Journal of Neuroscience Methods 173 (2008) 1–12

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

A nonparametric Bayesian alternative to spike sorting

Frank Wood [a,*], Michael J. Black [b]

[a] Gatsby Computational Neuroscience Unit, University College London, Alexandra House, 17 Queen Square, London WC1N 3AR, UK
[b] Department of Computer Science, Brown University, Providence, RI, USA

ARTICLE INFO

Article history:
Received 15 February 2008
Received in revised form 16 April 2008
Accepted 18 April 2008

Keywords:
Spike sorting
Nonparametric Bayesian methods
Dirichlet process mixture models

ABSTRACT

The analysis of extra-cellular neural recordings typically begins with careful spike sorting and all analysis of the data then rests on the correctness of the resulting spike trains. In many situations this is unproblematic as experimental and spike sorting procedures often focus on well isolated units. There is evidence in the literature, however, that errors in spike sorting can occur even with carefully collected and selected data. Additionally, chronically implanted electrodes and arrays with fixed electrodes cannot be easily adjusted to provide well isolated units. In these situations, multiple units may be recorded and the assignment of waveforms to units may be ambiguous. At the same time, analysis of such data may be both scientifically important and clinically relevant. In this paper we address this issue using a novel probabilistic model that accounts for several important sources of uncertainty and error in spike sorting. In lieu of sorting neural data to produce a single best spike train, we estimate a probabilistic model of spike trains given the observed data. We show how such a *distribution* over spike sortings can support standard neuroscientific questions while providing a representation of uncertainty in the analysis. As a representative illustration of the approach, we analyzed primary motor cortical tuning with respect to hand movement in data recorded with a chronic multi-electrode array in non-human primates. We found that the probabilistic analysis generally agrees with human sorters but suggests the presence of tuned units not detected by humans.

1. Introduction

A "spike train" is a temporal sequence containing the times of all action potentials for a given cell. Such spike trains are central to the analysis of neural data and are generated from electrophysiological recordings by a process called *spike sorting* (see, for example, Lewicki, 1998 for a review). Spike sorting involves detecting all action potentials that occur in a neurophysiological recording and labeling them to indicate from which neuron each came. In this paper we present a new approach to spike sorting that differs from almost all prior art in that no single best spike train is sought. Instead a *distribution over spike trains is estimated* which allows us to express uncertainties that arise during the spike sorting process and in the results of neural data analysis. To do so we expand on the infinite Gaussian mixture model (IGMM) spike sorter of Wood et al. (2004a) and illustrate how it can be used as a model of spike trains.

We are motivated by traditional neural data analyses that rest on the accuracy of spike trains. Among many others these include hypothesis tests about single cells or population characteristics, and decoding from single or multiple neurons. Currently it is typical in such problems to assume that the spike trains upon which the analysis is based are unambiguously correct. In other words, any uncertainties that might have arisen during the conversion from an analog voltage trace to a discrete spike train are ignored. While it is common practice to report in published results the spike sorting methodology employed and signal to noise ratios for analyzed data, this does not readily admit a quantification of the uncertainty in any conclusions based on the data. Unfortunately several studies have demonstrated that spike sorting is not always unambiguous and that expert human spike sorters can, and do, produce spike trains that vary extensively (Harris et al., 2000; Wood et al., 2004a). The effect of such variability is not widely reported and is not commonly analyzed. Many analyses consequently rely on only well isolated, and hence less ambiguous neurons. This, of course, may limit both the kind and amount of data that can be analyzed.

We argue that new statistical tools are needed to make the analysis of such ambiguous neural data practical. In particular there are situations in which discarding data that is potentially ambiguous may be undesirable or impractical. An example of this involves

* Corresponding author. Tel.: +44 7951495087.
E-mail addresses: fwood@gatsby.ucl.ac.uk (F. Wood), black@cs.brown.edu (M.J. Black).

0165-0270/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2008.04.030

chronically implanted human and non-human primates where one might expect the implanted recording device to work for years without post-implant adjustment. Chronic recordings can yield a wealth of information but may contain poorly isolated units (Joshua et al., 2007) which may in turn produce waveforms that are difficult to sort unambiguously. For human neural prostheses in particular, it may be important to make the best of the available signal, even if it is ambiguous.

The proposed nonparametric Bayesian (NPB) approach differs significant from typical automated spike sorting approaches. Our aim is not a new and improved automated spike sorter like those proposed by Nguyen et al. (2003), Shoham et al. (2003), Wood et al. (2004b), Hulata et al. (2002), Lewicki (1998), Takahashi et al. (2003), or Sahani et al. (1998). These previous methods attempt to produce a single "best" spike train given a recording. Rather than try to remove the inherent uncertainty we develop a probabilistic model to represent it explicitly and automatically account for its effect on spike train analysis. The approach enables the evaluation of the confidence in neural data analysis outcomes with respect to spike sorting variability.

In this paper we apply nonparametric Bayesian estimation and inference techniques to the problem of spike sorting. Specifically we develop an infinite Gaussian mixture model (Rasmussen, 2000) in which the number of units present on a recording channel does not need to be known *a priori*. In Nguyen et al. (2003) a similar approach was taken, and we, like them, demonstrate that our approach is a practical improvement over traditional Gaussian mixture model (GMM) spike sorting. In Nguyen et al. (2003) maximum *a posteriori* model estimation was used to learn the number of neurons in a recording. Our approach is similar in that we too estimate how many neurons are in the recording, but we go one step further and suggest how to use a probabilistic model to express spike sorting uncertainty in subsequent spike train analyses.

Like in Nguyen et al. (2003) and many other spike sorting papers we restrict our focus to a subproblem of the overall spike sorting problem. In particular our focus is on the process of determining from which neuron, out of an unknown number of neurons, each action potential arose based on action potential waveshape alone. In doing so we overlook the problem of spike detection and instead refer readers to Radons et al. (1994) for treatment of that topic. Additionally we ignore the problem of detecting overlapping spikes (deconvolving coincident action potentials) here referring readers to the work of Fee et al. (1996) and Görür et al. (2004). Finally, our analysis here does not take into account the refractory period of a cell. In principle, however, the Bayesian framework developed here can be extended to model these additional aspects of the problem.

Finally, the method is illustrated with an analysis of action potentials recorded from primary motor cortex in a monkey performing reaching tasks. Specifically we present an analysis of direction tuning in units recorded using a chronically implanted microelectrode array. We demonstrate how our method allows us to make use of all of the recorded data by accounting for spike sorting uncertainty. The modeling consequence of our approach is that, given a recording, we build a distribution over tuning directions. This stands in contrast to the typical finding of a single direction for each manually identified unit. Evidence gleaned from examining the estimated model suggests that additional, potentially valuable information can be extracted from suboptimal neural data by accounting for spike sorting uncertainty in the way we suggest.

The following section briefly summarizes the data used here. Section 3 introduces notation and reviews finite Gaussian mixture modeling. Section 4 reviews infinite Gaussian mixture modeling and two kinds of IGMM estimation, one batch algorithm and a second "sequential" or "iterative" algorithm. Section 5 then uses the IGMM to analyze neural signals.

2. Spike sorting

Suppose that we have N action potential waveforms from a single channel neurophysiological recording $R = [\vec{t}_1, \ldots, \vec{t}_N]$, where each waveform is represented by a vector of $n = 40$ voltage samples, $\vec{t}_i = [t_i^1, \ldots, t_i^n]^{\mathrm{T}} \in \mathbb{R}^n$. Our goal is to build a posterior distribution over sortings of this recording; a distribution which simultaneously represents how many neurons are present and which waveforms came from which neurons.

Instead of clustering the waveforms in the high dimensional ($n = 40$) space, we cluster a reduced dimensionality representation of the waveforms, where each $\vec{t}_i$ is represented in a lower dimensional basis obtained via principal component analysis (PCA). Given a set of recordings we perform PCA and keep only the first $D = 2$ eigenvectors characterizing the largest variance in the data and approximate a waveform as $\vec{t}_i \approx \mu + \sum_{d=1}^{D} y_i^d \vec{u}_d$. Here μ is the mean waveshape in R, $\vec{u}_d$ is the dth PCA basis vector, and the y_i^d are linear coefficients. The projection of six channels of monkey motor corti-

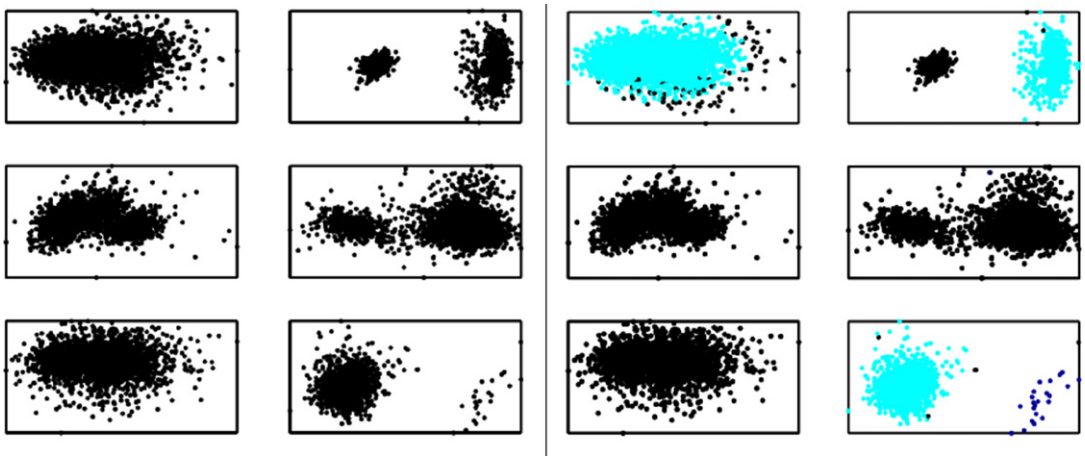

Fig. 1. Example spike sorting problem. Spiking data for six chronically implanted electrodes are shown. Spike waveforms are projected onto the first two principal component directions; each dot represents a single spike. Left: unsorted data. Right: human sorted (light and dark blue indicate a "spike" while black indicates that the datapoint was deemed too ambiguous to sort).

cal data onto the first two PC's is shown in Fig. 1 (left). A detailed description of this data is given in Section 5.

