

---

# Online Bayesian Inference for a Latent Marked Poisson Process: Applications in Spike Sorting

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

## 1 Introduction

1 paragraph on the following:

1. opening
2. challenge
3. approach
4. resolution

## 2 Model

Our data is a time-series of multielectrode recordings  $\mathbf{X} \equiv (\mathbf{x}_1, \dots, \mathbf{x}_T)$ , and consists of  $T$  recordings from  $C$  channels. The set of recording times lie on regular grid with interval length  $\Delta$ , while  $\mathbf{x}_t \in \mathbb{R}^C$  for all  $t$ . This time-series of electrical activity is driven by an unknown number of neurons and we want to... outline scientific goals. We let the number of neurons be unbounded, though only a few of the infinite neurons dominate. These neurons contribute the majority of the activity in any finite interval of time; however, as time passes, the total number of observed neurons increases (Justify?). The neurons themselves emit continuous-time voltage traces, with the outputs of all neurons superimposed and discretely sampled to produce the recordings  $\mathbf{X}$ . At a high level, we model the output of each neuron as a series of idealized spikes smoothed with appropriate kernels (the latter determines the shape of each action potential). We describe this in detail, starting first with the model for a single channel recording  $X \equiv (x_1, \dots, x_T)$ .

### 2.1 Modelling a single neuron output

We model the spiking activity of each neuron as stationary and memoryless, with its set of spike times distributed as a homogeneous Poisson process. Comment on refractoriness or leave for discussion/future work. Similarly on the generalization to inhomogeneity? The neurons themselves are heterogeneous, with  $r_i$  the (unknown) firing rate for neuron  $i$ . Call the ordered set of spike times of the  $i$ th neuron  $E_i$ ; then the time between successive elements of  $E_i$  is exponentially distributed with mean  $1/r_i$ . We write this as

$$E_i \sim \text{PoisProc}(r_i) \quad (1)$$

The actual electrical output of a neuron is not a binary event; instead each spiking event is a smooth voltage perturbation about a resting state. This perturbation forms the shape of the spike (without any

loss of generality, we set the resting state to zero). (figure? better biological description? comment on how we preprocess the data to get zero mean?). While the spike shape varies across neurons as well as across different spikes of the same neuron, each neuron has its own characteristic distribution over shapes. Figure? We let  $\theta \in \Theta$  parametrize this distribution, and whenever neuron  $i$  emits a spike, we draw a voltage trace independently from the corresponding distribution. This is then offset to the time of the spike, and the complete output of the neuron is the superposition of all these spike waveforms. (Figure?) More concretely, we model each spike shape as a linear combination of a dictionary of  $K$  basis functions  $A \equiv (A_1(t), \dots, A_K(t))$ , shared across all neurons. For the  $i$ th neuron, the  $j$ th spike  $e_{ij} \in E_i$ , is associated with a random  $K$ -dimensional weight vector  $\tilde{y}_{ij}$ , and the shape of this spike is given by the weighted sum  $\sum_{k=1}^K \tilde{y}_{ijk} A_k(t)$ . We let  $\tilde{y}_{ij}$  be Gaussian distributed, with  $\theta_i \equiv (\mu_i, \Sigma_i)$  determining its mean and variance. Then, at any time  $t$ , the output of neuron  $i$  is

$$x_i(t) = \sum_{j=1}^{|E_i|} \sum_{k=1}^K \tilde{y}_{ijk} A_k(t - e_{ij}) \quad (2)$$

The total signal recorded  $x(t)$  at any electrode is the superposition of the outputs of all neurons. Define  $E = \cup_{i=1}^{\infty} E_i$  as the (ordered) superposition of the spike times of all neurons. Furthermore, let  $n(j)$  be the neuron to which the  $j$ th element of  $E$  belongs, and let  $p(j)$  index the position of the  $j$ th spike of  $E$  in the spike train  $E_{n(i)}$  of neuron  $n(i)$  (so that  $e_j = e_{n(j)p(j)}$ ). Then, we have that

$$x(t) = \sum_{i=1}^{\infty} x_i(t) = \sum_{j=1}^{|E|} \sum_{k=1}^K y_{jk} A_k(t - e_j) \quad (3)$$

where

$$y_j \equiv \tilde{y}_{n(j)p(j)} \sim N(\mu_{n(j)}, \sigma_{n(j)}) \quad (4)$$

From the superposition property of the Poisson process [?], the overall spiking activity  $E$  is a Poisson process with rate  $R = \sum_{i=1}^{\infty} r_i$ . The signal  $x(t)$  is a functional of a marked Poisson process, where the  $j$ th event is labelled by the neuron to which it is assigned ( $n(j)$ ), and the shape of its spike waveform ( $y_j$ ). From the properties of the Poisson process, we have that the marks  $n(j)$  are i.i.d. distributed with  $P(n(j) = i) = \frac{r_i}{R}$ . Given  $n(j)$ ,  $y_j$  is distributed as in equation 4.

