Computing loss of efficiency in optimal Bayesian decoders given noisy or incomplete spike trains

Carl Smith
Department of Chemistry
cas2207@columbia.edu

Liam Paninski
Department of Statistics and Center for Theoretical Neuroscience
Grossman Center for the Statistics of Mind
Kavli Institute for Brain Science
www.stat.columbia.edu/∼liam

Columbia University
New York, NY 10027

June 6, 2013

Abstract

We investigate Bayesian methods for optimal decoding of noisy or incompletely-observed spike trains. Information about neural identity or temporal resolution may be lost during spike detection and sorting, or spike times measured near the soma may be corrupted with noise due to stochastic membrane channel effects in the axon. We focus on neural encoding models in which the (discrete) neural state evolves according to stimulus-dependent Markovian dynamics. Such models are sufficiently flexible that we may incorporate realistic stimulus encoding and spiking dynamics, but nonetheless permit exact computation via efficient hidden Markov model forward-backward methods. We analyze two types of signal degradation. First, we quantify the information lost due to jitter or downsampling in the spike-times. Second, we quantify the information lost when knowledge of the identities of different spiking neurons is corrupted. In each case the methods introduced here make it possible to quantify the dependence of the information loss on biophysical parameters such as firing rate, spike jitter amplitude, spike observation noise, etc. In particular, decoders that model the probability distribution of spike-neuron assignments significantly outperform decoders that use only the most likely spike assignments, and are ignorant of the posterior spike assignment uncertainty.

1 Introduction

Bayesian decoding of spike trains has received a great deal of previous attention; see e.g. (Pillow et al., 2011) for a recent review. Bayesian decoding methods are particularly appealing because they are optimal in principle (assuming that the "encoding model" — the probabilistic model that describes how information is encoded in the spike trains — is correctly specified), and also because prior knowledge can be explicitly incorporated into the Bayesian model (Kass et al., 2005).

Much of the previous neural decoding literature has assumed that the spike trains to be decoded have been observed completely and noiselessly. Of course, in practice this is rarely the case. For example, information about neural identity or correlations may be lost during spike sorting (Lewicki, 1998; Hill et al., 2011), or spike times measured near the soma may be corrupted with noise due to stochastic membrane channel effects in the axon (Aldworth et al., 2005; Faisal et al., 2005; Faisal and Laughlin, 2007). The recent rise in popularity of optical (e.g., calcium-based) methods for spike detection (which typically offer significantly less signal resolution than electrical recording methods) have made these issues more pressing (Cossart et al., 2003; Ohki et al., 2005); see, e.g., (Mishchenko et al., 2011) for a recent related discussion.

The main contribution of this work is a general framework, using flexible spiking models of populations of neurons, for computationally tractable Bayesian spike train decoding when spike trains are corrupted by either the presence of spike time jitter or spike sorting identity errors. Our goal is to obtain a better analytical and computational understanding of the impact of these spike corruptions on the optimal decoder. Our approach makes heavy use of well-known efficient inference algorithms for hidden Markov models.

Related questions have been previously investigated using simple Poisson neuron models (Aldworth et al., 2005; Gollisch, 2006; Ventura, 2008). In this paper we extend these analyses to a more realistic and flexible class of spiking models. We focus on a Markovian model of spiking dynamics that is similar to those treated in (Herbst et al., 2008; Toyoizumi et al., 2009; Calabrese and Paninski, 2011; Escola et al., 2011; Nossenson and Messer, 2011), and which is closely related to the spike-response/generalized-linear model framework that has become popular in the recent computational and statistical neuroscience literature (Truccolo et al., 2005; Paninski et al., 2007). This model class is sufficiently flexible that we may incorporate realistic stimulus encoding and spiking dynamics, but nonetheless permits exact computation via efficient forward-backward methods familiar from the theory of hidden Markov models.

Any such Bayesian approach must specify a prior distribution on the signals to be decoded. In the context of this Markovian neural encoding model, we note that the broad class of "low-rank" state-space models recently introduced in (Smith et al., 2012) provides a convenient set of conjugate priors (Casella and Berger, 2001) in our setting, implying that the posterior marginal distributions of stimuli given the model's sufficient statistics can be computed exactly, further enabling efficient computation.

We focus here on two major mechanisms of spike train corruption. First, we examine the impact of errors in the timing of each observed spike on decoder performance. Second, we quantify the loss of decoding efficiency when knowledge of the identities of the observed neurons is discarded or corrupted. A concrete example of the latter problem involves spike sorting in low-SNR regimes, where overlaps in spike clusters can lead to errors or excessive uncertainty in the identity of the neuron contributing any observed spike. In each case, our methods allow us to quantify and compute the loss in decoding performance efficiently over a range of parameter values, contributing to a more systematic understanding of the importance of these effects.

The paper is organized as follows. We first define more specifically the models of spike train corruption that we will consider, along with the relevant model assumptions. Then we will describe how to compute the resulting Bayesian decoders, given spike train data which has been corrupted by these mechanisms. In each case, the decoder involves the execution of a two-stage Gibbs sampler (Robert and Casella, 2005) that we will describe in detail. We will briefly describe a class of stimulus priors for which our methods are made particularly efficient by "Rao-Blackwellization" of the Gibbs

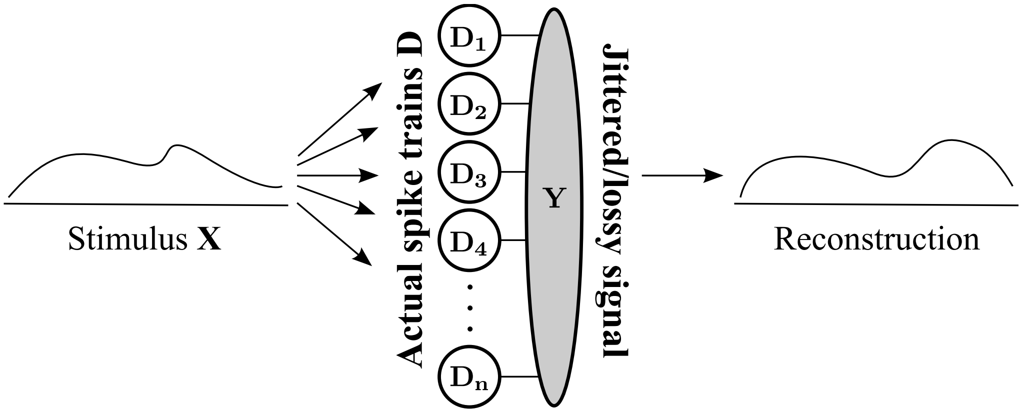

Figure 1: Problem schematic. Neurons are exposed to a common stimulus X. They generate spike trains D which are then degraded by sources of noise. By Y we denote the degraded signal, which is the input to a decoder that reconstructs the stimulus. The spike trains D are assumed to be conditionally independent given the stimulus X.

sampler, in which the sample average is replaced with an estimator with lower variance (Robert and Casella, 2005). Finally, we will present the results of some analyses of simulated data, where we can compare directly to Bayesian decoders given uncorrupted spike train data, and close with a discussion of some open directions for future research.

2 Sources of information loss

For simplicity, we assume that each observed neuron responds conditionally independently to the stimulus (Fig. 1). We will focus on Bayesian reconstruction of stimuli from spike trains corrupted by spike-time jitter and neuron identity loss. In both cases, we assume that the state of each neuron evolves according to some Markovian dynamics, passing through single-neuron states $Q = \{q_t\}$, where q_t is the state of the neuron at time t. Some or all of the transition probabilities may be functions of the stimulus $X = \{x_t\}$, where x_t is the value of the stimulus at time t. (As usual, the X may be viewed more generally as some filtered or transformed representation of the physical stimulus, or may represent some more abstract covariate that the neural activity depends upon.) We assume that time has been discretized into equal-length bins that are sufficiently small that a given neuron can fire at most once within each bin.

2.1 Spike time jitter

Biophysically, sources of the variability in spike-times include stochasticity in the activity of ion channels and in synaptic transmission, or in the timing of spike detection, particularly in optical recordings. (Note that temporal downsampling can also be interpreted as a form of temporal noise, once the spike times are upsampled back to their original resolution.) The input to the decoder, then, may be a set of spike trains $Y = \{y_t\}$, $y_t \in \{0, 1\}$, jittered from the true spike-time data D. In devising a decoding algorithm, one might ignore these sources of noise and take the observed spike-times as the actual times. Alternatively, one may perform inference directly on the temporally corrupted signal, explicitly modeling the sources of temporal noise.

The importance of spike-time noise has been recognized previously; for example, (Aldworth et al., 2005) describe an iterative algorithm for dejittering spike trains by aligning single stimuli to find the most likely stimulus to have preceded a jittered spike. Our approach differs by attempting to reconstruct the entire stimulus given a jittered full spike train, instead of a single spike. In addition, (Gollisch, 2006) introduces an iterative expectation-maximization (EM) algorithm that improves receptive field estimates by explicitly incorporating spike-time jitter into a linear-nonlinear Poisson (LNP) model. By contrast, we will focus on stimulus decoding, rather than the estimation of encoding models; in addition, our approach is able to handle more general non-Poisson spike generation.

2.2 Spike identity loss

A key challenge in neural recording is the problem of spike sorting: correctly assigning each spike to the neuron that generated the spike, given noisy extracellular voltage or optical recordings (Lewicki, 1998; Hill et al., 2011). Any single electrode (or optical pixel) will often record the activity of more than one neuron, in addition to any background noise. An optimal decoder therefore needs to keep track of the posterior uncertainty corresponding to the assignment of each spike, as previously emphasized, e.g., by (Wood and Black, 2008; Ventura, 2008; Ventura, 2009), in the context of spike trains which can be modeled as inhomogeneous Poisson processes. (See also (Chen et al., 2012), who like (Ventura, 2008) emphasize the importance of retaining as much information as possible in the raw extracellular voltage signal, and not discarding the “unsortable” spikes.) Again, a key extension here is to generalize to the non-Poisson setting.