The human sorting, Fig. 1 (right), was performed using offline spike sorting software (Plexon Inc., 2003). Note that in this case, the sorter appears to have been quite conservative and many of the waveforms were deemed too ambiguous to include in subsequent analysis. We will return to this example later in the paper.

Our spike sorting algorithm clusters the low dimensional representation of the waveforms $\mathcal{Y} = [\vec{y}_i, \ldots, \vec{y}_N]$ rather than the full waveforms, so, for the remainder of this paper, when we write "event", "spike", or "waveform" it should be read as shorthand for "low dimensional waveform representation".

We follow Lewicki (1998) in making the common assumption that the distribution of waveforms from a single neuron is well approximated by a multivariate Gaussian.

3. Finite Gaussian mixture modeling

A finite Gaussian mixture model is a latent variable probabilistic model formed by adding together weighted Gaussians. Finite GMM's are frequently used because they can both approximate arbitrary multi-modal probability densities and cluster data if the latent variables are interpreted as class labels.

In spike sorting this interpretation of the latent variables is crucial. Each neuron is identified by its mean action potential waveform and the statistics of how this waveform varies between spikes. The Gaussian assumption means that only second-order statistics of waveform variability are accounted for (by a per-neuron covariance matrix). The semantics of the Gaussian mixture model are such that each individual waveform is seen as being generated from a single stochastically chosen neuron. The traditional Gaussian mixture model notation

$$P(\vec{y}_i) = \sum_{k=1}^{K} P(c_i = k) P(\vec{y}_i | \theta_k)$$

makes this intuition precise. Here K is the number of units present on the channel (assumed known and finite for now) and c_i indicates from which neuron each waveform arose (i.e. $c_i = k$ means that the ith waveform came from neuron k). $P(c_i = k) = \pi_k$ represents the *a priori* probability that the waveform was generated by neuron k. Finally the likelihood $P(\vec{y}_i|\theta_k)$ is the generative model of waveforms arising from neuron k. This is taken to be a multivariate Gaussian with parameters θ_k for each neuron k.

For the remainder of the paper, we will specify this model with the following notation:

$$\begin{aligned} c_i | \vec{\pi} &\sim \text{Discrete}(\vec{\pi}) \\ \vec{y}_i | c_i = k; \Theta &\sim \text{Gaussian}(\cdot | \theta_k). \end{aligned} \tag{1}$$

The notation $y|x; z \sim f(y, x; z)$ means that y is conditionally distributed given x and parameter z according to the probability density function f. Here by "Gaussian" we mean the multivariate normal (MVN) density function and by "Discrete" we mean the discrete (multinomial) distribution defined by a vector whose entries sum to one.

Additionally we will sometimes utilize more general names for the variables in this model. For instance, instead of "neuron" we will often write "class". Thus $\mathcal{C} = \{c_i\}_{i=1}^N$ is the collection of all class indicator variables, $\Theta = \{\theta_k\}_{k=1}^K$, is the collection of all class parameters, $\theta_k = \{\vec{\mu}_k, \mathbf{\Sigma}_k\}$ are the mean and covariance for class k, and $\vec{\pi} = \{\pi_k\}_{k=1}^K$, $\pi_k = P(c_i = k)$ are the class prior probabilities.

To use a GMM for spike sorting, one must estimate model parameters from the data. Expectation maximization (EM), introduced by (Dempster et al., 1977) in general and Lewicki (1998) in the context of spike sorting, is a maximum likelihood parameter estimation approach that can be used to estimate model parameters in situations where there are missing or latent data. Here the parameters of the model are Θ and $\vec{\pi}$; $\mathcal{C}$ is a hidden (or latent) variable. Given a set of waveforms $\mathcal{Y}$ our estimation task is to find the waveform means and covariances Θ for each neuron and their relative firing rates, or equivalently their prior probability, $\vec{\pi}$.

Maximum likelihood approaches such as EM all seek a single "best" model of the data which is the intuitive thing to do if a single setting of the model parameters is the quantity of interest. However there can be problems with this approach, particularly if the objective being maximized leads to multiple equivalently good models of the data. Also, some quantities, such as the number of neurons in the recording (K) can be difficult to optimize explicitly, leading to difficult model selection problems.

While it is common practice to try to find a single best model (with the hope that this model corresponds to the ground truth), it may not always be necessary or beneficial to do so. With respect to spike sorting, if the ultimate data analysis goal is to reach some conclusion about the neural population recorded on a single channel ("Is there evidence of excess synchrony?", "Does this brain area have cells that are strongly correlated to some stimulus?", etc.) it may be possible to express the analysis in such a way that its result can be averaged over multiple models (spike sortings). Then, for instance in cases where determining the best model is difficult, it may be beneficial to report an average result computed across multiple models of the data. In other words, if a particular single channel recording is hard to spike sort because of overlapping clusters (for instance), sorting it multiple times and repeating the analysis using each different sorting will provide evidence about the robustness of the particular analytical finding. This is both because the distribution of results will tend towards the true distribution over results, and also because a distribution over results allows confidence in the analysis result to be reported.

One way to do this is to take a Bayesian approach to spike sorting. This means that the model parameters will themselves be treated as random variables and that generative models for the parameters will be specified as well; these being called "priors." Thus, instead of estimating a single best model (set of parameters), our goal will be to estimate a distribution over models (parameters) instead. This is called the posterior distribution. As a reminder, Bayes' rule tells us that the posterior probability of a model $\mathcal{M}$ given a set of observations $\mathcal{Y}$ is proportional to the likelihood of the observations under the model $P(\mathcal{Y}|\mathcal{M})$ times the prior probability of the model $\mathcal{M}$:

$$P(\mathcal{M}|\mathcal{Y}) \propto P(\mathcal{Y}|\mathcal{M}) P(\mathcal{M}). \tag{2}$$

To estimate the posterior distribution we first have to specify a prior for all of the parameters of the model. For both purposes of illustration and for use in subsequent sections we follow Fraley and Raftery (2005) in choosing priors of the following types for the model parameters: Dirichlet for $\vec{\pi}$ and Gaussian times inverse Wishart for the Gaussian parameters (Gelman et al., 1995). These priors are chosen for mathematical convenience and interpretable expressiveness. They are conjugate priors[1] which will allow us to analytically perform many of the marginalization steps (integrations) necessary in estimating the posterior distribution for this model. This choice of priors is denoted

$$\begin{aligned} \vec{\pi} | \alpha &\sim \text{Dirichlet}\left(\cdot \,\middle|\, \frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) \\ \Theta &\sim \mathcal{G}_0 \end{aligned} \tag{3}$$

[1] A prior is conjugate if it yields a posterior that is in the same family as the prior (Gelman et al., 1995).

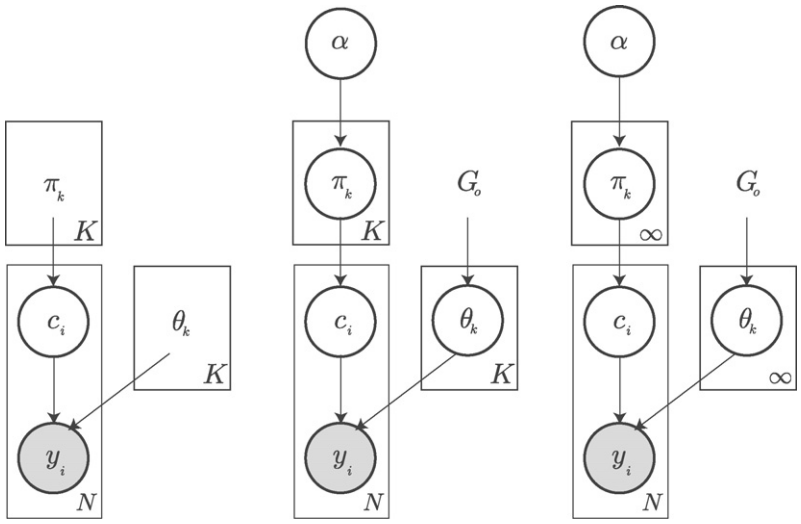

Fig. 2. From left to right: graphical models for a finite Gaussian mixture model (GMM), a Bayesian GMM, and an infinite GMM.

