
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

3 The model/introduction

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 Δ , 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... recap 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 generate 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 which are smoothed with appropriate kernels, the latter determining the shape of each spike. We describe this in detail, starting first with a model for a single channel recording $X \equiv (x_1, \dots, x_T)$.

3.1 A single neuron model

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 the i th neuron having an (unknown) firing rate r_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

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

The actual electrical output of a neuron is not binary; instead each spiking event is a smooth perturbation in voltage 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 shapes vary 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, with neuron i having parameter θ_i^* . Whenever this neuron emits a spike, a new shape is drawn independently from the corresponding distribution. This is then offset to the time of the spike, and forms the voltage trace associated with that spike. The complete output of the neuron is the superposition of all these spike waveforms. (Figure?) We model the random spike shapes themselves as weighted superpositions of a dictionary of K basis functions $A \equiv (A_1(t), \dots, A_K(t))$. The dictionary elements are shared across all neurons, and each is a real-valued function of time. For the i th neuron, the j th spike $e_{ij} \in E_i$, is associated with a random K -dimensional weight vector $\tilde{\mathbf{y}}_{ij}$, and the shape of this spike is given by the weighted sum $\sum_{k=1}^K \tilde{y}_{ijk} A_k(t)$. We let $\tilde{\mathbf{y}}_{ij}$ 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) union of the spike times of all neurons. To map elements $e_j \in E$ to those in the individual spike trains E_i , we need to introduce some more notation. Let $n(j)$ be the neuron to which the j th element of E belongs, and let $\theta_j \equiv (\mu_j, \Sigma_j)$ be the parameter associated with that spike. Then $\theta_j = \theta_{n(j)}^*$. Furthermore, let $p(j)$ index the position of the spike j of E in $E_{n(j)}$, the spike train of neuron $n(j)$. Thus $e_j = e_{n(j)p(j)}$. Finally, define $\mathbf{y}_j = \tilde{\mathbf{y}}_{n(j)p(j)}$. 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

$$\mathbf{y}_j \sim N(\mu_j, \Sigma_j) \quad (4)$$

From the superposition property of the Poisson process [1], the overall spiking activity E is a Poisson process with rate $R = \sum_{i=1}^\infty r_i$. Each event $e_j \in E$ is associated with a pair of labels, the parameter of the neuron to which it is assigned ($\theta_j = (\mu_j, \Sigma_j)$), and the weight-vector characterizing the spike shape (\mathbf{y}_j). We can view these as the ‘marks’ of a marked Poisson process. From the properties of the Poisson process, we have that the marks θ_j are i.i.d. distributed from a measure $G(d\theta)$, with $G(\theta_i) = P(n(j) = i) = \frac{r_i}{R}$, and 0 otherwise. Given θ_j , \mathbf{y}_j is distributed as in equation 4. The output waveform $x(t)$ is then a linear functional of this marked Poisson process.

3.2 Completely random measures (CRMs)

While the previous section described the output of a single neuron, our recordings are actually a superposition of the outputs of an unknown number of neurons. We deal with this by taking a nonparametric Bayesian approach, and letting the number of neurons be unbounded (so that $n(j) \in \{1, 2, \dots\}$). 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 [2]. Completely random measures are stochastic processes that form flexible and convenient priors over infinite dimensional objects like probability distributions [3], hazard functions [4], latent features [5] etc. These have been well studied in the Bayesian nonparametrics and machine learning communities, 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, θ_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 Θ :

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

For any subset Θ of Θ , 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 Θ_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 θ_i 's). For a CRM, the weights r_i are the jumps of a Lévy process [6], and their distribution is characterized by a Lévy measure $\rho(r)$. The locations θ_i are drawn i.i.d. from a base probability measure $H(\theta)$. 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 for our application 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 rate $R \equiv R(\Theta) = \sum_{i=1}^{\infty} r_i$ is Gamma distributed (and thus conjugate to the Poisson process prior on E). The Gamma process is also closely connected with the Dirichlet process [7], 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.