3 Model assumptions

We model the dynamics of the neurons in the observed populations in terms of discrete-time Markov chains, similar to the models treated, e.g., in (Sahani, 1999; Herbst et al., 2008; Calabrese and Paninski, 2011; Escola et al., 2011). These models are sufficiently flexible to handle refractoriness, burstiness, adaptation, and other aspects of spike train dynamics while at the same time remain highly computationally tractable, as we will describe in further depth below.

Fig. 2 illustrates a simple example of a Markovian model neuron with three states. In this model, spiking occurs in the state (1). States (2) and (3) are non-spiking states, and the only dependence of these dynamics on the stimulus X is that the transition from state (3) to state (1) at time t occurs with probability $f(x_t)$. We observe only the spikes (or a noisy function thereof); the state variables are never observed directly. Aside from the choice of response function $f(x_t)$, the only free parameter is p_{23}, which determines the average length of the refractory period[1]. In all of our experiments (results shown below) we assume that the neuronal population is composed of conditionally independent neurons of exactly this type, with $f(x_t) = x_t$ or $f(x_t) = 1 - x_t$. This choice is for simplicity and is not essential, as will become clear below; some conditional interdependence among the neurons is possible with only a slight cost in tractability, for example, and the stimulus-dependence of the various transition probabilities could similarly be more complicated.

[1] Note that the refractory period here is relative. Markov models that impose an absolute refractory period are easy to imagine. Indeed, similar models have been considered that include an absolute refractory period via some number of additional refractory states that the neuron passes through deterministically (e.g., (Herbst et al., 2008; Haslinger et al., 2010)). Our methods apply to such models as well.

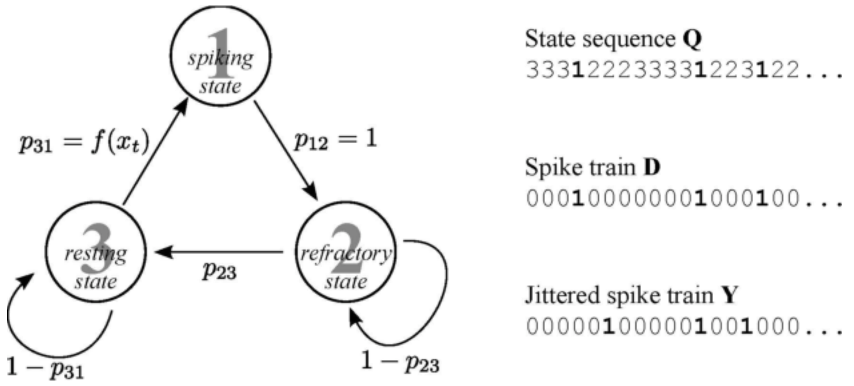

Figure 2: Left: A simple discrete-time model for a neuron responding to a time-varying stimulus $X = \{x_t\}$. A neuron in the resting state (3) enters the spiking state (1) with a probability that is a function of the current stimulus value. The neuron passes immediately from (1) to a refractory state (2) where it stays for a time drawn from the geometric distribution with parameter p_{23}, which is not a function of the stimulus. Right: example values of the hidden state sequence Q, the true (unobserved) spike times D, and the observed spike data Y.

Next we describe the model assumptions corresponding to each of the information-loss mechanisms discussed above.

3.1 Spike time jitter

In this setting we assume that we can only observe temporally-jittered versions of the true spike times. For concreteness, we assume that the jittered spike times have been drawn from discretized Gaussian distributions with standard deviation σ centered on the actual (unknown) spike time. We assume that σ is known or can be estimated based on previous experiments. (All of these assumptions can be relaxed considerably.)

3.2 Identity loss

In this setting we focus on spike-sorting errors. Observations here correspond to spike feature vectors, which represent, e.g., the amplitude and width of the spike voltage waveform, or a projection onto principal components in voltage waveform space. The distributions of these spike features overlap, so it is impossible to say with certainty which spike came from which neuron. For concreteness, we assume that these feature vectors are two-dimensional and drawn from Gaussian distributions whose mean and covariance are known or can be estimated. (Again, all of these assumptions can be relaxed considerably.)

4 Stimulus decoding

In both of the settings described above, the accuracy of stimulus reconstructions provides a measure of the information conveyed in the observed spike data. For a given stimulus prior $p(X)$ and noise distribution $p(\mathbf{Y}|X)$, we would like to, for example, compute the mean summed squared error

(MSE),

$$\mathbb{E}_{P(X,\mathbf{Y})} \sum_t (x_t - \hat{x}_t)^2 \,, \tag{1}$$

where $\hat{X}$ is the posterior mean $\mathbb{E}(X|\mathbf{Y})$, the optimal stimulus reconstruction for the given $\mathbf{Y}$ under the square error loss function. We use bold-face $\mathbf{Y}$ to refer to the noisy signals from all neurons, and Y to refer to the noisy signals from just one neuron. Similarly, we use $\mathbf{Q}$ to refer to the state sequence of the assembly of neurons, and Q to refer to a single neuron's state sequence. When appropriate, we include a superscript index to specify the neuron, as in Q^i.

Our approach to compute the expression in (1), is to 1) draw many stimuli X from the stimulus prior $P(X)$, 2) for each stimulus, draw a sample state sequence $\mathbf{Q}$ and noisy observations $\mathbf{Y}$ from the model $P(\mathbf{Q}, \mathbf{Y}|X)$, 3) decode each set of noisy observations separately to obtain $\hat{X}$, and finally 4) compute the squared error $\sum_t (x_t - \hat{x}_t)^2$ for each such sample and average to obtain an unbiased Monte Carlo estimate of the expected loss (1). Steps 1, 2 and 4 are straightforward. For these Markovian neuron models, step 3 may be accomplished by the use (described below) of Gibbs sampling (Robert and Casella, 2005) in time that scales linearly with the duration of the experiment, i.e. the number of observations.

4.1 Gibbs sampling and Rao-Blackwellizaition

Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm for sampling from complicated joint distributions $p(Z) = p(z_1, z_2, \cdots, z_n)$ whose conditional distributions $p(z_i|Z\backslash z_i)$ are relatively easy to sample from, where $Z\backslash z_i$ is the set of components of Z except for z_i (Robert and Casella, 2005). The algorithm proceeds by first initializing all the components of Z with some typical values. Then one cycles through the components (in whatever order, so long as all components are iterated over), sampling each z_i from its conditional distribution $p(z_i|Z\backslash z_i)$, holding all the other components fixed at their previous values. Then the value of z_i is updated to the sampled value before moving on to the next component in the cycle, and that value may be added to a tally. To estimate the marginal means of the components, these tallies are finally divided by the overall number of cycles. Since Gibbs sampling provides samples from the full joint distribution $p(Z)$, we may use these samples to compute estimates of any desired marginal moments or quantiles.

In an important variant of Gibbs sampling which we employ in the stimulus decoding problem, for each cycle and for each component, the conditional mean of the component, rather than the sampled value of the component, is tallied in the estimation of the marginal mean of the component. When this is possible, i.e. when the conditional expectation $\mathbb{E}(z_i|Z\backslash z_i)$ is known for any value of the other components $Z\backslash z_i$, it can be shown that the variance of this estimator is less than that of the average of sample values. Such an algorithm is termed a Rao-Blackwellized Gibbs sampler (Robert and Casella, 2005). More generally, when feasible, we can record the full conditional distribution $p(z_i|Z\backslash z_i)$ at each iteration, and therefore obtain better estimates of the conditional variance, quantiles, etc., as well.

The Gibbs approach is most useful when the sampling from these conditional distributions is simpler than sampling from the joint distribution directly, as is the case in the present stimulus decoding problem. The details of this application are detailed below.

4.2 The stimulus decoder Gibbs sampler

We proceed by conditioning on the hidden state sequence $\mathbf{Q}$ that led to the generated spike train D; the probability of a stimulus X given the observed data $\mathbf{Y}$ may be written as the marginal probability $p(X|\mathbf{Y}) = \int p(X, \mathbf{Q}|\mathbf{Y}) d\mathbf{Q}$. Thus, to obtain a sample from $p(X|\mathbf{Y})$, we draw samples from $p(X, \mathbf{Q}|\mathbf{Y})$ but only record the value of X. We sample from this joint distribution by Gibbs sampling: given the sample $\mathbf{Q}^{(i)}$, we draw $X^{(i)}$ from $p(X|\mathbf{Q}^{(i)}, \mathbf{Y})$ and record $X^{(i)}$ (or the Rao-Blackwellized statistics $p(X|\mathbf{Q}^{(i)}, \mathbf{Y})$ or $E(X|\mathbf{Q}^{(i)}, \mathbf{Y})$, if these are analytically available), then draw $\mathbf{Q}^{(i+1)}$ from $p(\mathbf{Q}|X^{(i)}, \mathbf{Y})$, and iterate. Fig. 3 shows an example sequence of samples of X and $\mathbf{Q}$.

The problem of stimulus reconstruction has been reduced to the problem of sampling from these two conditional distributions, $p(X|\mathbf{Q}, \mathbf{Y})$ and $p(\mathbf{Q}|X, \mathbf{Y})$. What remains is to sample from these conditional distributions in an efficient way; in particular, since we are interested in reconstructing stimuli given long samples of spike train data, it is important to develop methods that scale linearly with the length of the observed spike train data. In the following we will address each subproblem in turn, and we will illustrate the procedure with the simple three-state Markov model neuron introduced above.

Both in the jitter and identity loss cases, sampling from $p(\mathbf{Q}|X, \mathbf{Y})$ will be a matter of framing the dynamics and noisy signal as a hidden Markov model. Once that is done, we can sample from $p(\mathbf{Q}|X, \mathbf{Y})$ using standard forward-backward recursions (Rabiner, 1989). Sampling from $p(X|\mathbf{Q}, \mathbf{Y})$ turns out to require similar approaches in both jitter and identity loss cases, and will involve a simple application of Bayes rule to write down the distribution to be sampled from. In the case that this distribution can be integrated analytically, we can employ Rao-Blackwellization.