where $\Theta \sim \mathcal{G}_0$ is shorthand for

$$\mathbf{\Sigma}_k \sim \text{inverse Wishart}_{\upsilon_0}(\mathbf{\Lambda}_0^{-1}) \tag{4}$$

$$\vec{\mu}_k \sim \text{Gaussian}\left(\vec{\mu}_0, \frac{\mathbf{\Sigma}_k}{\kappa_0}\right). \tag{5}$$

The formulas for these distributions are widely available, e.g. in Gelman et al. (1995). Here we give a brief overview of the modeling assumptions this particular choice of priors embodies. The Dirichlet prior is used to encode prior knowledge about the number and relative activity of neurons in the recording. In the finite model the number of neurons is assumed to be known and is the same as the length of $\vec{\pi}$. The parameters to the Dirichlet prior is a vector of exactly the same length and, if parameterized uniformly as above, says that all neurons are *a priori* equally likely to have generated every spike. This uniform parameterization is chosen for mathematical convenience when generalizing to unknown K; however, if K is somehow known *a priori* then a more specific parameterization of the prior can be employed. For instance, if it is known that there are three neurons in a particular recording and the most active neuron produced 50% of the total activity in the recording and the other two produced the remaining spikes in nearly equal proportion then the Dirichlet prior should be initialized with some multiple of [.5 .25 .25] (i.e. [500 250 250]). The specific values chosen encode how confident we are in our prior. Each parameter of the Dirichlet prior is a count of pseudo-observations that follow the pattern expressed in the prior. The higher the pseudo-counts the more real data it takes to cause the posterior to deviate from the prior and vice versa. In the uniformly parameterized case we can simply think of α as a hyperparameter that indicates how confident we are that all neurons in the recording are equally active.

The parameters of the inverse Wishart prior, $\mathcal{H} = \{\mathbf{\Lambda}_0^{-1}, \upsilon_0, \vec{\mu}_0, \kappa_0\}$, are used to encode our prior beliefs regarding the action potential waveform shape and variability. For instance $\vec{\mu}_0$ specifies our prior belief about what the mean waveform should look like, where κ_0 is the number of pseudo-observations we are willing to ascribe to our belief (in a way similar to that described above for the Dirichlet prior). The hyperparameters $\mathbf{\Lambda}_0^{-1}$ and υ_0 encode the ways in which waveforms are likely to vary from the mean waveform and how confident we are in our prior beliefs about that.

We will refer to $\mathcal{H}$ as the hyperparameters of our model. In general these will be specified by a user and not estimated from the data. These are not the only choices that one could make for the priors.

A graphical model representing this model is shown in the middle of Fig. 2. This graphical model illustrates the conditional dependency structure of the model. For instance π_k is independent of $\mathbf{y}_i$ given c_i. In this figure circles indicate random variables and boxes (plates) indicate repetition (with the number of repetitions in the lower right). Links between variables indicate the assumed conditional dependency structure of the joint probability distribution. From such a graphical model the joint distribution of the data and model parameters can easily be read out. Here that joint distribution is given by

$$P(\mathcal{Y}, \Theta, \mathcal{C}, \vec{\pi}, \alpha; \mathcal{H}) = \left(\prod_{j=1}^{K} P(\theta_j; \mathcal{H})\right)\left(\prod_{i=1}^{N} P(\vec{y}_i|c_i, \theta_{c_i}) P(c_i|\vec{\pi})\right) P(\vec{\pi}|\alpha)P(\alpha). \tag{6}$$

Applying Bayes rule and conditioning on the observed data we see that posterior distribution is simply proportional to the joint (rewritten slightly):

$$P(\mathcal{C}, \Theta, \vec{\pi}, \alpha|\mathcal{Y}; \mathcal{H}) \propto P(\mathcal{Y}|\mathcal{C}, \Theta)P(\Theta; \mathcal{H})\left(\prod_{i=1}^{N} P(c_i|\vec{\pi})\right) P(\vec{\pi}|\alpha)P(\alpha). \tag{7}$$

where $P(\mathcal{Y}|\mathcal{C}, \Theta) = \prod_{i=1}^{N} P(\vec{y}_i|c_i, \theta_{c_i})$ and $P(\Theta; \mathcal{H}) = \prod_{j=1}^{K} P(\theta_j; \mathcal{H})$.

In Eq. (7) the proportionality sign hides the marginal probability of the data under the model (also called the "evidence") which cannot be computed analytically. In such cases Markov chain Monte Carlo (MCMC) methods such as those reviewed in Neal (1993) can be used to obtain a discrete representation of the posterior by sampling from this unnormalized density. These methods simulate a discrete Markov process whose equilibrium distribution is the posterior distribution. Other posterior inference methods can be used here, particularly the variational approach outlined in Jordan et al. (1999), but in this work we will largely focus on MCMC methods.

Unfortunately we have not yet specified how we are going to get around the problem of needing to know the number of neurons in the recording, K, *a priori*. Since varying the number of neurons in the model will certainly alter spike sorting results and thereby spike train analysis results, we would either like to be extremely certain that the number we choose is correct or ideally we would like to be able to represent our uncertainty about the choice.

Our approach for dealing with an unknown K is "nonparametric Bayesian" modeling. NPB models are characterized by having infinite complexity. In this case that means taking the limit of the model defined above as $K \to \infty$, resulting in a mixture model with an infinite number of components. Of consequence to the spike sorting problem, the posterior distribution in such a NPB mixture model consists of infinite mixture model realizations that vary in the number of realized (non-zero) mixture components. Put another way, given a finite set of observations (spikes), any particular sorting can only have a finite number of non-zero mixture components. Realizations from the infinite mixture (samples) all have a finite number of components but the number and characteristics of these non-zero components may vary.

4. Infinite Gaussian mixture model

We use a NPB model called the infinite Gaussian mixture model which was developed by Rasmussen (2000). Towards understanding the IGMM and how to apply it to the spike sorting problem note that the only terms in Eq. (7) that explicitly depend on K are $P(c_i|\vec{\pi})$ and $P(\vec{\pi}|\alpha)$ as both involve $\vec{\pi}$ whose dimensionality is K. Of course the likelihood $P(\mathcal{Y}|\mathcal{C}, \Theta)$ and the class parameter prior $P(\Theta; \mathcal{H})$ also implicitly depend on K as it is the number of hidden classes. This dependence on K is indirect however as both the likelihood and the class parameter prior only need to be evaluated for realized hidden classes (i.e. hidden classes to which at least one observation (spike) is attributed).

The IGMM arises as $K \to \infty$. While it is possible to work directly with such infinite dimensional models it is easier to consider the limit of a Bayesian mixture model having integrated $\vec{\pi}$ out. So, operating on $P(c_i|\vec{\pi})$ and $P(\vec{\pi}|\alpha)$, recognize that $\vec{\pi}$ can be marginalized out because the Dirichlet prior is conjugate to the discrete (multinomial) likelihood:

$$\begin{aligned} P(\mathcal{C}_{|\alpha}) &= \int \mathrm{d}\vec{\pi} \prod_{i=1}^{N} P(c_i|\vec{\pi}) P(\vec{\pi}|\alpha) \\ &= \frac{\prod_{k=1}^{K} \Gamma(m_k + (\alpha/K))}{\Gamma(\alpha/K)^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}. \end{aligned} \tag{8}$$

This expression is the joint probability of a *single* labeling of all observations. Notice that permutating the labels assigned to sets of observations does not change structure of the labeling, i.e. only the labels are switched around. Because of this we would prefer a model that expresses the probability of partitions of the data rather than a specific labelings per se. A partitioning of the data is a grouping of the data into nonoverlapping subsets, each being sets of spikes attributed to a neuron. The number of different ways of applying K labels to a single partitioning of the data with $K_+ < K$ bins is $K!/(K-K_+)!$. As every one of these labelings has the same marginal probability under Eq. (8) the total probability of any one partitioning of the data is

$$P(\mathcal{C}|\alpha) = \frac{K!}{(K-K_+)!} \frac{\prod_{k=1}^{K} \Gamma(m_k + (\alpha/K))}{\Gamma(\alpha/K)^K} \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}. \tag{9}$$

where $m_k = \sum_{i=1}^{N} \mathbb{I}(c_i = k)$ is the number of items in class k ($\mathbb{I}(\,)$ is the indicator function) and $\Gamma(\,)$ is the Gamma function with $\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$. Here K_+ is the number of non-empty bins in the partitioning; this being equivalent to the number of neurons that generated at least one action potential waveform in the recording.

It is the limit of this expression as $K \to \infty$ that turns the Bayesian GMM into the IGMM (Rasmussen, 2000). The limiting expression (taken from Griffiths and Ghahramani, 2005) is given here without derivation:

$$P(\mathcal{C}|\alpha) = \alpha^{K_+} \left(\prod_{k=1}^{K_+} (m_k - 1)! \right) \frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}. \tag{10}$$

Given Eqs. (7) and (10) one can accomplish model estimation by sampling. In fact Gibbs sampling, as developed in general by Geman and Geman (1984), is possible in this model. A Gibbs sampler for this model requires an expression for the conditional distribution of a single class label given the value of all of the others. This conditional distribution is given here

$$P(c_i = k|\mathcal{C}_{-i}) = \begin{cases} \dfrac{m_k}{i-1+\alpha} & k \le K_+ \\ \dfrac{\alpha}{i-1+\alpha} & k > K_+ \end{cases}. \tag{11}$$

These equations characterize a process known as the Chinese restaurant process (CRP) because of a story that illustrates the process (Pitman, 2002). Repeated here, the story describes a method of sequentially seating customers stochastically at tables in a Chinese restaurant with an infinite number of tables: the first customer enters the restaurant and is seated at the first table. Subsequent customers enter the restaurant and are stochastically seated according to the number of people already sitting at each table. If there are many people already sitting at a table (i.e. m_k, the number of customers sitting at table k is large) then the probability of being seated at that table is high and vice versa. However, there is always a small probability of being seated at a table that has no current occupants; $\alpha/(i-1+\alpha)$ where i is the number of the current customer being seated. Seating arrangements drawn from this seating process are distributed according to the density given in Eq. (10).