## 2.2 Completely random measures (CRMs)

In this work, we take a nonparametric approach, letting the number of neurons be unbounded (so that  $n(i) \in \{1, 2, \dots\}$ ). Since only a finite number of spikes are observed in any finite interval, the total rate  $R$  must also be finite; moreover, as we described earlier, we want this to be dominated by a few  $r_i$ . A natural framework that captures these modelling requirements is that of completely random measures [?]. Completely random measures are stochastic processes that form flexible and convenient priors over infinite dimensional objects like probability distributions, hazard functions etc. These have been well studied in the Bayesian nonparametrics and machine learning community, and there exists a wealth of literature on their theoretical properties, as well as on computational approaches to posterior inference.

Recall that each neuron is characterized by a pair  $(r_i, \theta_i)$ ; the former characterizes the distribution over spike times, and the latter over spike shapes. We map the infinite collection of pairs  $\{(r_i, \theta_i)\}$  to an atomic measure on  $\Theta$ :

$$R(d\theta) = \sum_{i=1}^{\infty} r_i \delta_{\theta_i} \quad (5)$$

For any subset  $\Theta$  of  $\Theta$ , the measure  $R(\Theta)$  equals  $\sum_{\{i: \theta_i \in \Theta\}} r_i$ . We allow  $R(\cdot)$  to be random, modelling it as a realization of a completely random measure. Such a random measure has the property that for any two disjoint subsets  $\Theta_1$  and  $\Theta_2 \in \Theta$ , the measures  $R(\Theta_1)$  and  $R(\Theta_2)$  are independent. This distribution over measures is induced by a distribution over the infinite sequence of weights (the  $r_i$ 's), and a distribution over the sequence of their locations (the  $\theta_i$ 's). For a CRM,

the weights  $r_i$  form the jumps of a Lévy process [? ], and their distribution is characterized by a Lévy intensity  $\rho(r)$ . The locations  $\theta_i$  are drawn i.i.d. from a base probability measure  $H(\theta)$ ; we let this be the conjugate normal-Wishart distribution. As it typical, we assume these to be independent (though this is not necessary). **if there's space, I can elaborate on the construction of the CRM from its Levy measure, though this is not necessary**

The CRM we choose is the Gamma process (GP); this has Lévy intensity  $\rho(r) = r^{-1} \exp(-r\alpha)$ . The Gamma process has the convenient property that the total mass  $R \equiv R(\Theta) = \sum_{i=1}^{\infty} r_i$  is Gamma distributed (and thus conjugate to the Poisson process prior on  $E$ ). The Gamma process is also closely connected with the Dirichlet process [? ], which will prove useful later on. Other choices of the Lévy intensity can capture greater uncertainty in the number of neurons active in any finite interval, power-law behaviour etc. In any case, our overall model is then:

$$R(d\theta) \sim \Gamma P(\alpha, H(\theta)) \quad (6)$$

$$E_i \sim \text{PoisProc}(r_i) \quad i \text{ in } 1, 2, \dots \quad (7)$$

$$\tilde{y}_{ij} \sim N(\mu_i, \Sigma_i) \quad i, j \text{ in } 1, 2, \dots \quad (8)$$

$$x_i(t) = \sum_{j=1}^{|E_i|} \tilde{y}_{ij} A_j(t - e_{ij}) \quad (9)$$

$$X = \sum_{i=1}^{\infty} x_i \quad (10)$$

It will be more convenient from the point of inference to work with the marked Poisson process representation of equations 3 and 4. The superposition process  $E$  is a rate  $R$  Poisson process, and under a Gamma process prior,  $R$  has a conjugate Gamma distribution with shape and scale parameters 1 and  $\alpha$  respectively. As we saw, the labels  $n(\cdot)$  assigning events to neurons are drawn i.i.d. from a normalized Gamma process  $G(d\theta)$ :