4.3 Hidden Markov models

Hidden Markov models (HMMs) (Rabiner, 1989) are convenient graphical models for the analysis of discrete-time systems. HMMs represent joint distributions over latent (unobserved) variables on which some observed variables depend. To be concrete, a HMM consists of two components. First is a discrete Markov chain of latent variables $Z = \{z_t\}_{t=1}^{T}$, so that

$$p(z_t|z_1, \cdots, z_{t-1}) = p(z_t|z_{t-1}).$$

Second is a discrete set of observations $W = \{w_t\}_{t=1}^{T}$, one observation corresponding to each latent variable, and each of which has the Markov property

$$p(w_t|Z, W \backslash w_t) = p(w_t|z_t).$$

Many systems can be reasonably modelled by HMMs. In our case, the variables w_t will correspond to the binary presence or absence of a spike at time t, or the observed spike feature vector at time t; the z_t variables correspond to the unobserved neuronal state illustrated in Fig. 2.

It turns out that inference in a HMM – i.e. estimation of the latent variables given only values of the observed variables – is particularly efficient. In particular, inference time is linear in the length of the Markov chain. When Z and W form a hidden Markov model (HMM), sampling from the posterior state distribution $p(Z|W)$ proceeds via the standard filter-forward sample-backward HMM recursion (Fruhwirth-Schnatter, 2006). We first compute the "forward probabilities"

$$a_t(i) = p(w_1, \cdots, w_t, z_t = i),$$

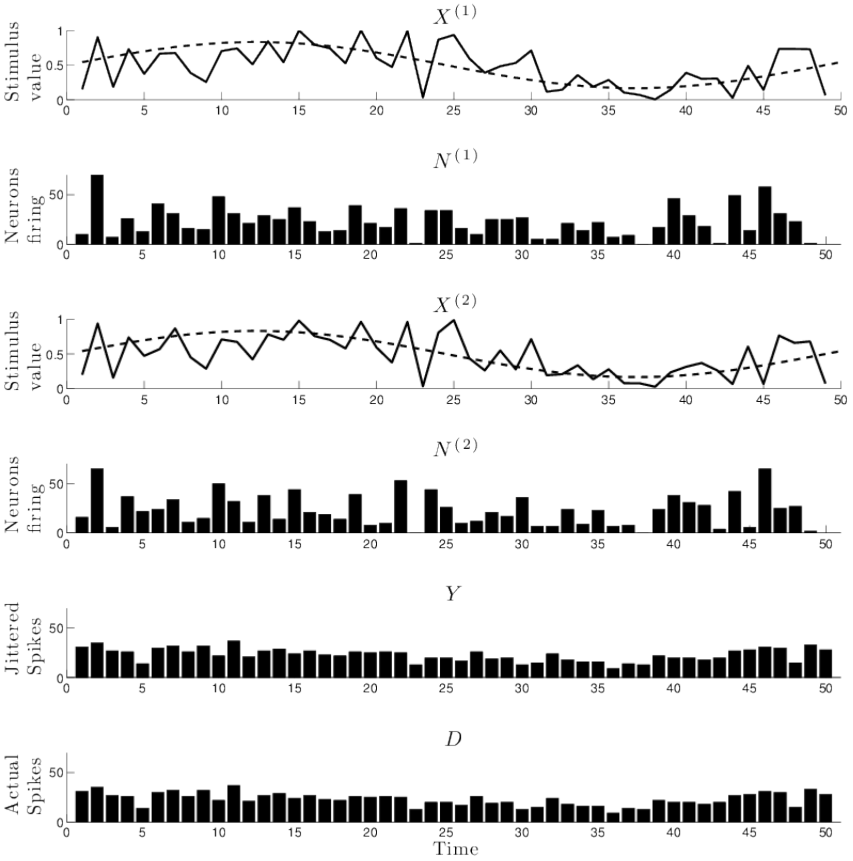

Figure 3: An example sequence of samples from $p(X|\mathbf{Q})$ and $p(\mathbf{Q}|X, \mathbf{Y})$, drawn via the Gibbs method described in the main text. The dashed curve is the actual stimulus. The solid curves in the stimulus panels are samples of X. The bars in the neurons firing panels show the number N of neurons firing at each time bin in the sample state sequence sample $\mathbf{Q}$. I.e., N_t is the number of neurons in the spike state ($q_t = 1$) at time t. These data are from a simulation of 300 neurons responding to a sinusoidal stimulus with $p_{23} = 0.1$. The jitter amplitude is $\sigma = 2$. 100 equilibration sweeps were made before these samples were recorded. The bottom two panels show the number of spikes in the jittered and actual spike trains at each point in time.

which can be computed recursively in $O(N^2T)$ time, where N is the number of possible states of each z_i. The forward variables are indexed by state, $i = 1, 2, \cdots, N$, and form a $N \times T$ matrix. We then recurse backwards to sample:

$$\begin{aligned}
z_T &\sim p(z_T = i|W) = \frac{a_T(i)}{\sum_i a_T(i)} \\
z_t &\sim p(z_t|Z_{t+1:T}, W) \\
&= p(z_t|z_{t+1}, W) \\
&\propto p(z_{t+1}|z_t)p(z_t|W_{1:t}) \\
&\propto p(z_{t+1}|z_t)a_t(z_t).
\end{aligned}$$

In the first line, the use of "$\sim$" means that z_T is drawn from the distribution $p(z_T = i|W)$. By appending all the sampled z_t variables into a single vector Z, we obtain a sample from the full conditional distribution $p(Z|W)$ in $O(T)$ total time.

The model parameters themselves (the transition probabilities $\alpha_{ij} = p(z_{t+1} = j|z_t = i)$ and the observation probabilities $\eta_{ij} = p(w_t = j|z_t = i)$ can be inferred, e.g., via the standard Expectation Maximization algorithm (Rabiner, 1989); see (Escola et al., 2011) for further discussion. In the following we take the model parameters as known for simplicity.

4.4 Sampling from the posterior state distribution p(Q—X,Y)

To apply the efficient inference methods associated with HMMs to our information loss settings, we must frame each as a HMM but with the noisy (as opposed to the actual) spikes as the observations. This entails characterizing the neurons in each setting as a system whose states $Q_{1:T}$ form a Markov chain, given stimulus X, and whose noisy observations $Y_{1:T}$ depend on the states $Q_{1:T}$ in a Markovian fashion (Rabiner, 1989). In either setting, the neurons would trivially form a HMM if not for the information loss; for any Markovian neuron model, whether or not there is a spike at a given time bin depends only on the neuron's state q_t at that time. Once the information loss mechanism is introduced, however, it is the noisy signal $Y_{1:T}$ (i.e., the jittered spike times, or the observed noisy feature vectors) that we must formulate as the observation of a conditional HMM.

4.4.1 Spike time jitter

We seek to frame the generation of spike trains in the jitter setting as a HMM. However, note that each neuron's spike train and state sequence are conditionally independent of all other neurons' spike trains and state sequences, given the stimulus. Therefore, to sample the whole set $\mathbf{Q}$ of state sequences given all the jittered spike trains $\mathbf{Y}$ and given the stimulus X, we can sample Q given Y for each neuron in isolation.

Consider a neuron with Markovian dynamics as described above, so that D is its unjittered spike train, with spike times $\{t_i\}$, $i = 1, \cdots, N_{spikes}$. Then that neuron's state sequence Q forms a HMM with D as the observations. Given D and the stimulus X and with knowledge of the underlying Markovian dynamics of the neuron, we can sample $Q \sim p(Q|X, D)$ by the standard filter forward sample backward algorithm.

If instead we are given a jittered spike train Y, with jittered spike times $\{y_i\}$, $i = 1, \cdots, N_{spikes}$, then sampling Q is not so straightforward. Y and Q do not form a HMM. To see this, consider a given jittered spike time y_i. This spike corresponds to some true spike time t, whose spike is the observation from the neuron in state q_t. However, t could be anywhere in a window of time bins surrounding y_i. The width of this window is determined by the jitter amplitude. In this sense the spike time y_i may contain information regarding the state of the neuron at any of several or many time bins. We therefore need to take an alternate approach to sampling Q in the jittered case.

One approach is to break up the sampling into two Gibbs steps. First, sample the true spike train D, given X and Y. Then, with D in hand, sample $Q \sim p(Q|X, D)$ by the forward backward recursion described above. (Note that $p(Q|X, D, Y) = p(Q|X, D)$, since Y is nothing but a corrupted copy of D.) There are many options for sampling D. One straightforward approach is to use Gibbs again, this time sampling one spike at a time. (Of course in some cases — for example, in the case of a highly bursty neuron — it might make more sense to move multiple spikes at once. The discussion below can easily be generalized to this case; see, e.g., (Mishchenko and Paninski, 2011) for further discussion.) More precisely, for each $1 \leq i \leq N_{spikes}$, we sample t_i from

$$p(t_i = t|\{t_{j\neq i}\}, X, Y) \propto p(Y|t_i = t, \{t_{j\neq i}\})p(t_i = t|\{t_{j\neq i}\}, X). \tag{2}$$

The second factor on the right hand side can be computed by a forward-backward recursion. (It is worth noting that we do not have to perform the full recursion every time we update a spike time t_i: a local change in t_i will have only a local effect on the values of the forward and backward probabilities. As we recursively update these probabilities, therefore, we can stop once we see that the difference between the updated values and the previous values is negligible. This ensures that the cost of a spike time update does not scale with the total length of the spike train.)

To compute the the first factor on the right hand side of eq. (2), we need to specify a model for the temporal noise process that defines $p(Y|D)$. The simplest model is that each spike time is jittered independently:

$$p(Y|D) = \sum_{\sigma} \prod_{j=1}^{N_{spikes}} p(y_{\sigma(j)}|t_j).$$

The sum over σ here is over all N_{spikes}! possible permutations (relabelings) of the spikes, accounting for the fact that we don't know which spike in the observed set Y corresponds to a given spike t_i in the unobserved true set D. However, once we pick a labeling σ that maps D to Y, then computing the conditional probability of Y just reduces to a product over the individual jitter densities $p(y_{\sigma(j)}|t_j)$.