The language of Chinese restaurants, tables, and customers can be directly converted into the language of spike sorting. In spike sorting the restaurant is a single recording, each table is a neuron, and each customer is an action potential waveform. Bearing in mind that this is only the *prior* over assignments of spikes to neurons, and does not take into account the likelihood at all, the Chinese restaurant process story can be used to describe our model for how spikes are attributed to neurons. Spikes enter the recording one at a time and are attributed to a neuron according to how many spikes that neuron has already emitted. Neurons that have been responsible for many spikes are more likely to have new spikes attributed to them.

In the infinite Gaussian mixture model the generative view of this prior is that the class identifiers for all N datapoints are first generated via the Chinese restaurant process. Depending on α this will yield some stochastic number of classes $K_+ < N$. Observations from each class are then are generated from each of K_+ Gaussian densities whose parameters are each drawn independently from the multivariate normal inverse Wishart (MVN-IW) prior.

Given this nonparametric Bayesian prior for our spike sorting model we need now to perform posterior estimation; that is, determine a distribution over classes and assignments given the observed spike waveforms. Here we review two different samplers for doing so. The first corresponds to algorithm "three" in Neal (1998), which itself derives from the work of Bush and MacEachern (1996), West et al. (1994), MacEachern and Muller (1998), and Neal (1992). The second is a sequential posterior estimation algorithm that corresponds to the techniques in Fearnhead and Clifford (2003), Fearnhead (2004), Sanborn et al. (2006) and is related to Wood and Griffiths (2007). The first we call the collapsed Gibbs sampler (following Liu (2001)) because the latent class parameters are integrated out and the sampler state consists of only the

class identifiers. In the language of spike sorting this means that we integrate out the waveform means and covariances. It is an offline or batch sampling algorithm requiring that the entire dataset be present. In contrast, the second algorithm is a sequential posterior estimation algorithm, meaning that each observation (spike) is processed once in sequence when estimating the posterior. This latter type of model estimation makes online spike sorting possible.

4.1. Collapsed Gibbs sampler

Remember that our goal is build an estimate of the posterior distribution for the infinite Gaussian mixture model by using MCMC to draw samples from the posterior; that is, we want to learn a distribution over spike sortings where both the number of units and the assignments of spikes to units can vary.

Having chosen the multivariate normal inverse Wishart prior for the multivariate normal class distributions makes it possible to analytically integrate these parameters out yielding an expression for a posterior over only $\mathcal{C}$:

$$\begin{aligned} P(\mathcal{C}|\mathcal{Y};\mathcal{H}) &= \int \mathrm{d}\Theta P(\mathcal{C},\Theta|\mathcal{Y};\mathcal{H}) \\ &\propto P(\mathcal{C};\mathcal{H}) \int \mathrm{d}\Theta P(\mathcal{Y}|\mathcal{C},\Theta;\mathcal{H}) P(\Theta;\mathcal{H}). \end{aligned} \tag{12}$$

Remember that $\mathcal{C}$ are variables that indicate which neuron each spike came from and $\mathcal{Y}$ are the spike waveforms themselves.

Algorithm 1. Collapsed Gibbs sampler for the IGMM

$c_1, \ldots c_N \sim$ uniform-random-integer(1..N)
$\mathcal{C}_0 \leftarrow \{c_1, \ldots c_N\}$
$K_+ = 0$
for $s = 1 : \#$ samples **do**
  $\mathcal{C}_s \leftarrow \mathcal{C}_{s-1}$
  for $i = 1 : N$ **do**
    $m_{-i} \leftarrow \sum_{j=1}^{N} \mathbb{I}(c_j = c_i) - 1$
    if $m_{-i} = 0$ **then**
      $c_j = c_j - 1 \; \forall \; j > i$
      $K_+ = K_+ - 1$
    end if
    sample $c_i \sim P(\cdot|\mathcal{C}_{-i}, \mathcal{Y}, \ldots)$ // *Using Eqn's: 13 and 15*
    if $c_i > K_+$ **then**
      $K_+ = K_+ + 1$
    end if
  end for
  sample α using a Metropolis or Gibbs step
end for

The advantage of analytically marginalizing out these parameters is that the state space of the sampler is reduced. Typically this leads to faster convergence of the sampler to the equilibrium distribution (Liu, 2001).

For the collapsed Gibbs sampler in Algorithm 1 the sampler state consists of $\mathcal{C}$ and α. The updates for the class labels are

$$\begin{aligned} P(c_i = j|\mathcal{C}_{-i},\mathcal{Y},\alpha;\mathcal{H}) &\propto P(\mathcal{Y}|\mathcal{C};\mathcal{H})P(\mathcal{C}|\alpha) \\ &\propto \prod_{j=1}^{K_+} P(\mathcal{Y}^{(j)};\mathcal{H}) P(c_i = j|\mathcal{C}_{-i},\alpha) \\ &\propto P(\mathcal{Y}^{(j)};\mathcal{H}) P(c_i = j|\mathcal{C}_{-i},\alpha) \\ &\propto P(y_i|\mathcal{Y}^{(j)} \setminus y_i;\mathcal{H}) P(c_i = j|\mathcal{C}_{-i},\alpha) \end{aligned} \tag{13}$$

where $\mathcal{Y}^{(j)} \setminus y_i$ is the set of observations currently assigned to class j except y_i (y_i is "removed" from the class to which it belongs when sampling). Because of our choice of conjugate prior we know that $P(y_i|\mathcal{Y}^{(j)} \setminus y_i;\mathcal{H})$ is multivariate Student-t (Gelman et al., 1995, p. 88):

$$y_i|\mathcal{Y}^{(j)} \setminus y_i;\mathcal{H} \sim t_{\nu_n - D + 1}(\vec{\mu}_n, \mathbf{\Lambda}_n(\kappa_n + 1)/(\kappa_n(\nu_n - D + 1))) \tag{14}$$

where

$$\begin{aligned} \vec{\mu}_n &= \frac{\kappa_0}{\kappa_0 + N}\vec{\mu}_0 + \frac{N}{\kappa_0 + N}\bar{y} \\ \kappa_n &= \kappa_0 + N \\ \nu_n &= \nu_0 + N \\ \mathbf{\Lambda}_n &= \mathbf{\Lambda}_0 + \mathbf{S} + \frac{\kappa_0 n}{\kappa_0 + N}(\bar{y} - \vec{\mu}_0)(\bar{y} - \vec{\mu}_0)^{\mathrm{T}} \end{aligned}$$

and D is the dimensionality of y_i ($D = 2$ or 3 in our case). The subscript $\nu_n - D + 1$ indicates the number of degrees of freedom of the multivariate Student-t distribution.

In Eq. (11) there are two cases. The first case corresponds to attributing a spike to one of the units that has already been identified in the recording. The second case corresponds to positing a new unit and attributing the spike to it. This means that Eq. (13) splits into two cases as well.

We also must consider the probability of starting a new cluster (unit). This is given by

$$P(c_i > K_+|\mathcal{C}_{-i},\mathcal{Y},\alpha;\mathcal{H}) \propto P(y_i;\mathcal{H}) P(c_i > K_+|\mathcal{C}_{-i},\alpha). \tag{15}$$

In positing a new cluster, $P(y_i;\mathcal{H})$ has the same form as $P(y_i|\mathcal{Y}^{(j)} \setminus y_i;\mathcal{H})$ above; however, as there are no other observations the original hyperparameters are used instead, i.e.

$$y_i;\mathcal{H} \sim t_{\nu_0 - D + 1}\frac{\vec{\mu}_0, \mathbf{\Lambda}_0(\kappa_0 + 1)}{\kappa_0(\nu_0 - D + 1)}. \tag{16}$$

For ease of reference the multivariate Student-t density function is given here

$$\begin{aligned} P(y) &= t_\nu(y|\mu, \mathbf{W}) \\ &= \frac{\Gamma((\nu + D)/2)}{\Gamma(\nu/2)\nu^{D/2}\pi^{D/2}} |\mathbf{W}|^{1/2} \left(1 + \frac{1}{\nu}(y - \mu)^{\mathrm{T}}\mathbf{W}^{-1}(y - \mu)\right)^{-(\nu+D)/2}. \end{aligned}$$

4.2. Sequential posterior estimation

An appealing property of the nonparametric mixture modeling approach to spike sorting and spike train modeling is that sequential posterior estimation is possible. In spike sorting, sequential posterior estimation means that estimation of the posterior distribution is accomplished by processing each spike once in sequence. In the IGMM this is possible because the prior on $\mathcal{C}$ yields an analytic expression for the conditional distribution of the label of a single observation given the labels for all of the others, $P(c_i|\mathcal{C}_{-i})$.

