{x_1}}{\lambda_z-\lambda_{x_N}}, \dots, \frac{\lambda_{x_{N-1}}}{\lambda_z-\lambda_{x_N}}\right) \cdot \mathcal{B}\left(x_N \mid z\right) \\
&= \frac{(z-x_N)!}{x_1!\cdots x_{N-1}!}\left(\frac{\lambda_{x_1}}{\lambda_z-\lambda_{x_N}}\right)^{x_1} \cdots \left(\frac{\lambda_{N-1}}{\lambda_z-\lambda_{x_N}}\right)^{x_{N-1}} \cdot \\
&\quad \frac{z!}{x_N!(z-x_N)!}\left(\frac{\lambda_{x_N}}{\lambda_z}\right)^{x_N} \cdot \left(\frac{\lambda_z-\lambda_{x_N}}{\lambda_z}\right)^{z-x_N} \\
&= \frac{z!}{x_1!\cdots x_N!}\left(\frac{\lambda_{x_1}}{\lambda_z}\right)^{x_1} \cdots \left(\frac{\lambda_{x_N}}{\lambda_z}\right)^{x_N} \\
&= \mathcal{M}\left(x_1,x_2,\dots,x_N \,\middle|\, z, \frac{\lambda_{x_1}}{\lambda_z}, \frac{\lambda_{x_2}}{\lambda_z}, \dots, \frac{\lambda_{x_N}}{\lambda_z}\right).
\end{aligned}$$

■

Fig. 17. Gamma Ray Burst Piecewise Polynomial Estimate