Clearly, direct computation of the sum over σ becomes intractable as N_{spikes} becomes large, since the number of terms in the sum grows exponentially. However, note that we do not have to compute the full sum. Instead, we just need to compute the change in the sum as the i-th spike is moved from the current time t_i to a new time t'_i. If the variance of the jitter distributions $p(y_j|t_j)$ is not too large, any given spike time t_j will only be close to a few observed spikes y_i, and the computation of the change in $p(Y|D)$ becomes tractable. As a concrete example, if y_i is sufficiently distant from the nearest other observed spikes that we can neglect the probability that the spike times have been switched by the jitter, then we can uniquely associate the observed spike y_i with a single true unobserved spike t_i, and we have

$$\begin{aligned}
\frac{p(Y|D)}{p(Y|D')} &= \frac{\sum_\sigma p(y_{\sigma(i)}|t_i)\prod_{j\neq i} p(y_{\sigma(j)}|t_j)}{\sum_\sigma p(y_{\sigma(i)}|t'_i)\prod_{j\neq i} p(y_{\sigma(j)}|t_j)} \\
&\approx \frac{p(y_i|t_i)\sum_\sigma \prod_{j\neq i} p(y_{\sigma(j)}|t_j)}{p(y_i|t'_i)\sum_\sigma \prod_{j\neq i} p(y_{\sigma(j)}|t_j)} \\
&= \frac{p(y_i|t_i)}{p(y_i|t'_i)}.
\end{aligned}$$

(We have abused notation slightly here: this sum over σ includes all permutations of the $N_{spikes} - 1$ spike times not including t_i or y_i.) Thus we can now combine the two necessary factors in eq. (2), and update t_i either via standard Gibbs or a Metropolis-within-Gibbs (Robert and Casella, 2005) approach. (See (Chen et al., 2009; Tokdar et al., 2010) for some related approaches.)

4.4.2 Neural identity loss; spike addition or deletion

In the identity loss / spike sorting setting, any given observed spike could have been emitted from any of multiple neurons. Separate independent neuron models cannot capture such ambiguity, and therefore we must combine the models across neurons appropriately.

One direct approach is to construct a single state space that represents the dynamics of all the neurons at a given recording electrode. Suppose there are m discernible neurons at a given location, each with at most K Markovian internal states. We can track the state of all of these neurons simultaneously by forming the direct product of each individual state's transition and observation matrices (obtaining a joint state variable that can take on at most K^m possible values), as discussed in (Calabrese and Paninski, 2011). If each neuron i has a transition matrix P_i, then the joint state transition matrix is the Kronecker product (Horn, 1986) of the single-neuron transition matrices,

$$P = \bigotimes_{i=1}^{m} P_i.$$

To construct a HMM it remains to specify an emission matrix, whose entry (i, j) is the probability of seeing observation i given that the system is in state j. For our three state neuron, observed without noise, this would be a 3×2 matrix:

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix},$$

where the rows correspond, from top to bottom, to the states *firing*, *refractory*, and *resting*, and the columns correspond, from left to right, to the observations *firing* and *not firing.* Changing this emission matrix would be one way of modeling spontaneous spiking or dropped spikes.

In the spike sorting setting, we replace this simple discrete observation alphabet with a continuous, typically multi-dimensional feature vector $\mathbf{y}$ defined in terms of, e.g., the amplitude of the voltage (or the magnitude of some projection onto a principal component) triggered on a threshold crossing. We write this in terms of the density function

$$b_{\mathbf{q}}(\mathbf{y}) = \text{Prob}[\mathbf{y}_t = \mathbf{y} | \mathbf{q}_t = \mathbf{q}].$$

If we assume for concreteness that the observed feature vectors have a Gaussian distribution given the identity of the neuron that emitted the spike, then $b_{\mathbf{q}}(\mathbf{y})$ is readily evaluated as the Gaussian density at $\mathbf{y}$, with mean and covariance corresponding to the neuron that is firing (as indexed by the multineuron state $\mathbf{q}$).

We can easily extend this model to treat the possibility of dropped spikes or spontaneous spiking. To account for dropped spikes, we add a cluster corresponding to no spike, with its own mean and covariance. Assume that the probability of a spike being dropped is p_d. Then whenever $\mathbf{q}$ is a spiking state for one of the neurons, there is a chance, p_d, that the observation is drawn from the cluster corresponding to the absence of a spike. The observation density may be modified accordingly to handle this possibility. See also (Goodman and Johnson, 2008), who find that in at least some cases false positives will have a greater negative effect on decoding than either dropped or mislabeled spikes.

If all the neurons at some electrode have the same response function (a very special case), then all we are interested in is the number of neurons in each state. In this case the state space size scales polynomially in the number of neurons m at the recording location, not exponentially. The transition probabilities in this condensed state-space may be computed as follows. Define the vector $\vec{n}_t$ such that $n_t^{(k)}$ is the number of neurons in state k at time t. For each state k, define the vector $\vec{\nu}_{tk}$ as follows: $\nu_{tk}^{(i)}$ is the number of neurons in state i at time $t-1$ that are in state k at time t. Then $\vec{n}_t = \sum_k \vec{\nu}_{tk}$. The transitions of the neurons in state i at time $t-1$ are independent of the transitions of the neurons in state $j \neq i$ at times $t-1$. Therefore

$$\begin{aligned} p(\vec{n}_t|\vec{n}_{t-1}) &= p\left(\sum_k \vec{\nu}_{tk}\middle|\,\vec{n}_{t-1}\right) \\ &= conv\left[p\left(\vec{\nu}_{tk}\,\middle|n_{t-1}^{(k)}\right)\right] \\ &= conv\left[Mult\left(n_{t-1}^{(k)}, p(q_t|q_{t-1}=k)\right)\right] \end{aligned}$$

where the q_t index the single-neuron states, $conv()$ denotes m-fold discrete convolution, and $Mult(N, p)$ denotes the multinomial distribution with parameters N and p. The first equality follows by definition. The second equality follows because $\vec{n}_t$ is the sum of independent random variables $\vec{\nu}_{tk}$, and because $p\left(\vec{\nu}_{tk}\,|\vec{n}_{t-1}\right) = p\left(\vec{\nu}_{tk}\,\middle|n_{t-1}^{(k)}\right)$, i.e. $\vec{\nu}_{tk}$ is conditionally independent of the other components of $\vec{n}_{t-1}$. The transition of each neuron from state k to some state i is a trial with a fixed finite number of possible outcomes, independent of all other trials. Therefore each vector $\vec{\nu}_{tk}$ is drawn from a multinomial distribution with $n_{t-1}^{(k)}$ trials and event probabilities $p(q_t = i|q_{t-1}), i = 1, \cdots, K$.

Finally, note that we can easily extend our model to treat multiple spikes in a time bin, whether by the same neuron or by different neurons at the same electrode, but the model becomes more complex as the time bin width increases, and as the maximum possible number of spikes per bin grows. For example, at an electrode monitoring two neurons, A and B, multiple spikes could be treated by the addition of Gaussian modes for events like “A spikes, B does not”, “A does not spike, B spikes twice”, “A and B both spike once”, and so on.

In summary, whether or not all neurons have the same transition matrix, we have established that we may sample from the posterior $p(\mathbf{Q}|X, \mathbf{Y})$ in a computationally efficient way. However, the computational cost does grow exponentially with the number of neurons m in a single cluster. If m is very large, it becomes more attractive to use a blockwise-Gibbs approach (as described, e.g., in (Mishchenko and Paninski, 2011)), in which we sample the states of a small subset of neurons while holding the states of the other neurons fixed.

4.5 Sampling from the posterior stimulus distribution p(X—Q,Y)

We have shown that we can tractably sample from one side of the Gibbs sampler decoder (the $p(\mathbf{Q}|X, \mathbf{Y})$ side) in a way that scales linearly with the temporal length T of the dataset to be decoded. In the other stage, we sample the stimulus X from $p(X|\mathbf{Q}, \mathbf{Y})$. We will show that this half of the Gibbs sampler is relatively straightfoward; moreover, this step can be made particularly efficient in certain special cases.

First note that $\mathbf{Y}$ contains no more information about the stimulus than $\mathbf{Q}$, so that $p(X|\mathbf{Q},\mathbf{Y}) = p(X|\mathbf{Q}) \propto p(\mathbf{Q}|X)p(X)$. The conditional density $p(\mathbf{Q}|X)$ factorizes into a product of its transition probabilities: for each neuron,

$$p(Q|X) \propto p(q_0)\prod_t p(q_{t+1}|q_t, x_t),$$

by the Markov assumption.

If we restrict our model to log-concave transition probabilities $p(q_{t+1}|q_t, x_t)$ (i.e., assume that $\log p(q_{t+1}|q_t, x_t)$ is concave in x_t), and if our stimulus prior is log-concave in X as well, then the posterior on X is log-concave and therefore unimodal, which means that the sampler will not get trapped in local optima, and we can use efficient sampling methods as discussed in (Ahmadian et al., 2011). The restriction to log-concave nonlinearities is not severe in our class of Markov models; see e.g. (Escola et al., 2011) for further discussion.

If we specialize further we can obtain an even more efficient sampler. In our simple three-state model, let's assume that the transitions $p(q_{t+1} = \text{“}\textit{spike}\text{”}|q_t = \text{“}\textit{rest}\text{”}, x_t) = f(x_t)$ depend on x_t in a simple linear fashion: either $f(x_t) = x_t$ (for cells of “ON” type) or $f(x_t) = 1 - x_t$ (for cells of “OFF” type). In this case the likelihood term $p(\mathbf{Q}|X)$ can be written in a very simple form:

$$p(X|\mathbf{Q},\mathbf{Y}) \propto \prod_t x_t^{C(t)}(1-x_t)^{D(t)} \tag{3}$$

where $C(t)$ counts the number of ON neurons passing into the spike state from the rest state, plus the number of OFF neurons remaining in the rest state at time t; $D(t)$ is defined similarly, but with ON and OFF reversed.

This likelihood term leads to a remarkably simple posterior if $p(X)$ has a similar form. For example, if x_t is chosen independently at each time step, with a uniform distribution on the interval $[0, 1]$, then the posterior on X is given by a simple independent product of beta distributions, which can be sampled trivially. (Similar expressions involving sums of incomplete beta functions hold for the case that the stimulus prior is composed of an independent product of polynomials, or if the response functions $f(.)$ have a more general polynomial form.)