To explain this, recall that $\vec{y}_i$ is the ith observation, and let $\mathcal{Y}^{(1:i)}$ be all observations up to and including the ith (similarly for the class identifiers $\mathcal{C}^{(1:i)}$). Note that this superscript notation means something different than it did in the previous section. Also remember that it is easy to sample from $P(c_i|\mathcal{C}^{(1:i-1)})$; to do so, sample the ith class identifier given the first $i - 1$ identifiers directly from the prior (using Eq. (11)). Applying Bayes' rule we write the unnormalized posterior of $\mathcal{C}^{(1:i)}$ given $\mathcal{Y}^{(1:i)}$ in the following way:

$$\hat{P}(\mathcal{C}^{(1:i)}|\mathcal{Y}^{(1:i)}) \propto P(\vec{y}_i|\mathcal{C}^{(1:i)}, \mathcal{Y}^{(1:i-1)}) P(c_i|\mathcal{Y}^{(1:i-1)}).$$

Since we can evaluate $P(\vec{y}_i|\mathcal{C}^{(1:i)}, \mathcal{Y}^{(1:i-1)})$, we can obtain weighted samples from the normalized posterior $P(\mathcal{C}^{(1:i)}|\mathcal{Y}^{(1:i)})$ using impor-

tance sampling with a proposal distribution of

$$P(c_i|\mathcal{Y}^{(1:i-1)}) = \sum_{\substack{\Delta\mathcal{C}^{(1:i-1)} \\ L}} P(c_i|\mathcal{C}^{(1:i-1)})P(\mathcal{C}^{(1:i-1)}|\mathcal{Y}^{(1:i-1)}) \approx \sum_{l=1} w_{\{l\}}^{(1:i-1)} P(c_i|\mathcal{C}_{\{l\}}^{(1:i-1)})$$

where $w_{\{l\}}^{(1:i)}$ is the weight for the lth sample having integrated i observations into the model, $\mathcal{C}_{\{l\}}^{(1:i)}$ is the corresponding sample and $\Delta\mathcal{C}^{(1:i-1)}$ is shorthand for all possible assignments of labels 1 through $i-1$. "Particle filtering" is the name given to sequential density estimation approaches of this form. In a particle filter, a set of L weighted "particles" (the samples and weights) are used to form a discrete representation of the distribution of interest (here the posterior distribution over class identifiers given observations):

$$\{w_{\{l\}}^{(1:i-1)}, \mathcal{C}_{\{l\}}^{(1:i-1)}\} \approx P(\mathcal{C}^{(1:i-1)}|\mathcal{Y}^{(1:i-1)}).$$

These particles are updated once per observation in sequence. The discrete particle representation of the posterior distribution is an approximation in the traditional Monte Carlo integration sense. As the number of particles goes to infinity, averages over the discrete representation converge to expectations of the true distribution; that is, for some function $g(\,)$:

$$\lim_{L\to\infty}\sum_{l=1}^{L} w_{\{l\}}^{(1:i-1)} g(\mathcal{C}_{\{l\}}^{(1:i-1)}) = E_{P(\mathcal{C}^{(1:i-1)}|\mathcal{Y}^{(1:i-1)})}[g] = \sum_{\Delta\mathcal{C}^{(1:i-1)}} P(\mathcal{C}^{(1:i-1)}|\mathcal{Y}^{(1:i-1)}) g(\mathcal{C}^{(1:i-1)}). \quad (17)$$

Algorithm 2. Particle filter posterior estimation for the IGMM

1: $\mathcal{C}_{\{:\}}^{(1)} \leftarrow 1$
2: **for** $i = 2 : N$ **do**
3: **for** $l = 1 : L$ **do**
4: sample $\mathcal{C}_{\{l\}}^{(i)} \sim P(\mathcal{C}_{\{l\}}^{(i)} = k|\mathcal{C}_{\{l\}}^{(1:i-1)})$ // *Using Eqn. 11*
5: calculate weight $\omega_{\{l\}}^{(i)} \propto P(\mathcal{Y}^{(i)}|\mathcal{C}_{\{l\}}^{(1:i)}, \mathcal{Y}^{(1:i-1)}, \ldots)$ // *Using Eqn.'s 13 and 15*
6: **end for**
7: normalize the weights $\omega_{\{l\}}^{(i)} \leftarrow \omega_{\{l\}}^{(i)} / \sum_{j=}^{L} \omega_{\{j\}}^{(i)} \; \forall\, l$
8: $\mathcal{C}_{\{:\}}^{(i)} \leftarrow \text{resample}(\omega_{\{:\}}^{(i)}, \mathcal{C}_{\{:\}}^{(i)})$
9: **end for**

Referring to Algorithm 2 and noting that $P(\bar{y}_i|\mathcal{C}^{(1:i)}, \mathcal{Y}^{(1:i-1)})$ is given by Eq. (14) for existing clusters and Eq. (16) for putative new clusters this completes the mathematical description of the basic particle filter.

The resampling step used in sequential importance resampling particle filters (Doucet et al., 2001) (the function "resample" in Algorithm 2) helps but does not completely alleviate a problem with particle filtering for Dirichlet process mixture models (Fearnhead and Clifford, 2003) known as weight degeneracy. To get around this weight degeneracy problem we follow Fearnhead and Clifford (2003) in adopting a resampling strategy that ensures a well distributed particle set. In Fearnhead and Clifford (2003) we are reminded that these two particle sets are equivalent

$$\frac{1}{3}, \{1, 2, 1\}; \frac{1}{3}, \{1, 1, 1\}; \frac{1}{3}, \{1, 1, 1\}$$

and

$$\frac{1}{3}, \{1, 2, 1\}; \frac{2}{3}, \{1, 1, 1\}$$

where here the fraction represents the weight given to each particle and the three integers in the brackets are the particles which themselves indicate the class labels given to three hypothetical observations. Clearly, the second representation is more efficient in that it requires fewer particles to represent the distribution with the same accuracy. This efficiency is merely a product of the measure being weighted rather than unweighted.

While the details of the exact methodology are available in Fearnhead and Clifford (2003), we give here a brief accounting for IGMM's. In the step in which the class label for the ith observation is sampled from the CRP prior we need not restrict ourselves to sampling only a single new class label from the prior. In this model it is possible to exhaustively enumerate all possible values of $c_i|\mathcal{C}_{-i}$ (i.e. no sampling from the prior, merely enumerating all possible next labels). Doing this generates a particle set of size M from the original N particles where $M > N$, with each particle replicated as many times as the number of pre-existing unique labels in the particle plus one to account for the possibility of the new observation having been generated by an as yet unseen latent class. Weights for each of these particles can be computed as before using Eqs. (13) and (15), only now computing M instead of N weights. Having increased the diversity and cardinality of our particle set we now must down-sample the representation to avoid exponential growth in the number of particles. One can resample N particles directly from this discrete distribution; however in Fearnhead and Clifford (2003) a resampling algorithm is developed that is provably optimal in terms of preserving the expected Monte Carlo integration error in expectations using the reduced particle set. Additionally, their approach limits the number of times a particle can be selected in the down-sampling step to one which yields an optimally diverse particle set. A consequence of their approach is that the particle set is no longer unweighted after resampling like it is in regular particle filtering.

In all of our experiments where we report results for particle filter posterior estimation the approach of Fearnhead and Clifford (2003) is used to resample the particle set.

5. Experiments

Having introduced the IGMM for spike sorting in the previous section it remains to demonstrate its utility in performing neural data analysis. In particular, we show how this model represents *uncertainty* in the assignment of waveforms to units and demonstrate how this uncertainty can be incorporated into the analysis of neural signals. In particular we use our NPB spike train model to study the encoding properties of motor cortical neurons. Our aim is to show how the IGMM posterior distribution over spike trains allows us to analyze neural data in a new way, potentially utilize data that might otherwise not be utilized, and to express (un)certainty in our conclusions.

We start by describing two primate motor cortical datasets analyzed here. Both were recorded from the same monkey but on two different dates over a year apart. The tasks performed by the monkey on each occasion were different; the first was the "pursuit tracking" task described by Wu et al. (2005) and the second was the "pinball" task described by Serruya et al. (2003). In the discussion and results these datasets will be distinguished by task name.

Both the pursuit tracking task and the pinball task are voluntary arm movement tasks. In the pursuit tracking task the objective was to track a continuously moving target on a computer monitor by moving a manipulandum on a 2D surface. Both the target

and a cursor representing the current manipulandum location were shown to the monkey. In the pinball task the task objective was to acquire (move a cursor onto) a sequence of targets that appeared sequentially at random points in the space reachable by the manipulandum. This can be seen as a generalization of the center-out reaching task commonly utilized in the motor control literature. The pursuit tracking dataset consisted of 10 trials each containing approximately 5–10 s of movement. The pinball task dataset consisted of 50 trials each consisting of approximately the same movement duration.