$$G(d\theta) = \frac{r_j}{R} \quad (11)$$

$G(d\theta)$  is a random probability measure that belongs to a class called a normalized random measures [? ]; for the Gamma process, this is a draw from the Dirichlet process. For the  $j$  spike, given its neuron assignment  $n(i)$ , its shape vector is drawn from a normal distribution with parameters  $(\mu_{n(j)}, \Sigma_{n(j)})$ . Thus the weight vectors are distributed according to a Dirichlet process mixture model, with the neurons forming clusters. This insight allows us to marginalize out the infinite-dimensional rate vector  $R$ , and assign spikes to neurons via the Chinese restaurant process (CRP). Under the CRP, the  $j$ th spike is assigned to one of the earlier neurons with probability proportional to the number of earlier spikes assigned to that neuron. It is assigned to a new neuron with probability  $\alpha$ . Unlike most applications with observe the outputs of a CRP, we observe a functional of it. Furthermore, (again, for the Gamma process), the random probability measure  $G$  is independent of the total mass  $R(\Theta)$ . We thus have the following model equivalent to the one above:

$$R \sim \text{Gamma}(1, \alpha) \quad (12)$$

$$G(d\theta) \sim \text{DP}(\alpha) \quad (13)$$

$$E \sim \text{PoisProc}(R) \quad (14)$$

$$n(j) \sim G, \quad j = 1, \dots, |E| \quad (15)$$

$$y_e \sim N(\mu_{n(e)}, \Sigma_{n(e)}), \quad j = 1, \dots, |E| \quad (16)$$

$$x(t) = \sum_{j=1}^{|E|} \sum_{k=1}^K y_{jk} A_k(t - e_j) \quad (17)$$

### 2.3 A discrete-time approximation

In the previous paragraphs, we described a continuous-time voltage output by a neuron. Our data on the other hand consists of recordings at a discrete set of times. While it is possible to make inferences about the continuous-time process that underlies these discrete recordings, in this work we restrict ourselves to discrete-time inferences. Towards this, we start by providing a discrete-time

approximation to the model above. This follows easily from the marked Poisson process characterization of the model. Recall first the Bernoulli approximation to the Poisson process: a sample from a Poisson process with rate  $R$  can be approximated by discretizing time at a granularity  $\Delta$ , and assigning each interval an event independently with probability  $R\Delta$ . This approximation becomes exact as  $\Delta$  tends to 0.

This suggests the following approximation at a time resolution  $\Delta$ . Draw the random Poisson process rate  $R$  drawn from a  $\text{Gamma}(1, \alpha)$  distribution. Simultaneously, draw a random probability measure  $G$  from a Dirichlet process. Assign an event to an interval independently with probability  $R\Delta$ , and to each event, assign a random mark drawn from the DP. Given the marks, we can evaluate the recordings at each time.

## 2.4 Noise and nonstationarity

The signal recorded by an electrode is the neuron output corrupted by noise, we model this noise as independent of the signal, additive and Gaussian. However, rather than modelling the noise as independent across time bins, we model it as a first-order autoregressive process. This can capture effects like the movement of electrodes during the experiment. Furthermore, rather than keeping the cluster parameters fixed, we model these as AR processes as well, capturing the evolution of the neuron shape with time.

## 2.5 Modelling multielectrode recordings

## 3 Inference

We perform inference in an online manner [? ]. As observations arrive, our inference algorithm decides whether or not a new spike is present, which neuron (cluster) to assign that spike to, as well as the shape of the spike waveform. On the other hand, our algorithm maintains a posterior distribution over the cluster parameters that characterize the distribution over shapes. Having identified the location and shape of spikes from earlier times, we subtract these from the observations treat the residual as an observation from a DP mixture model. The cluster assignment of earlier spikes determines the seating arrangement of customers in the Chinese restaurant associated with the DP. Given the corresponding distribution over parameters,  $p(\theta)$ , we decide whether there is an underlying spike, which cluster it is assigned to, and what the shape of that spike is. We simultaneously update the distribution over parameters of clusters. Assume each spike waveform spans  $W$  time intervals. Define the residual at time  $t$  as  $X_t - \sum_{i=1}^W A_i$ . At time  $t$ , let  $y_t$  represent the shape of the action potential. Letting  $\tilde{x}_t$  be the observation at time  $t$ , we have

$$z_t \sim \text{Bern}(p) \tag{18}$$

$$\text{if } z_t = 1$$

$$\gamma_t | \gamma_{1:t-1} \sim \text{CRP} \tag{19}$$

$$\theta_t | \gamma_t = i \sim \mathcal{N}(\theta_{t-1}, \Sigma) \tag{20}$$

## 4 Results

### 4.1 Performance on Partial Ground Truth Data

### 4.2 Time-Varying Waveform Adaptation

### 4.3 Overlapping Spike Detection

### 4.4 Michigan Data

### 4.5 dec draft stuff

### 4.6 Publicly Available Data

Experiments were performed on a subset of the publicly available<sup>1</sup> dataset described in [? ]. We used the dataset d533101 that consisted of an extracellular tetrode and a single intracellular electrode. The recording was made simultaneously on all electrodes and was set up such cell with the intracellular electrode was also recorded on the extracellular array.