Of course, the assumption that each x_t is a priori independent in time is typically too strong. In general, the choice of stimulus prior will depend on the details of the particular experimental setup. What is most important is to have available a broad class of tractable models to choose from. (Smith et al., 2012) recently introduced a class of models which we can use to provide a convenient correlated prior for X. These so-called low-rank models are joint distributions whose structure allows for fast exact inference in settings where standard models such as the discrete HMM or linear Gaussian models are not applicable. These models consist of joint distributions over continuous variables X whose dependency structure can be expressed via discrete latent variables $Z = \{z_t\}$ coupling the main variables X. In the case that $p(X)$ forms a Markov chain, such a low-rank $p(X)$ may be decomposed as follows:

$$\begin{aligned} p(X) &= p(x_1) \prod_{t=1}^{T-1} p(x_{t+1}|x_t) \\ &= p(x_1) \prod_{t=1}^{T-1} \sum_{z_t}^{R} p(z_t|x_t) p(x_{t+1}|z_t) \end{aligned} \tag{4}$$

for appropriate discrete auxiliary variables z_t. (The first equation above is the standard Markov condition; the second equation expresses the low-rank nature of the conditionals $p(x_{t+1}|x_t)$, with R denoting the "rank" of the model.) See (Smith et al., 2012) for full details; the key point is that exact inference over X is tractable in this case via standard forward-backward recursions in $O(TR^2)$ time, despite the fact that the x_t variables may be non-discrete and highly non-Gaussian. As one useful example of a low-rank model, consider $p(X)$ of the nearest-neighbor polynomial form

$$p(X) \propto \prod_t \sum_{i=0}^{R} a_{ti} x_t^{\alpha_i} (1-x_t)^{\beta_i} x_{t+1}^{\gamma_i} (1-x_{t+1})^{\delta_i}, \; 0 \le x_t \le 1. \tag{5}$$

(The normalization constant of $p(X)$ is found via the same forward recursion as is used to perform inference.) As discussed in (Smith et al., 2012), this class of priors has several helpful features. First, with an appropriate choice of the polynomial order R and the coefficients $\{a_{ti}\}$, we have a good deal of flexibility in modeling the correlations in x_t. Second, this prior is conjugate to the likelihood term (3); i.e., the posterior has the same form as the prior. Finally, by the general theory mentioned above, exact samples can be drawn from this prior (and posterior) in $O(TR^2)$ time. As an example, Fig. 4 illustrates the effect of changing the prior when computing the posterior expectation $E(X|\mathbf{Q})$. In particular, we used

$$p(X) \propto \prod_t \sum_{i=0}^{R} \binom{R}{i}^2 x_t^i (1-x_t)^{R-i} x_{t+1}^i (1-x_{t+1})^{R-i} \tag{6}$$

for two different choices of the parameter R, which (as discussed further in (Smith et al., 2012)) serves to set the smoothness of samples from $p(X)$: larger values of R correspond to smoother samples from the prior (and in turn to a smoother posterior expectation). (We point out that, as discussed in (Smith et al., 2012), R can be chosen by standard model-selection methods (e.g., maximum marginal likelihood); in addition, for sources of signals that tend to have occasional large jumps, we can add a "slab" term – i.e. a small constant – to each pair potential to allow for such jumps, and that such a prior would still be an instance of Eq. (5).) We do not present detailed error statistics here, as this example is meant only to illustrate the behavior of the smoother given fully-observed spike trains; we will discuss applications to corrupted spike trains below.

We can further improve the efficiency of the sampler by the Rao-Blackwellization procedure discussed above, because it is also possible to compute the posterior moments $E(x_t|\mathbf{Q})$, $E(x_t^a|\mathbf{Q})$, $E(x_t x_s|\mathbf{Q})$, etc., using a similar $O(TR^2)$ forward-backward approach. Thus, for example, if we want to estimate the posterior mean $E(X|\mathbf{Y})$, instead of recording $X^{(i)}$ at each iteration of the Gibbs sampler, we record $E(X^{(i)}|\mathbf{Q}^{(i)})$, and average over these quantities at the end of the sampling run. In the more general case of log-concave posteriors on X, we typically can not Rao-Blackwellize exactly, but nonetheless we can often substitute an approximation to the mean to obtain a more efficient

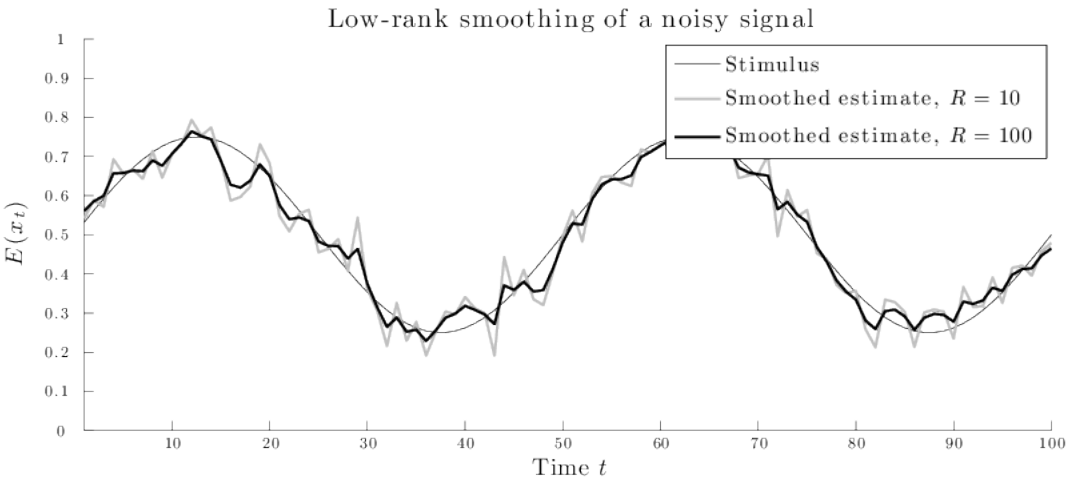

Figure 4: Estimating a stimulus from spike trains with a smoothing low-rank prior. A sinusoidal stimulus (thin black line) was the success probability for a neuron to fire, i.e. $p(n_t = 1|x_t) = x_t$, $n_t \in \{0, 1\}$; n_t not shown. The stimulus was estimated as its posterior mean $E(x_t|\{n_t\})$ with a low-rank smoothing prior of the form of Eq. (6) with ranks $R = 10$ and $R = 100$. Note that the spikes n_t were completely and noiselessly observed here; no spike train corruption has been applied.

estimator. For example, for smooth and unimodal posteriors, the MAP estimate $\arg\max_X p(X|\mathbf{Q})$ tends to be a good approximation to the posterior mean $E(X|\mathbf{Q})$. (Pillow et al., 2011) discuss efficient methods for computing the MAP estimate, which can subsequently be plugged in to approximately Rao-Blackwellize the estimate of the posterior mean.

5 Results

5.1 Spike time jitter

Fig. 5 illustrates the effects of spike-time jitter on decoding accuracy of a stimulus drawn from a low-rank prior. We simulated the responses of 300 neurons driven by this stimulus, using the simple Markovian model described in Fig. 2, with $p_{23} = 0.1$ (recall that this parameter is inverse to the average refractory time) and firing rate $p_{31} = x_t$. The elements of the initial state probability vector were chosen at random. We then independently jittered each true spike according to a discretized Gaussian distribution with zero mean and standard deviation 0 (top) and 2 (bottom). For these parameter values, the probability of spike crossing or overlap was negligible (recall the discussion in section 4.4.1 about the computation of $p(Y|Q)$). The Rao-Blackwellized Gibbs sampler was run for 1,000 iterations after a burn-in period of 100 sweeps, using a prior for X which was independent (stimulus values at different time bins are independent) and uniform on $[0, 1]$. As expected, the accuracy of the reconstruction diminishes with increasing jitter scale.

Fig. 6 summarizes how the reconstruction accuracy varies as the number of neurons changes, and also as the jitter amplitude grows, for this square-wave stimulus. Similar reconstructions for several different types of stimulus are shown in Fig. 7 and Fig. 8. In Fig. 7, the same independent uniform prior is used in the decoder. In Fig. 8, the decoder uses a low-rank prior of the form of

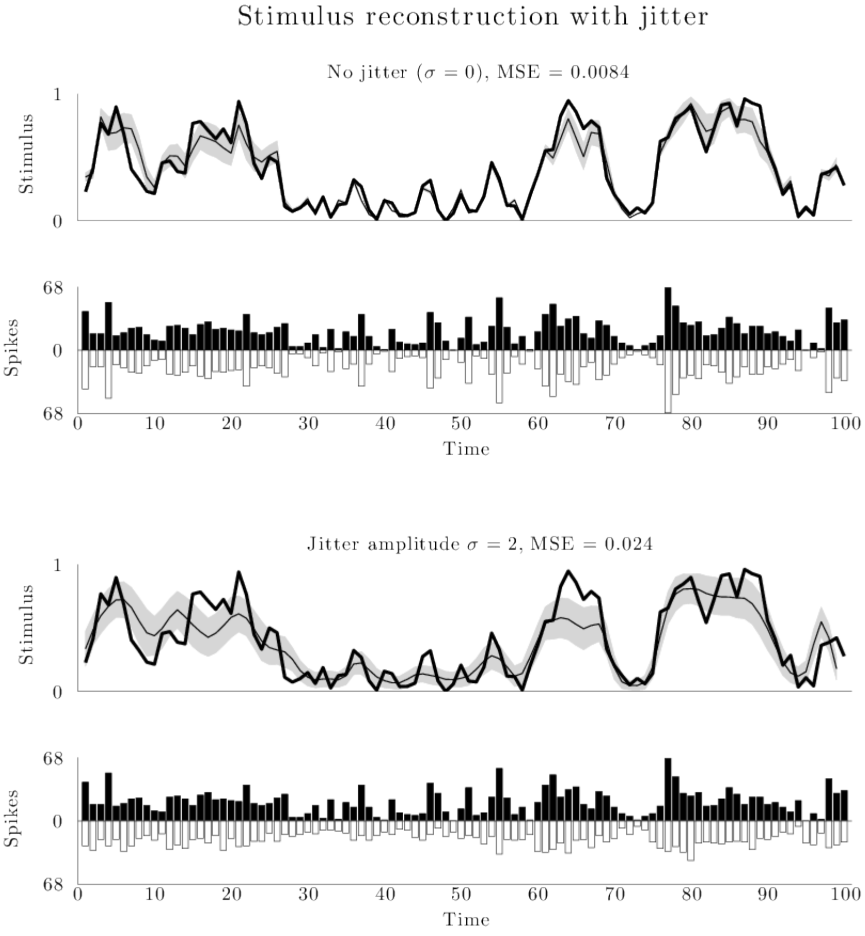

Figure 5: Effects of jitter scale on decoding accuracy for stimulus drawn from the prior of Eq. (6) with $R = 10$. In the decoder, the same low-rank prior was used. The reconstruction algorithm was run for 1,000 sweeps after 100 burn-in sweeps using 300 neurons and $p_{23} = 0.1$. The thick solid line is the actual stimulus. The thin solid line is the reconstruction, with plus or minus one posterior standard deviation shaded gray. The black and white bars show the number of actual (black) and jittered (white) spikes at each instant. The accuracy deteriorates with increasing jitter while the variance of the posterior density increases.