For both tasks the firing activity of a population of cells was recorded from a chronically implanted microelectrode array manufactured by Cyberkinetic Neurotechnology Systems Inc. (2005) from which six electrodes were selected for analysis. The inter-electrode spacing in these microelectrode arrays is large enough that each recording electrode is independent. Threshold crossing

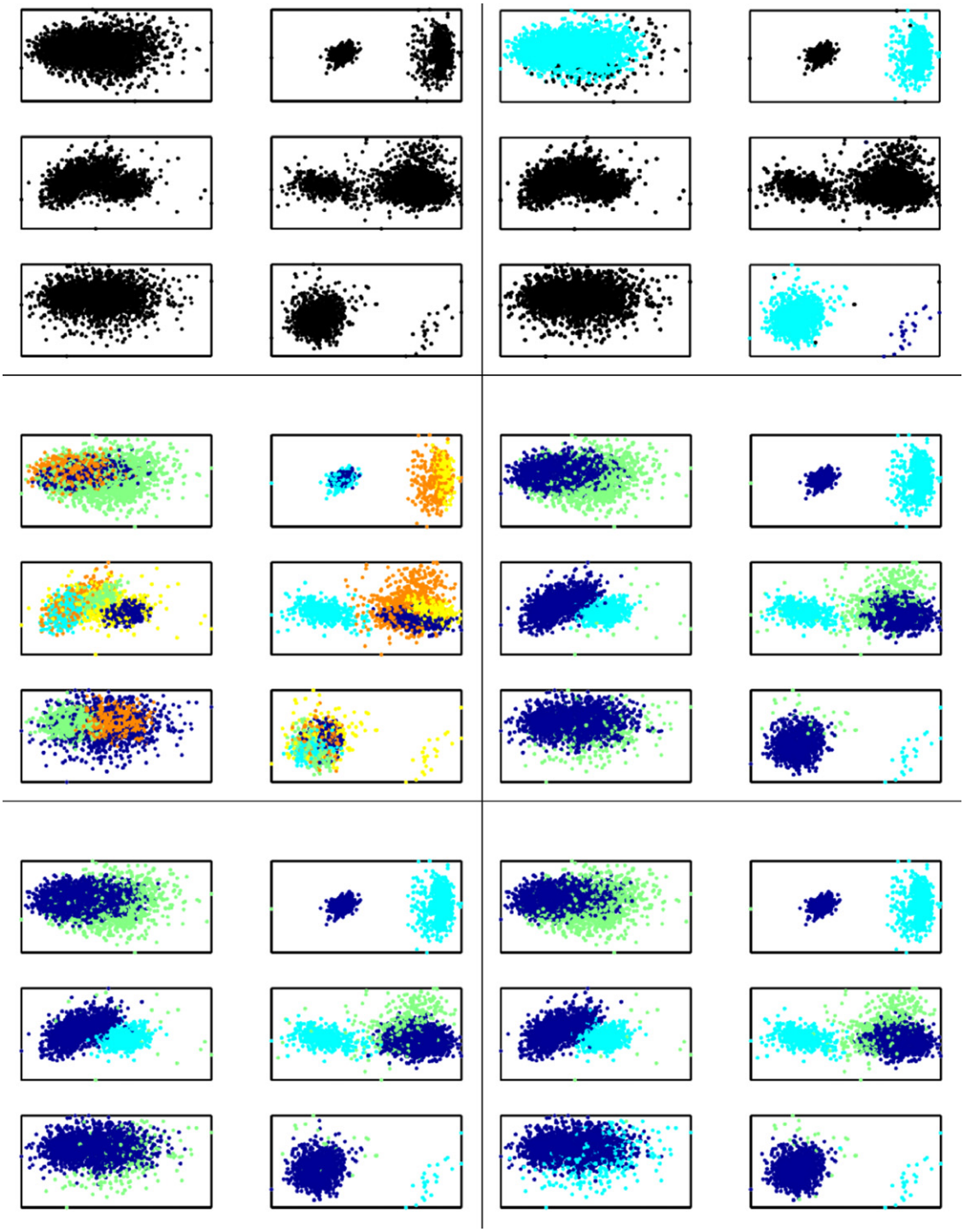

Fig. 3. Results from sorting six channels of the pursuit tracking neural data using different techniques. Projections of waveforms from channels 1 to 6 onto the first two principal component directions are shown in each of the six panels. The top left panel shows the unsorted waveforms, the top right shows a manual labeling (the black points are not assigned to any neuron), the left panel in the second row shows a maximum likelihood labeling, and the remaining three are samples from the IGMM posterior.

events (waveforms) were saved on all channels along with the time at which they occurred. The thresholds were empirically set by the experimenter at the time of recording to exclude non-neural waveforms from being captured. Following application of the waveform dimensionality reduction procedure given above to each channel individually, the waveforms from the pursuit tracking task were projected onto the first two PCA basis vectors accounting for on average 66% of the waveform variance (51%, 87%, 69%, 78%, 58%, and 55% per channel, respectively, all rounded to the nearest percent). The waveforms for the pinball data were projected onto the first three PCA basis vectors accounting for on average 56% of the total waveform variance (59%, 72%, 52%, 54%, 54%, and 47% per channel).

Both datasets were sorted three ways: (1) expectation maximization with finite Gaussian mixture models and Bayesian information criterium model selection (we refer to this as the maximum likelihood approach), (2) IGMM with Gibbs sampling, and (3) with sequential (particle filter) posterior estimation. The hyperparameters chosen for the IGMM were the same for both datasets, $\mu_0 = [0\ \ 0]$, $\kappa_0 = .2$, $\mathbf{\Lambda}_0 = \text{diag}(.1)$, and $\nu_0 = 20$ (for the pinball data

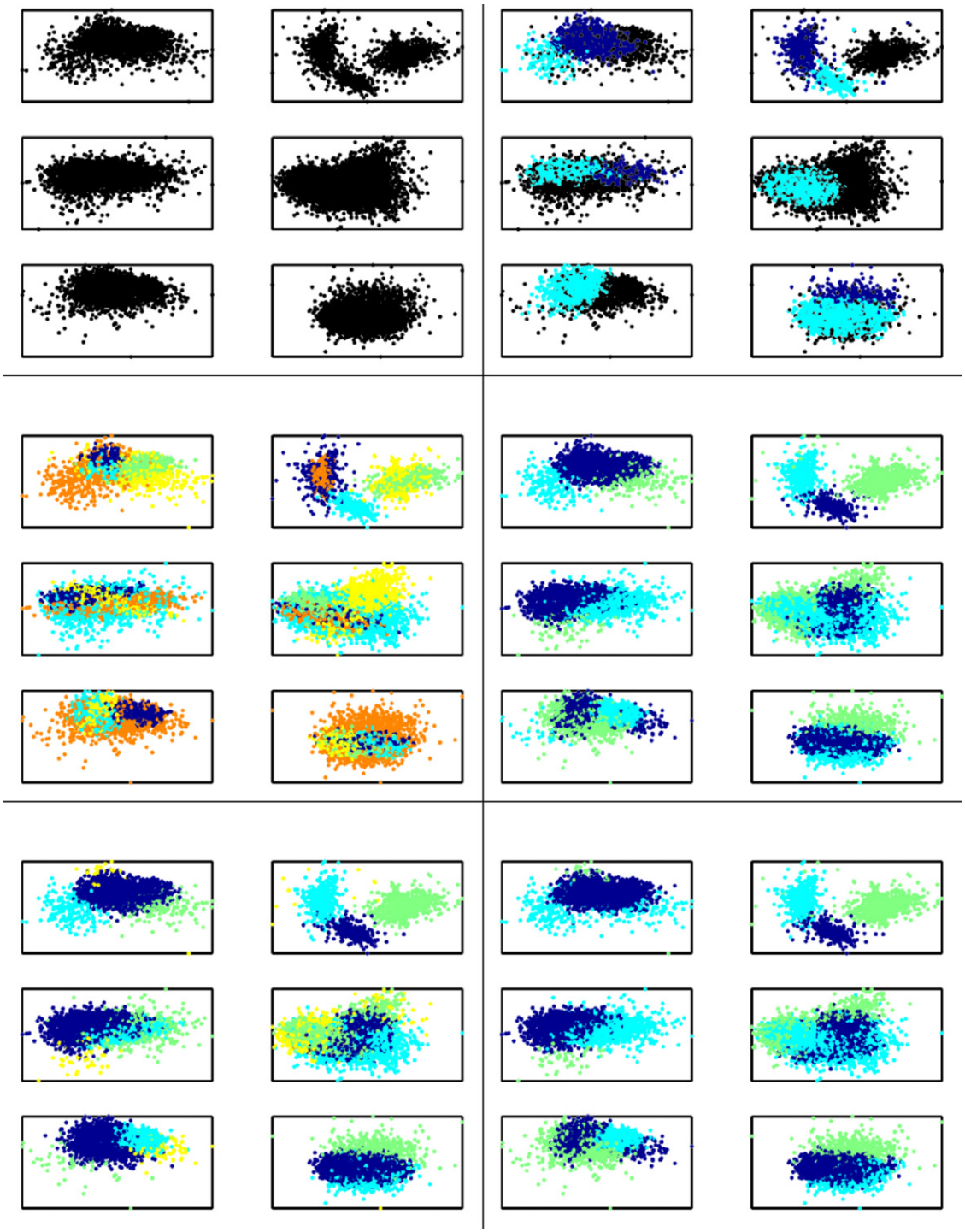

Fig. 4. Results from sorting six channels of the pursuit tracking neural data using different techniques. See the caption of Fig. 3.