To complete the specification on the Gamma process, we need to set the base-measure $H(\theta)$. Recalling that $\theta \equiv (\mu, \Sigma)$ gives the mean and variance of the weight-vector \mathbf{y} of each neuron; we set $H(\theta)$ to be the conjugate normal-Wishart distribution. Our overall model is then:

$$R(d\theta) \sim \text{GP}(\alpha, H(\theta)) \quad (6)$$

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

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

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

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

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 α respectively. As we saw, the labels θ_i 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 [3]; for the Gamma process, this is a draw from the Dirichlet process. For the j spike, given its parameter θ_j , its shape vector is drawn from a normal distribution with parameters (μ_j, Σ_j) . Thus the spike parameters θ are i.i.d. from a Dirichlet process, while the weight vectors are draws from a DP mixture model with Gaussian likelihoods.

With this insight, we can now 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 the same parameter as an earlier spike with probability proportional to the number of earlier spikes having that parameter. It is assigned to a new parameter with probability α . A final point is that 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:

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

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

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

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

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

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

Unlike most applications with observe the outputs of a CRP, we observe a functional of it.

3.3 A discrete-time approximation

In the previous sections, we described a continuous-time voltage waveform 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 paper we restrict ourselves to discrete case.

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 Δ , and assigning each interval an event independently with probability $R\Delta$. The accuracy of this approximation increases as Δ tends to 0. In our case, all we require additionally is to assign the marks θ_i and \mathbf{y}_i to each event in this approximation. The θ_i 's are distributed according to a Chinese restaurant process, while each \mathbf{y}_i is drawn from a normal distribution parametrized by the corresponding θ_i . The elements of dictionary \mathbf{A} are discretized as well, and each element is now a finite-dimensional vector instead of a function. We set the length of each dictionary element to L , so that \mathbf{A} is an K -by- L matrix. The shape of the j th spike is now a vector of length L , and is given by $\mathbf{A}\mathbf{y}$.

3.4 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.

3.5 Modelling multielectrode recordings

4 Inference

We perform inference in an online manner [8]. 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 $X_t - \sum_{i=1}^W A$. 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

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

if $z_t == 1$

$$\gamma_t | \gamma_{1:t-1} \sim CRP \quad (19)$$

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

5 Inference

We perform inference in an online manner [8]. 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 $X_t - \sum_{i=1}^W A$. 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

$$z_t \sim \text{Bern}(p) \quad (21)$$

if $z_t == 1$

$$\gamma_t | \gamma_{1:t-1} \sim CRP \quad (22)$$

$$\theta_t | \gamma_t = i \sim \mathcal{N}(\theta_{t-1}, \Sigma) \quad (23)$$

6 Results

6.1 Performance on Partial Ground Truth Data

6.2 Time-Varying Waveform Adaptation

6.3 Overlapping Spike Detection

6.4 Michigan Data

6.5 dec draft stuff

6.6 Publicly Available Data

Experiments were performed on a subset of the publicly available¹ 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 a 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

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

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.

7 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 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).