Eq. (6). This corresponds to Eq. (4) where

$$\begin{aligned} x_1 &\sim Uniform([0,1]) \\ z_t|x_t &\sim Binomial(z_t; R, x_t) \\ x_{t+1}|z_t &\sim Beta(z_t, R - z_t) \end{aligned}$$

This simple beta-binomial form of the prior ($p(X)$) makes it possible to analytically compute the

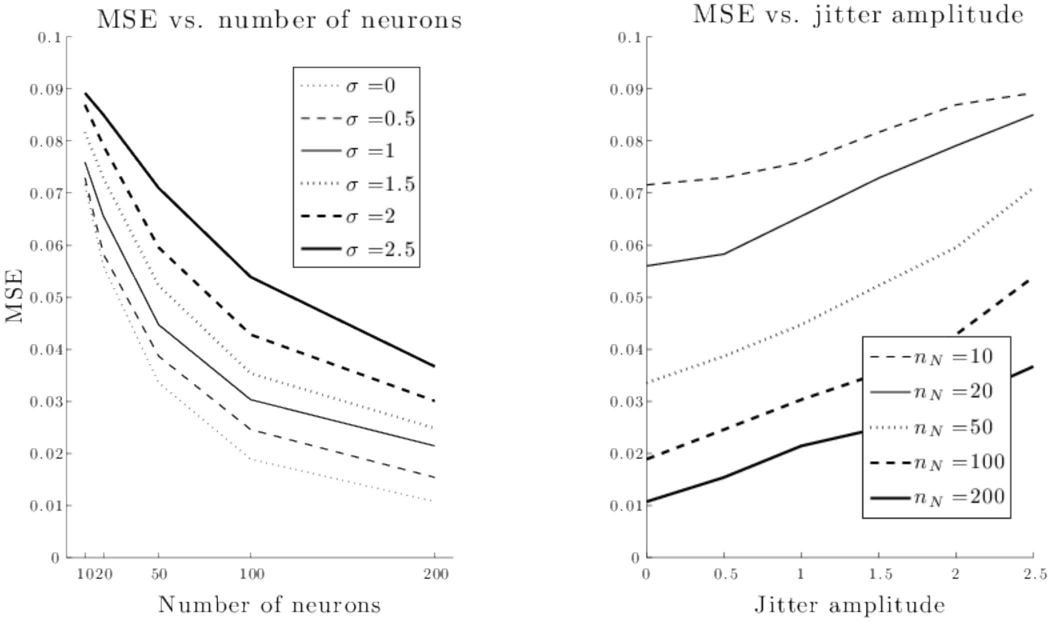

Figure 6: Accuracy (MSE) of reconstruction in the jitter case for various values of number of neurons and jitter amplitude, averaged over 100 reconstructions. A value of $p_{23} = 0.1$ was used for the refractory transition probability. The stimulus was modeled with a low-rank prior of the form of Eq. (6) with a rank of $R = 10$. The experiments ran for 1,000 Gibbs sweeps after a burn-in period of 100 sweeps. The actual stimulus used was a square wave with a period of 10 time bins.

necessary integrals in the Rao-Blackwellized estimator.

The relative accuracy of the reconstruction of the jittered signal varies significantly across different test stimuli. This is consistent with the fact that none of these stimuli (except the one drawn from the independent uniform prior) are "typical" stimuli, under this stimulus prior. However, it is true for any sensible prior that the relative accuracy of the reconstruction will increase with the smoothness of the actual stimulus; the smoother the signal, the less information is lost by jittering a spike a small amount.

We have confirmed in our computer experiments that the computing time involved in inference in the spike time jitter setting scales linearly with the number of neurons as well as with the time T. For example, with 50 neurons, $p_{23} = 0.1$, jitter amplitude $\sigma = 1$, $T = 50$ time bins, and 1,000 Gibbs sweeps after 100 burn-in sweeps, estimation of a square-wave X took 41.7 seconds on a MacBook Pro with a 2.6 GHz Intel Core i7 processor and 8 GB 1600 MHz DDR3 RAM. For the same parameter values except for 100 neurons, estimation took 80.9 seconds. For the same parameter values except for 50 neurons and $T = 100$ time bins, estimation took 79.5 seconds.

5.2 Identity loss

In the case of identity loss, we compare the performance of our decoder to that of two simple decoders. The first decoder is given the spike feature vectors and spike-times that our full decoder receives, computes the overall most likely state path for $\mathbf{Q}$ (i.e., the Viterbi path (Ra-

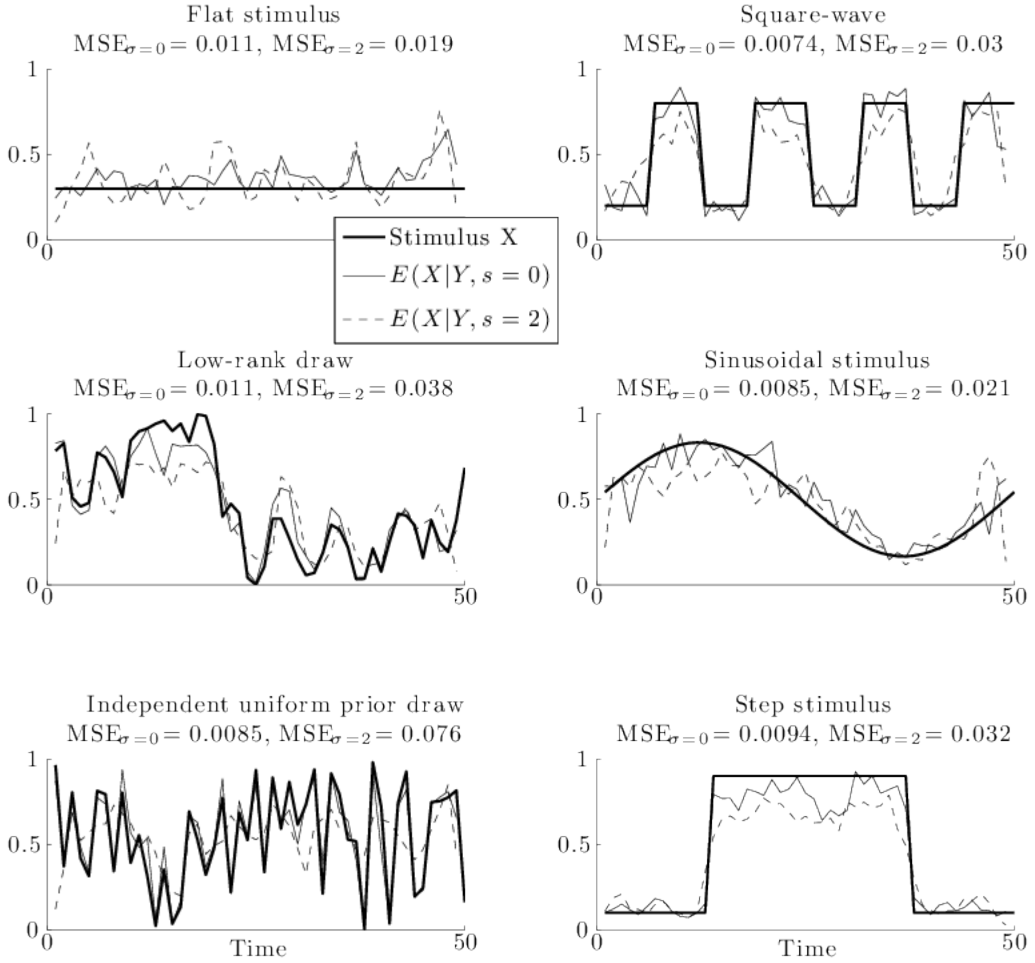

Figure 7: Reconstructions of different stimuli both for no jitter and for jitter using the independent uniform prior. The same parameter values as in Fig. 5 were used, except that 10,000 Gibbs sweeps were run after 1,000 burn-in sweeps. For very smooth stimuli (flat; sinusoidal) reconstruction from the jittered spike train may in fact be more accurate (in terms of MSE) than reconstruction from the original spike train. For less smooth stimuli (more typical of samples from the independent uniform prior) reconstruction from the original spike train is more accurate than reconstruction from the jittered spike train.

biner, 1989)) given the spike-times, and takes the spike-neuron assignments of this state path to be the true assignments; i.e., all uncertainty about the spike sorting is discarded. Subsequently, this decoder computes the posterior mean assuming these spike-neuron assignments – $E(X|D_{MAP}) = E(X|\arg\max_D p(D|Y))$. The second simple decoder is given the actual spike-neuron assignments and subsequently computes the posterior mean $E(X|D)$. (We expect this decoder to outperform any decoder that does not have access to the uncorrupted spike train data D.) In each case, in our experiments, an independent and uniform prior on X was used for the