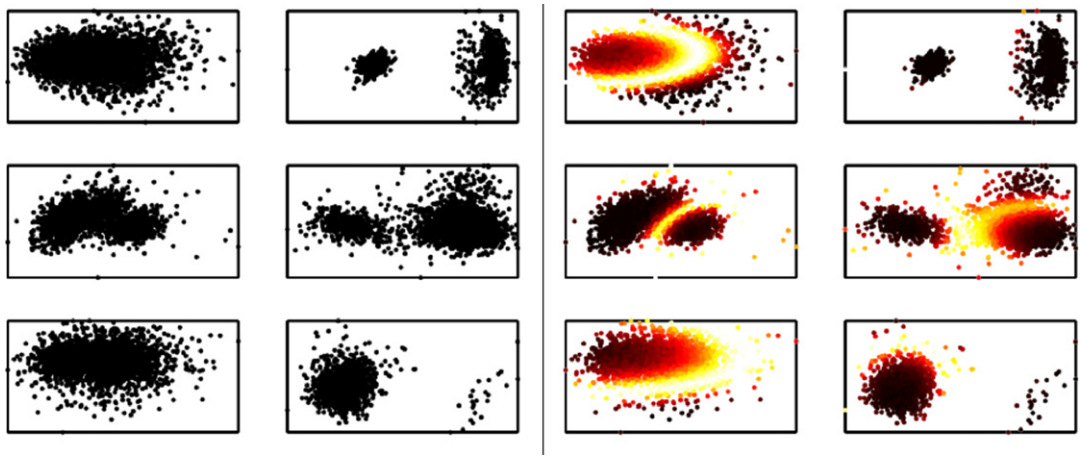

Fig. 5. Uncertainty in the sorting of the pursuit tracking neural data. Channels 1–6 are shown in both panels. On the left is the unsorted data, on the right the same points are plotted but with colors that indicate how uncertain the cluster label for that point is (the entropy of the conditional label distribution over labels for that point). Black means certain; red, orange, and yellow through white mean more uncertain.

μ_0 and $\mathbf{\Lambda}_0$ were appropriately sized for three-dimensional observations).

Sorting results for the pursuit tracking data are shown in Fig. 3 and for the pinball data in Fig. 4. Both of these figures are divided into six sections by thin black lines. We refer to these sections as panels. The upper left panel in both figures shows the six selected channels before spike sorting. Each dot is the projection of one waveform onto the first two principal components. The upper right panel shows a manual labeling of the data. In this panel the color black is significant as it indicates waveforms that were identified by the human sorter as being too ambiguous to sort. The left panel in the second row is the maximum likelihood solution. The remaining three panels illustrate the posterior distribution over labelings induced by the IGMM. Each of these three panels is *one sample* from the IGMM posterior, where each sample represents one complete sorting of the data. Looking closely one can see that many of the individual waveform labels change from one sample to another as both new and different neurons are posited and individual waveforms are attributed.

We also provide plots that illustrate how uncertainty is represented by this model in Figs. 5 and 6. Both figures redisplay the unsorted (unlabeled) data for ease of reference along with a plot that illustrates the relative (to the dataset) uncertainty of the label given to each datapoint. This plot was produced by computing the entropy of the marginal distribution over class identity at every point. The higher the entropy of a point's labeling the hotter the color assigned (black → red → orange → yellow→ white). Points that are nearly white have label distributions that have high entropy which means that the cluster to which they are attributed is uncertain under the model.

To evaluate the applicability of the model for neural data analysis we use it to investigate movement direction tuning properties of motor cortical neurons. We show how our model helps us to use all of the available data and allows us to display our confidence in our findings as well.

In Georgopoulos et al. (1982) it was established that neural firing rates in primate motor cortex change in response to the direction of hand movement in two-dimensional movement tasks. The direction that evokes a cell's maximum firing rate is called its preferred direction. Furthermore they found that the firing rate of a cell is higher for movement in directions near a cell's preferred direction and falls off smoothly as a function of the difference between the direction of movement and the cell's preferred direction. It was shown that a cosine function (cosine tuning curve) fit the relationship between average firing rate and movement direction well. The preferred direction for a cell may be established by cosine regression and differentiation.

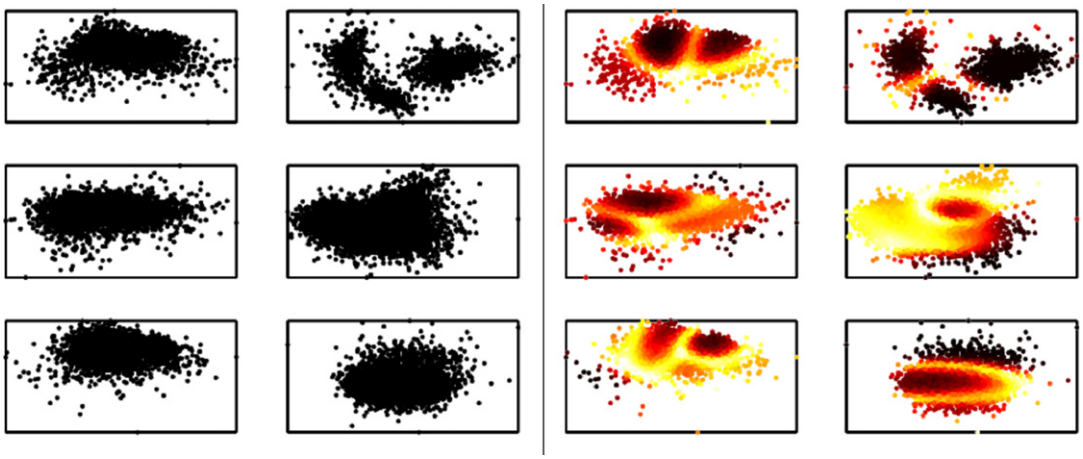

Fig. 6. Uncertainty in the sorting of six channels of pinball neural data. See the caption of Fig. 5.

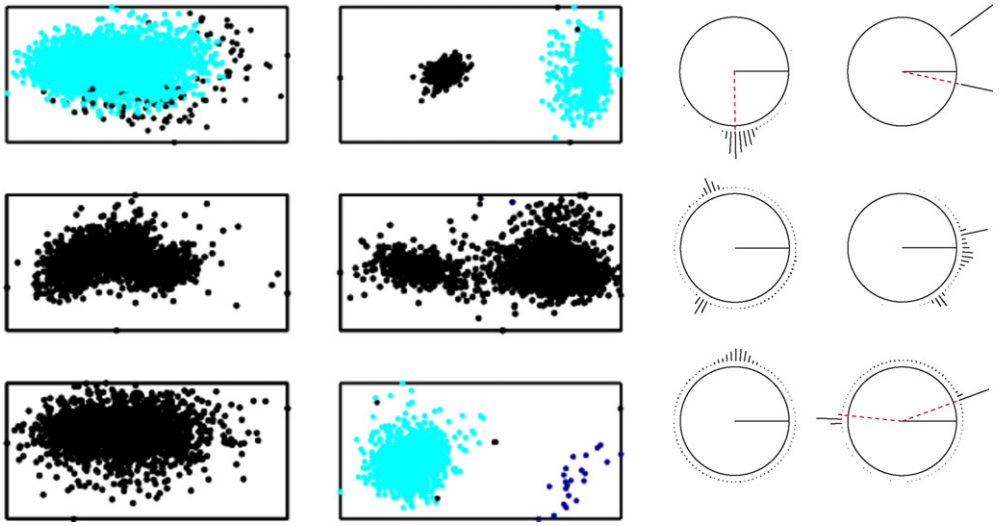

Fig. 7. On the left: human sorted waveforms from the pursuit tracking task. On the right: preferred direction distributions for the pursuit tracking channels. The solid black line in the circle indicates movement to the right. The dashed red lines indicate the preferred direction of cells identified in the manually sorted data. The radial ticks outside of the circle are the normalized histogram of cell preferred direction counts from the IGMM posterior.

The right panel of Fig. 7 shows a visualization of the distribution over preferred directions for the pursuit tracking neural data. Both panels of Fig. 7 have the same row and column ordering as the panels in Fig. 3. The red dashed lines indicate the estimated preferred directions for the manual labeling. The solid black lines of varying length protruding like spines out from the circle are a visualization of a marginal of the estimated IGMM posterior distribution. The length of each of these lines is proportional to the normalized count of the number of times a cell with a given preferred direction was found in the posterior distribution.

To be more concrete, remember that the posterior distribution of the IGMM can be thought of as a collection of different sortings. For each sorting the preferred directions for all posited cells were determined. The normalized count of the number of times a cell with a given preferred direction was found was computed by dividing the total number of times a cell with a particular preferred direction was posited by the number of cells identified in all of the posterior samples (sortings). This we call the marginal distribution over cell preferred directions. It indicates the level of certainty one can have in concluding that a cell with a certain preferred direction exists on the channel.

Remember that the red dotted lines indicate the preferred directions of the neurons identified by the human sorter. In the right panel of Fig. 7 the middle row and first column of the third row have no red dotted lines. This is because the human sorter did not have enough confidence in the separability of the cell to include it in subsequent analyses. The IGMM spike train model, on the other hand, reveals evidence that there are directionally tuned cells on each of these channels. Moreover, the spread of these marginals provides evidence for how well the detected units fit the cosine tuning model. In some cases the tuning is very precise, indicated by a tight peak, and in others it is much more diffuse. Knowledge of this may be useful for decoding algorithms that exploit cosine tuning.