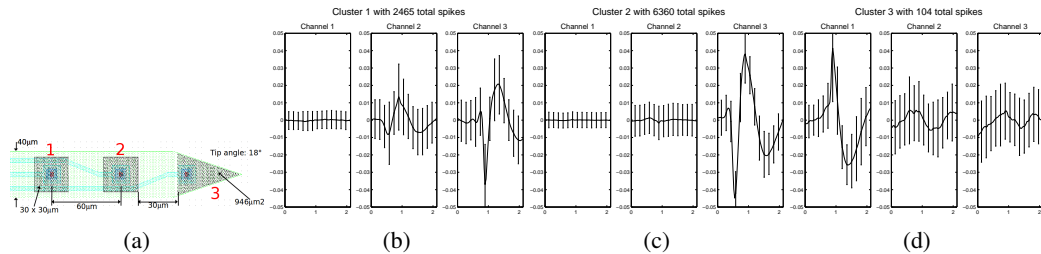
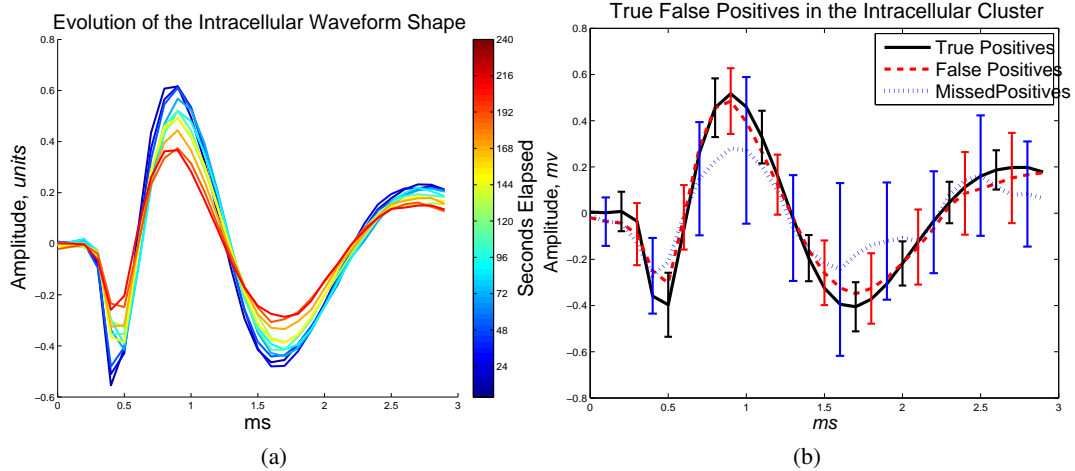
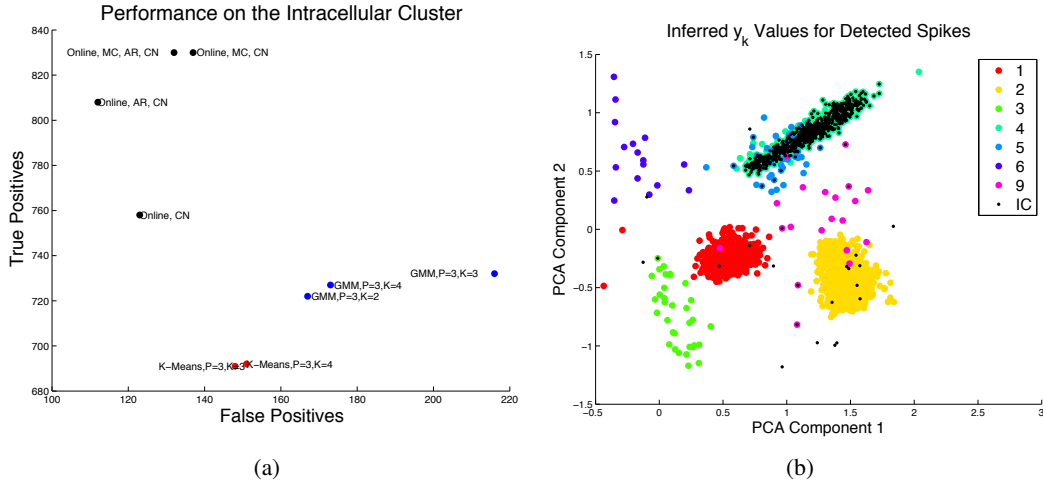
8 Discussion

1 paragraph on:

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

References

- [1] J. F. C. Kingman. *Poisson processes*, volume 3 of *Oxford Studies in Probability*. The Clarendon Press Oxford University Press, New York, 1993. Oxford Science Publications.



378 [2] J.F.C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78,
379 1967.

380 [3] L.F. James, A. Lijoi, and I. Pruenster. Posterior analysis for normalized random measures with
381 independent increments. *Scand. J. Stat.*, 36:76–97, 2009.

382 [4] N. L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history
383 data. *Annals of Statistics*, 18(3):1259–1294, 1990.

384 [5] R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In
385 *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 11,
386 2007.

387 [6] Ken ti Sato. *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press,
388 1990.

389 [7] Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of*
390 *Statistics*, 1(2):209–230, 1973.

391 [8] L. Wang and D.B. Dunson. Fast bayesian inference in dirichlet process mixture models. *Journal*
392 *of Computational & Graphical Statistics*, 2009.

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431