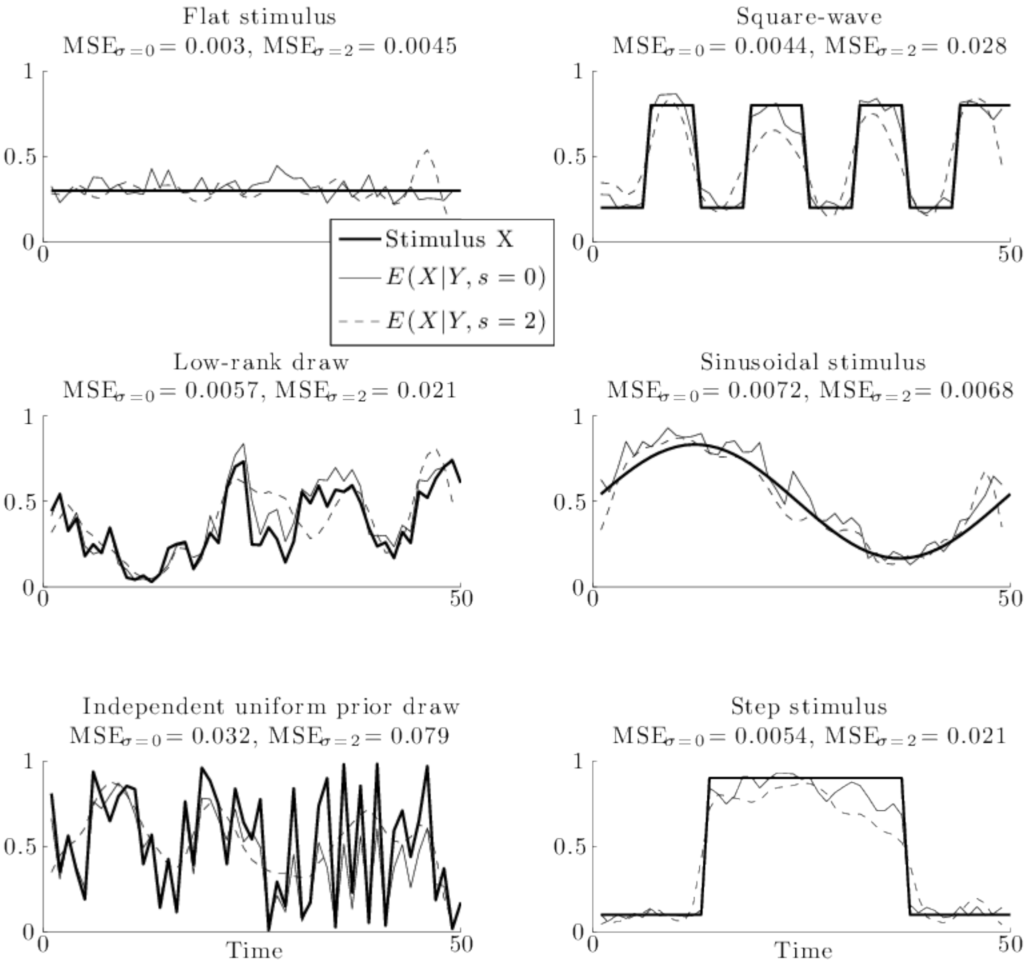

Figure 8: Reconstructions of different stimuli both for no jitter and for jitter using the low-rank prior of Eq. (6) with rank $R = 10$. The same parameter values as in Fig. 5 were used, except that 10,000 Gibbs sweeps were run after 1,000 burn-in sweeps. Here, smoother signals are more accurately estimated than they were with the independent uniform prior (see Fig. 7).

decoding. For each neuron, p_{23} was set at 0.5. We simulated 5-10 electrodes (see figure captions for specific values), on each of which two neurons were recorded. At each electrode, one of the neurons had the same transition matrix as the simple Markovian model above. The other neuron had a transition matrix identical except that $p_{31} = 1 - x_t$ and $p_{33} = x_t$ instead of $p_{31} = x_t$ and $p_{33} = 1 - x_t$. (I.e., on each electrode we simulated a neuron of ON and OFF type.) The observed feature vectors were sampled for each neuron from a Gaussian distribution with each neuron's assigned mean and covariance. Reconstruction proceeded by Rao-Blackwellized Gibbs sampling, assuming zero jitter.

A sample stimulus along with its reconstructions for varying degrees of spike cluster overlap is shown in Fig. 9. Reconstruction quality is poor when the overlap between the feature Gaussian distributions is high (i.e., when spike sorting is challenging). The Bayesian estimate, however, stays

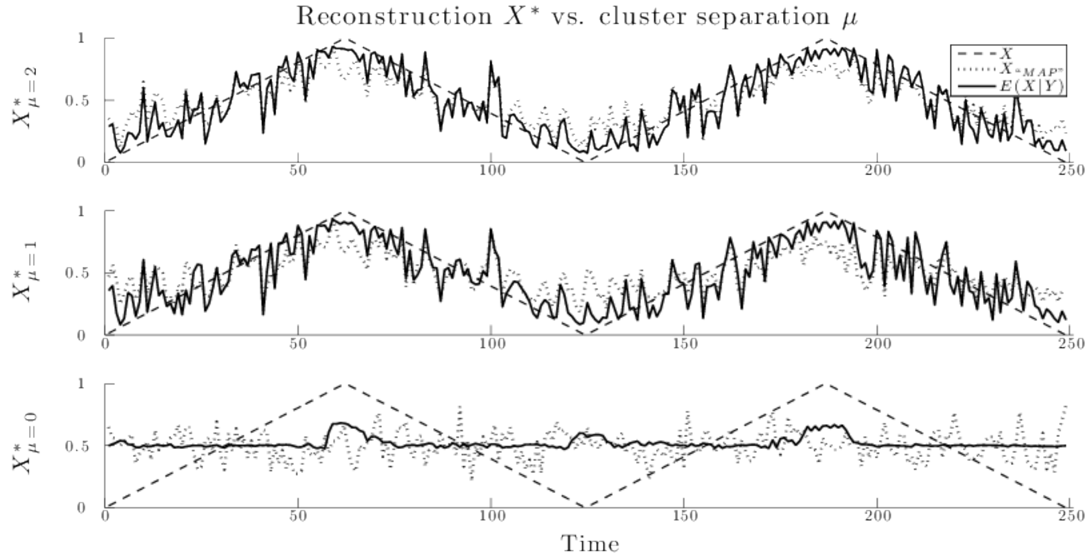

Figure 9: Sample reconstructions of a simple sawtooth stimulus for three examples of monotonically decreasing spike sorting difficulty. When spike clusters overlap completely (top panel), neither the MAP-based estimate nor the Bayesian estimate performs well. The Bayesian estimate, however, is more agnostic, providing estimates which are close to the posterior mean stimulus value. The MAP-based estimate "overcommits" to its assignments of spikes to neurons. In the case of partial overlap (middle panel), both the MAP-based and Bayesian estimates perform better, with the Bayesian estimate noticeably more accurate than the MAP-based estimate in several regions. In the case of negligible overlap (bottom panel), the three decoders produce nearly identical reconstructions. Reconstructions were conducted using 20 neurons (10 electrodes) and a uniform stimulus prior. The Gibbs sampler ran for 10,000 sweeps after a burn-in period of 1,000 sweeps.

closer to the mean, whereas the MAP-based estimate ranges widely, having committed to certain spike assignments to neurons. The same is true in the case of partial spike cluster overlap, where the reconstructions look somewhat better. The Bayesian decoder remains somewhat agnostic as to the assignment of spikes to neurons: the assignments vary from sweep to sweep of the sampler (data not shown). Meanwhile, the MAP-based decoder does not vary its assignments; since every sweep of the sampler will involve the same assignments, the reconstruction will be further from the mean than for the Bayesian decoder, with a smaller posterior variance, making this estimator "overconfident" and more frequently wrong than the fully Bayesian estimator.

We averaged the results of 100 reconstructions to compare the MSE of each decoder. Results are shown in Fig. 10. We find that that the full Bayesian decoder outperforms the Viterbi-spikes decoder. This practical example (Fig. 10) illustrates that properly accounting for spike sorting uncertainty can lead to significantly improved decoding, echoing results from earlier work (e.g., (Wood and Black, 2008; Ventura, 2008; Ventura, 2009; Chen et al., 2012)) using simpler, Poisson-based encoding models.

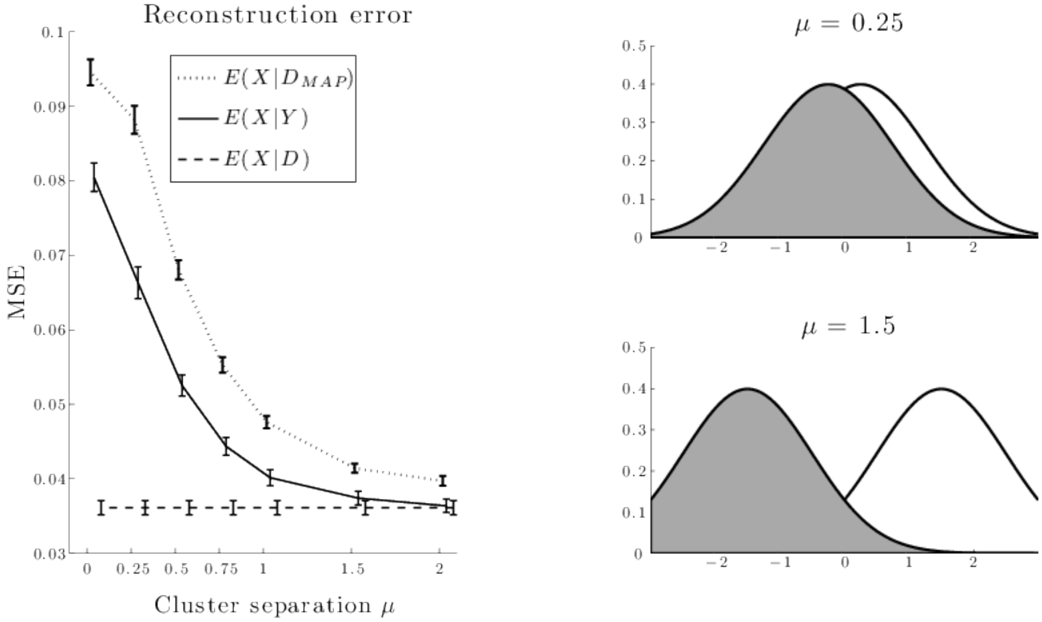

Figure 10: **Left**. Reconstruction error for the three spike-sorting decoders described in the main text. For each separation distance of the two spike clusters 100 reconstructions were performed of stimuli randomly drawn from the uniform prior over $[0, 1]^T$ incident on 10 neurons (5 electrodes) with refractory probability $p_{23} = 0.5$. The dotted line traces the error of the MAP-based estimate. The solid line traces that of the Bayes estimate, which consistently outperforms the MAP-based decoder. For reference, the performance of an optimal decoder that knows the correct spike assignments is included (dashed line). Samplers were run for 10,000 sweeps after burn-in periods of 1,000 sweeps. **Right**. Illustration of the degree of overlap in one-dimensional Gaussians with with unity standard deviation.