The intracellular recording is relatively noiseless and gives nearly certain firing times of the intracellular neuron. The extracellular recording contains the spike waveforms from the intracellular neuron as well as an unknown number of additional neurons. The data is a 4-minute recording at a 10kHz sampling rate and preprocessed with a high-pass filter at 800 Hz.

Each algorithm gives a clustering of the detected spikes. In this dataset, we only have a partial ground truth, so we are only able to analyze whether a spike comes from the intracellular (IC) recording or not. In these experiments, we define a detected spike to be an IC spike if the IC recording has a spike within .5ms of the detected spike. We define the cluster with the greatest number of intracellular spikes as the "IC cluster." Figure 1a shows the performance on the intracellular cluster versus competing methods. The single-channel experiments were all run on channel 2 (the results were nearly identical for all channels). The spike detections for the offline methods were given by using a threshold at 3 times the noise standard deviation [? ]. (Need to do more comparisons and descriptions) and windowed at a size  $P = 30$  (check if  $P$  matches with Vinayak notation). The action potential window was set at 30, and PCA was used to reduce the space to  $K=3$  for the experiments. The results were very similar for  $K=2$ ,  $K=3$ , and  $K=4$ . There were 3742 spikes detected with the threshold, and 753 of those corresponded to the intracellular spikes.

The online algorithms were all run with weakly informative parameters (add parameters once I get vinayak's notation). The parameters were insensitive to minor changes. Running time in unoptimized MATLAB code for 4 minutes of data was 31s was a single channel and 3 minutes for all 4 channels on a 3.2GHz Intel Core i5 machine with 6GB of memory. We show the inferred  $y_k$  for detected spikes in the 2-PCA space in Figure 1b.

As as been demonstrated previously (cite paninski), the waveform shape of a neuron may change over time. The mean waveform over time for the intracellular neuron is shown in Figure 2a. For the IC cluster, it is of interest to investigate the properties of the true positives, the false positives, as well as the IC spikes that are missed by the algorithm. Errorbar plots for these groups are shown in Figure 2b; this figure demonstrates the great similarity of the true positives and the false positives, while the missed positives (spike waveforms not detected or not in the IC cluster) have high variability and a different mean shape.

To get an idea of the robustness of the algorithm, we split the data into random segments of 2 minutes, and ran the algorithms on that dataset. Add plot and information.

## 5 Multi-Electrode Sensors

In the tetrode case the spike undoubtedly appears on all channels at once. Often, people will just concatenate the channels to process the data (eg. cites). When the action potential will only appear on a subset of channels it is nice to allow the action potential to vary in a low-dimensional subset

<sup>1</sup><http://crcns.org/data-sets/hc/hc-1/>

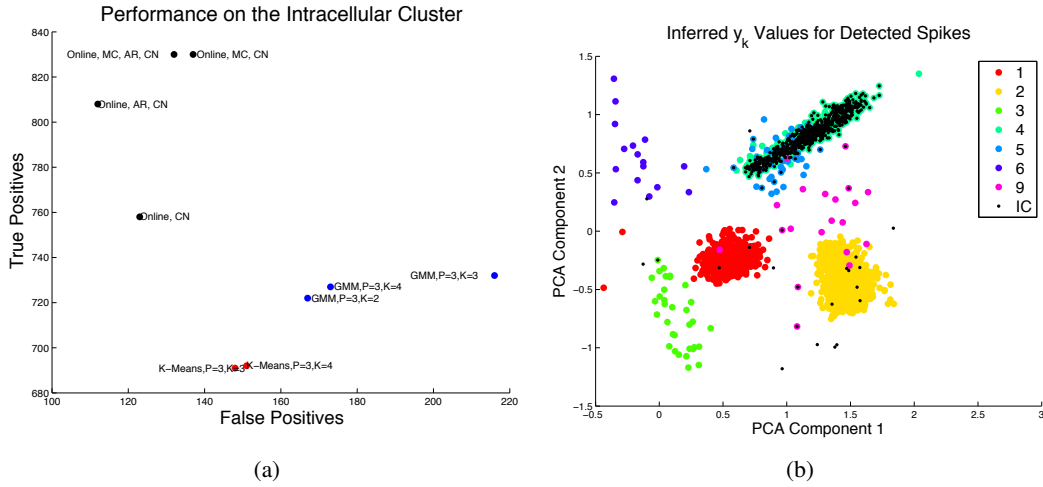


Figure 1: Results on the d533101 dataset. (a) This shows the number of true positives versus the number of false positives. Our methods do better, yay! (b) Results on the d533101 dataset from the online algorithm with an AR mean