We note from this that (1) the distribution of preferred directions include the directions found in the human sorted data, (2) several channels suggest clear tuning where the human found the waveforms too uncertain to sort, and (3) several channels contained evidence for additional putative tuned units beyond those found by the human.

6. Discussion

We have presented a new way of thinking about spike sorting and uncertainty in spike sorting. Rather than seek a single best spike sorting, we represent a distribution over sortings and apply this to the analysis of neural signals. The spike sorting results and example analysis from the previous section suggest that our NPB approach will aid in neural data analysis by virtue of modeling some types of spike sorting uncertainty. Although we show only a simple example analysis, it should be clear that the extra information about spike sorting uncertainty captured by our model would be of use in many other types of neural data analysis. It is worth noting that this analysis is fully automatic.

There are a number of ways in which our model can be extended and improved. One extension to consider is directly modeling the raw voltage trace. Because our model does not extend to spike detection, clearly not all variability arising in the spike sorting process can be captured by our model. Additionally, as we do not model overlapping spikes, it would be desirable to extend it by integrating the work of Görür et al. (2004) on mixtures of factor analyzers for modeling overlapping waveforms. Our model also does not explicitly exclude spikes from occurring in the refractory period, i.e. two spikes can be generated by the same "neuron" under our model even if the time between them is less than a few milliseconds. Extending this model to exclude refractory period violations is also feasible. Likewise our model can be extended to account for changing spike waveshape.

Lastly, for long-term chronic spike sorting applications such as unattended neuroprostheses, disappearance and appearance of classes should be modeled. In its current form the model cannot handle disappearance and appearance of new neurons. This is possible to do by leveraging the dependent Dirichlet process work of Srebro and Roweis (2005) and Griffin and Steel (2006). Doing so will make application of nonparametric spike sorting models to chronic spike sorting applications realistic.

While these avenues for improvement are important to pursue as the applicability of this particular nonparametric spike sorting approach is limited to a subset of neural data analyses (for instance, tests for excess synchrony would be an inappropriate use of our model because in its current form overlapping waveforms are not detected), it still remains that this IGMM approach to spike sorting

is a significant step towards accounting for spike sorting uncertainty in analyses of neural data. While our approach is similar in this respect to Nguyen et al. (2003), they did not demonstrate the utility of the full posterior distribution. Furthermore, the IGMM posterior estimation approach is simpler to implement and admits a sequential estimation variant.

Beyond accounting for spike train variability arising from spike sorting uncertainties, particle filter posterior estimation makes sophisticated online spike sorting possible, a long time goal of the community. Looking towards a future in which online IGMM spike sorting is computationally practical (specialized software and/or dedicated hardware would be necessary to achieve this now) reveals a number of possible novel spike sorting applications. For instance, in the case of a chronically implanted neural prosthesis an online spike sorter could run for the duration of the implant, sorting everything as it is recorded, dramatically decreasing the amount of human time necessary to prepare and maintain the prosthesis.

Acknowledgments

This work was supported by the Gatsby Charitable Foundation, the Office of Naval Research (N00014-07-1-0803 and N0014-06-0185), the National Science Foundation (IIS-0534858) and the NSF/NIH Collaborative Research in Computational Neuroscience Program (NIH–NINDS R01 NS 50967-01). We also thank the European Neurobotics Program (FP6-IST-001917). Data for this research was provided by the Donoghue Laboratory at Brown University; in particular we thank Matthew Fellows, Wilson Truccolo, and John Donoghue for their assistance.

References

Bush CA, MacEachern SN. A semi-parametric Bayesian model for randomized block designs. Biometrika 1996;83:275–86.
Cyberkinetic Neurotechnology Systems Inc., 2005. http://www.cyberkineticsinc.com/.
Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B 1977;39:1–38.
Doucet A, de Freitas N, Gordon N. Sequential Monte Carlo Methods in Practice. Springer; 2001.
Fearnhead P. Particle filters for mixture models with an unknown number of components. Journal of Statistics and Computing 2004;14:11–21.
Fearnhead P, Clifford P. Online inference for well-log data. Journal of the Royal Statistics Society Series B 2003;65:887–99.
Fee MS, Mitra PP, Kleinfeld D. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability. J Neuroscience Methods 1996;69:175–88.
Fraley C, Raftery AE. Bayesian regularization for normal mixture estimation and model-based clustering. Tech. Rep. 05/486. Seattle, WA: Department of Statistics, University of Washington; 2005.
Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis. New York: Chapman & Hall; 1995.
Geman S, Geman D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 1984;6:721–41.
Georgopoulos A, Kalaska J, Caminiti R, Massey J. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. Journal of Neuroscience 1982;2:1527–37.
Görür D, Rasmussen CR, Tolias AS, Sinz F, Logothetis N. Modeling spikes with mixtures of factor analyzers. In: Proceeding of the DAGM symposium. Springer; 2004. p. 391–8.
Griffin J, Steel M. Order-based dependent Dirichlet processes. Journal of the American Association of Statistics 2006;101(473):179–94.
Griffiths TL, Ghahramani Z. Infinite latent feature models and the Indian buffet process. Tech. Rep. 2005-001. Gatsby Computational Neuroscience Unit; 2005.
Harris KD, Henze DA, Csicsvari J, Hirase H, Buzsáki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. Journal of Neurophysiology 2000;81(1):401–14.
Hulata E, Segev R, Ben-Jacob E. A method for spike sorting and detection based on wavelet packets and Shannon's mutual information. Journal of Neuroscience Methods 2002;117:1–12.
Jordan MI, Ghahramani Z, Jaakkola T, Saul LK. An introduction to variational methods for graphical models. Machine Learning 1999;37:183–233.
Joshua M, Elias S, Levine O, Bergman H. Quantifying the isolation quality of extracellularly recorded action potentials. Journal of Neuroscience Methods 2007;163(2):267–82.
Lewicki MS. A review of methods for spike sorting: the detection and classification of neural action potentials. Network 1998;9(4):R53–78.
Liu JS. Monte Carlo strategies in scienfitic computing. New York, NY: Springer–Verlag; 2001.
MacEachern S, Muller P. Estimating mixture of Dirichlet process models. Journal of Computational and Graphical Statistics 1998;7:223–38.
Neal RM. Connectionist learning of belief networks. Artificial Intelligence 1992;56:71–113.
Neal RM. Probabilistic inference using Markov chain Monte Carlo methods. Tech. Rep. CRG-TR-93-1. University of Toronto; 1993.
Neal RM. Markov chain sampling methods for Dirichlet process mixture models. Tech. Rep. 9815. Department of Statistics, University of Toronto; 1998.
Nguyen D, Frank L, Brown E. An application of reversible-jump Markov chain Monte Carlo to spike classification of multi-unit extracellular recordings. Network 2003;14(1):61–82.
Pitman J. Combinatorial stochastic processes. Notes for Saint Flour Summer School; 2002.
Plexon Inc., 2003. http://www.plexoninc.com/OFS.htm.
Radons G, Becker JD, Dülfer B, Krüger J. Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. Biological Cybernetics 1994;71:359.
Rasmussen C. The infinite Gaussian mixture model. In: Advances in Neural Information Processing Systems, vol. 12. Cambridge, MA: MIT Press; 2000. p. 554–60.
Sahani M, Pezaris JS, Andersen RA. On the separation of signals from neighboring cells in tetrode recordings. In: Advances in neural information processing systems, vol. 10. MIT Press; 1998. p. 222–8.
Sanborn AN, Griffiths TL, Navarro DJ. A more rational model of categorization. In: Proceedings of the 28th annual conference of the on Cognitive Science Society; 2006.
Serruya M, Hatsopoulos N, Fellows M, Paninski L, Donoghue J. Robustness of neuroprosthetic decoding algorithms. Biological Cybernetics 2003;88(3):201–9.
Shoham S, Fellows MR, Normann RA. Robust, automatic spike sorting using mixtures of multivariate *t*-distributions. Journal of Neuroscience Methods 2003;127(2):111–22.
Srebro N, Roweis S. Time-varying topic models using dependent Dirichlet processes. Tech. Rep. UTML TR 2005-003. University of Toronto; 2005.
Takahashi S, Anzai Y, Sakurai Y. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. Journal of Neurophysiology 2003;89:2245–58.
West M, Muller P, Escobar M. Hierarchical priors and mixture models, with application in regression and density estimation. In: Freeman P, Smith A, editors. Aspects of uncertainty. New York: Wiley; 1994. p. 363–86.
Wood F, Black MJ, Vargas-Irwin C, Fellows M, Donoghue JP. On the variability of manual spike sorting. IEEE Transactions on Biomedical Engineering 2004a;51(6):912–8.
Wood F, Fellows M, Donoghue JP, Black MJ. Automatic spike sorting for neural decoding. In: Proceedings of the 27th IEEE conference on engineering in medicine and biological systems; 2004b. p. 4126–9.
Wood F, Griffiths TL. Particle filtering for nonparametric Bayesian matrix factorization. In: Advances in neural information processing systems. Cambridge, MA: MIT Press; 2007.
Wu W, Gao Y, Bienenstock E, Donoghue JP, Black MJ. Bayesian population coding of motor cortical activity using a Kalman filter. Neural Computation 2005;18:80–118.