6 Conclusion and extensions

We have described methods for optimal Bayesian decoding of noisy or corrupted spike train data, and for quantification of the information lost due to several different possible sources of noise, including uncertainty in neural identity, spike deletion or insertion, and loss of temporal resolution due to noise or low-temporal-resolution recording techniques. Our methods allow us to quantify the loss of decoding accuracy as a function of various biophysical and experimental variables of interest, including the jitter amplitude, stimulus frequency content (cf. (Goldwyn et al., 2010)), the number of neurons, firing rates, the refractoriness of the observed neurons, and so on. One practical example (Fig. 10) illustrates that properly accounting for spike soting uncertainty can lead to significantly improved decoding, echoing results from earlier work (e.g., (Wood and Black, 2008; Ventura, 2008; Ventura, 2009; Chen et al., 2012)) using simpler, Poisson-based encoding models.

A couple directions for potential future work are clear. First, we have focused on the case of neural populations which fire in a conditionally independent manner given the stimulus. Such an assumption might be sensible in the analysis of spike train data obtained via extracellular recordings where electrodes are relatively widely spaced. However, we have made this assumption here purely for the sake of clarity and simplicity. A good deal of recent work has focused on models which can account for additional dependence structure; see, e.g., (Vidne et al., 2012) for a recent review of this literature. It would be natural to extend our framework to handle models of this type, perhaps via the efficient blockwise-Gibbs samplers discussed in (Mishchenko and Paninski, 2011). Second, while we have focused on a model-based approach here, there is a long history of more nonparametric jitter-based approaches for addressing hypotheses about the importance of temporal precision in the nervous system; see (Amarasingham et al., 2012) for a nice recent review. It would be interesting and valuable to explore further links between these nonparametric approaches and the parametric, decoding-oriented approach we have taken here. Finally, (Naud and Gerstner, 2012) consider the problem of stimulus decoding given observations of summed spike counts from a population of identical neurons. This can be considered a special case of our spike identity loss setting. It would be interesting to investigate the possibility of combining their analytical approaches with our MCMC-based techniques developed here.

Declaration of interest

We certify that there is no financial or personal conflict of interest with any person or organization regarding the material discussed in this manuscript.

Acknowledgements

This work was supported by an NSF CAREER grant, a McKnight Scholar award, NSF grant IIS-0904353, NEI grant EY018003, and by the Defense Advanced Research Projects Agency (DARPA) MTO under the auspices of Dr. Jack Judy, through the Space and Naval Warfare Systems Center, Pacific Grant/Contract No. N66001-11-1-4205. C. Smith is supported by an NSF GRFP fellowship. The authors would like to thank Yashar Ahmadian and Eric Shea-Brown for helpful conversations, Alex Ramirez and Kamiar Rahnama Rad for helpful feedback, and Jong Bahk and Chintan Gohil for their help with an early version of the work presented here. We also gratefully acknowledge the use of the Hotfoot shared cluster computer at Columbia University.

References

Ahmadian, Y., Pillow, J., and Paninski, L. (2011). Efficient Markov Chain Monte Carlo methods for decoding population spike trains. Neural Computation, 23:46–96.

Aldworth, Z., Miller, J., Gedeon, T., Cummins, G., and Dimitrov, A. (2005). Dejittered spike-conditioned stimulus waveforms yield improved estimates of neuronal feature selectivity and spike-timing precision of sensory interneurons. Journal of Neuroscience, 25:5323–5332.

Amarasingham, A., Harrison, M. T., Hatsopoulos, N. G., and Geman, S. (2012). Conditional modeling and the jitter method of spike re-sampling. Journal of Neurophysiology, 107:517–531.

Calabrese, A. and Paninski, L. (2011). Kalman filter mixture model for spike sorting of non-stationary data. Journal of neuroscience methods, 196:159–69.

Casella, G. and Berger, R. (2001). Statistical Inference. Duxbury Press.

Chen, Z., Kloosterman, F., Layton, S., and Wilson, M. A. (2012). Transductive neural decoding for unsorted neuronal spikes of rat hippocampus. In Proc. IEEE EMBC ’12, pp. 1310-1313, San Diego, CA.

Chen, Z., Vijayan, S., Barbieri, R., Wilson, M. A., and Brown, E. N. (2009). Discrete-and continuous-time probabilistic models and algorithms for inferring neuronal up and down states. Neural Comput., 21:1797–1862.

Cossart, R., Aronov, D., and Yuste, R. (2003). Attractor dynamics of network up states in the neocortex. Nature, 423:283–288.

Escola, S., Fontanini, A., Katz, D., and Paninski, L. (2011). Hidden Markov models for the stimulus-response relationships of multistate neural systems. Neural Computation, 23:1–62.

Faisal, A. and Laughlin, S. (2007). Stochastic simulations on the reliability of action potential propagation in thin axons. PLoS Comput Biol, 3(5).

Faisal, A. A., White, J. A., and Laughlin, S. B. (2005). Ion-channel noise places limits on the miniaturization of the brainï¿œs wiring. Curr Bio, 15:1143–1149.

Fruhwirth-Schnatter, S. (2006). Finite Mixture and Markov Switching Models. Springer.

Goldwyn, J. H., Shea-Brown, E., and Rubinstein, J. T. (2010). Encoding and decoding amplitude-modulated cochlear implant stimuli - a point process analysis. Journal of Computational Neuroscience, 28(3):405–424.

Gollisch, T. (2006). Estimating receptive fields in the presence of spike-time jitter. Network, 17(2):103–129.

Goodman, I. and Johnson, D. (2008). Information theoretic bounds on neural prosthesis effectiveness: The importance of spike sorting. In Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, pages 5204–5207.

Haslinger, R., Klinkner, K., and Shalizi, C. (2010). The computational structure of spike trains. Neural computation, 22:121–157.

Herbst, J. A., Gammeter, S., Ferrero, D., and Hahnloser, R. (2008). Spike sorting with hidden markov models. Journal of Neuroscience Methods, 174(1):126 – 134.

Hill, D. N., Mehta, S. B., and Kleinfeld, D. (2011). Quality metrics to accompany spike sorting of extracellular signals. Journal of Neuroscience, 31(24):8699–8705.

Horn, R. A. (1986). Topics in matrix analysis. Cambridge University Press, New York, NY, USA.

Kass, R., Ventura, V., and Brown, E. (2005). Statistical issues in the analysis of neuronal data. Journal of neurophysiology, 94:8–25.

Lewicki, M. (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. Network: Computation in Neural Systems, 9:R53–R78.

Mishchenko, Y. and Paninski, L. (2011). Efficient methods for sampling spike trains in networks of coupled neurons. Annals of Applied Statistics, 5:1893–1919.

Mishchenko, Y., Vogelstein, J., and Paninski, L. (2011). A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. Annals of Applied Statistics, 5(2B):1229–1261.

Naud, R. and Gerstner, W. (2012). Coding and decoding with adapting neurons: A population approach to the peri-stimulus time histogram. Plos Computational Biology, 8(10).

Nossenson, N. and Messer, H. (2011). Optimal sequential detection of stimuli from multi-unit recordings taken in densely populated brain regions. Neural Computation, 24(4):895–938.

Ohki, K., Chung, S., Ch'ng, Y., Kara, P., and Reid, C. (2005). Functional imaging with cellular resolution reveals precise micro-architecture in visual cortex. Nature, 433:597–603.

Paninski, L., Pillow, J., and Lewi, J. (2007). Statistical models for neural encoding, decoding, and optimal stimulus design. Progress in brain research, 165:493–507.

Pillow, J., Ahmadian, Y., and Paninski, L. (2011). Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. Neural Computation, 23(1):1–45.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2).

Robert, C. P. and Casella, G. (2005). Monte Carlo Statistical Methods (Springer Texts in Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Sahani, M. (1999). Latent variable models for neural data analysis. PhD thesis, California Institute of Technology.

Smith, C., Wood, F., and Paninski, L. (2012). Low rank continuous-space graphical models. In Proc. 15th Intl. Conf. on Artificial Intelligence and Statistics (AISTATS), pages 1064–1072.

Tokdar, S., Xi, P., Kelly, R. C., and Kass, R. E. (2010). Detection of bursts in extracellular spike trains using hidden semi-markov point process models. J. Comput. Neurosci., 29:203–212.

Toyoizumi, T., Rad, K., and Paninski, L. (2009). Mean-field approximations for coupled populations of generalized linear model spiking neurons with Markov refractoriness. Neural computation, 21(5):1203–1243.

Truccolo, W., Eden, U., Fellows, M., Donoghue, J., and Brown, E. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects. Journal of Neurophysiology, 93:1074–1089.

Ventura, V. (2008). Spike train decoding without spike sorting. Neural computation, 20(4):923–963.

Ventura, V. (2009). Automatic spike sorting using tuning information. Neural Computation, 21:2466–2501.

Vidne, M., Ahmadian, Y., Shlens, J., Pillow, J., Kulkarni, J., Litke, A., Chichilnisky, E., Simoncelli, E., and Paninski, L. (2012). Modeling the impact of common noise inputs on the network activity of retinal ganglion cells. J Comput Neurosci, 33:97–121.

Wood, F. and Black, M. (2008). A nonparametric Bayesian alternative to spike sorting. Journal of Neuroscience Methods, 173:1–12.