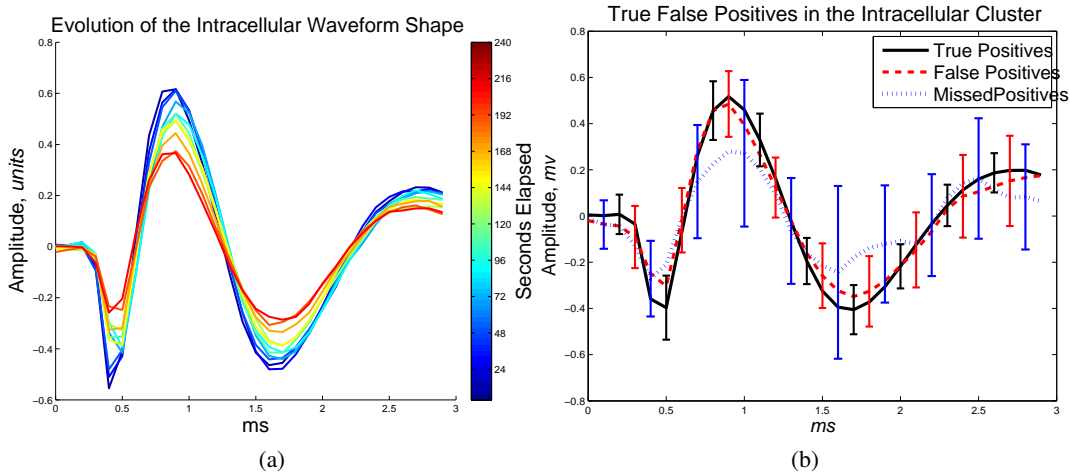


Figure 2: (a) Mean IC waveforms over time. Each colored line represents the mean of the waveform averaged over 24s and the color gives the elapsed time. This neuron decreases in amplitude over the period of the recording. (b) Errorbar plots of the true positives and the false positives in the IC cluster. While the false positives have slightly more variability, the mean shape for the false positives and the true positives is nearly identical. The true misses have a significantly lower amplitude as well as high variability

in each of the channels instead of a low-dimensional subset over all the channels. This is especially important when the appropriate dictionary is not known beforehand.

We use processed data from novel NeuroNexus devices. (Need to give a description of this). The first device we consider is a set of 3 channels of data shown in Figure 3a. The neighboring electrode sites in these devices have  $30 \mu\text{m}$  between electrode edges and  $60 \mu\text{m}$  between electrode centers. These devices are close enough that a locally-firing neuron could appear on multiple electrode sites (cite that paper on action potential overlap), and neighboring channels warrant joint processing.

The top 3 clusters found in the first 10 minutes of data are shown in Figure (add in a bit).

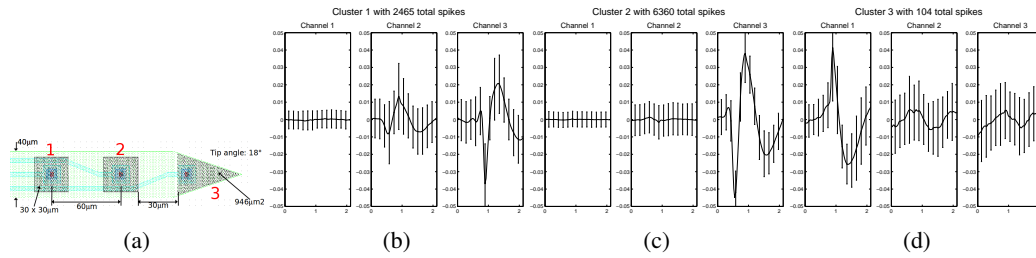


Figure 3: Device used. Channels in large, red numbers.

## 6 Discussion

1 paragraph on:

1. summary
2. relate work - including Franke and Pillow
3. future work - including adaptive stimulus design & decoding