Statistical models for neural encoding, decoding, and optimal stimulus design[*]

Liam Paninski[1], Jonathan Pillow[2], and Jeremy Lewi[3]

[1] Department of Statistics and Center for Theoretical Neuroscience, Columbia University
liam@stat.columbia.edu; http://www.stat.columbia.edu/∼liam

[2] Gatsby Computational Neuroscience Unit, University College London
pillow@gatsby.ucl.ac.uk

[3] Georgia Institute of Technology
jlewi@gatech.edu

August 30, 2006

Abstract

There are two basic problems in the statistical analysis of neural data. The "encoding" problem concerns how information is encoded in neural spike trains: can we predict the spike trains of a neuron (or population of neurons), given an arbitrary stimulus or observed motor response? Conversely, the "decoding" problem concerns how much information is in a spike train: in particular, how well can we estimate the stimulus that gave rise to the spike train?

This chapter describes statistical model-based techniques that in some cases provide a unified solution to these two coding problems. These models can capture stimulus dependencies as well as spike history and interneuronal interaction effects in population spike trains, and are intimately related to biophysically-based models of integrate-and-fire type. We describe flexible, powerful likelihood-based methods for fitting these encoding models and then for using the models to perform optimal decoding. Each of these (apparently quite difficult) tasks turn out to be highly computationally tractable, due to a key concavity property of the model likelihood. Finally, we return to the encoding problem to describe how to use these models to adaptively optimize the stimuli presented to the cell on a trial-by-trial basis, in order that we may infer the optimal model parameters as efficiently as possible.

Introduction

The neural coding problem is a fundamental question in systems neuroscience [26]: given some stimulus or movement, what is the probability of a neural response? For example, can we predict the activity of a population of neurons in response to a given visual stimulus? Conversely, can we decode this neural activity — e.g., can we reconstruct the image that the eye is actually seeing at any given moment, given only a few observed spike trains? What information is discarded in the neural code, and what features are most important? These questions are central both for our basic understanding of neural information processing and for engineering "neural prosthetic" devices that can interact with the brain directly [7]. The problem is difficult both because neural responses are stochastic and because we want to identify these response properties given any possible stimulus in

some very large set (e.g., all images that might occur in the world), and there are typically many more such stimuli than we can hope to sample by brute force. Thus the neural coding problem is fundamentally *statistical*: given a finite number of samples of noisy physiological data, how do we estimate, in a global sense, the neural codebook?

This basic question has taken on a new urgency as neurophysiological recordings allow us to peer into the brain with ever greater facility: with the development of fast computers, cheap memory, and large-scale multineuronal recording and high-resolution imaging techniques, it has become feasible to directly observe and analyze neural activity at a level of detail that was impossible even ten years ago. Experimentalists now routinely record from hundreds of neurons simultaneously, providing great challenges and opportunities for computational neuroscientists and statisticians. Indeed, it has become clear that, just as in systems biology more generally, sophisticated statistical techniques are necessary to understand the neural code: many of the key questions quite simply cannot be answered without powerful statistical tools. Conversely, many classical statistical methods are unenlightening when applied blindly to neural data; the most successful statistical models of the neural code incorporate increasingly detailed knowledge of biophysics and functional neuroanatomy.

This chapter summarizes some recent advances in model-based techniques for the analysis of spike train data. We begin (section 1) by describing models of spike trains which have been developed to solve the neural "encoding" problem — the prediction of spike train responses (from one or multiple neurons simultaneously) given novel stimuli. In section 2 we show how to use these encoding models to optimally decode population spike trains. The emphasis in each case is to employ well-justified, flexible, likelihood-based methods for model fitting and inference. Finally, section 3 brings us back to the encoding problem, describing how we can apply the ideas developed in the first two sections to adaptively choose the optimal stimuli for characterizing the response function.

1 Neural encoding models

A neural "encoding model" is a model that assigns a conditional probability, $p(D|\vec{x})$, to any possible neural response D (for our purposes, D will represent an instantiation of a spike train, or of a population of spike trains), given an observed stimulus $\vec{x}$. The vector $\vec{x}(t)$ could include the stimulus presented at time t, or more generally the concatenated spatiotemporal stimulus history up to time t. As emphasized above, it is not feasible to directly estimate this probability $p(D|\vec{x})$ for all stimulus-response pairs $(\vec{x}, D)$; instead, therefore, we hypothesize some model, $p(D|\vec{x}, \theta)$, and then fit the model parameters θ to the observed data. Once θ is in hand we may compute the desired response probabilities as $p(D|\vec{x}) \approx p(D|\vec{x}, \theta)$; that is, knowing θ in some sense allows us to interpolate between the observed (noisy) stimulus-response pairs, in order to predict the response probabilities for novel stimuli $\vec{x}$ for which we have not yet observed any responses.

In choosing an encoding model, we must satisfy three (in some sense competing) desiderata: first, the model must be flexible and powerful enough to fit the observed data. Second, the model must be tractable: we need to be able to fit the model given the modest amount of data available in a physiological recording (preferably using modest computational resources as well); moreover, the model must not be so complex that we cannot interpret the form of the inferred parameters. Finally, the model must respect what is already known about the underlying physiology and anatomy of the system; ideally, we should be able to interpret the model parameters and predictions not only in statistical terms (e.g., confidence intervals, significance tests) but also in biophysical terms (membrane noise, dendritic filtering, etc.).

While in general there are many varieties of encoding models that could conceivably satisfy these three conditions, in this chapter we will focus on a particular model class known as the "generalized linear" model (GLM) [19, 31]. This model class is a natural mathematical representation of the basic physiological concept of a "receptive field" and has proven useful in a wide variety of experimental
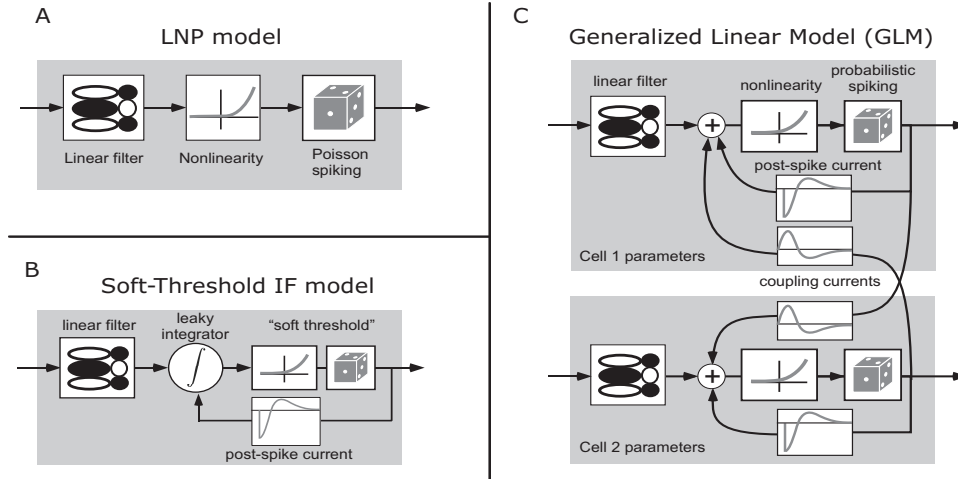
Figure 1: Schematic diagrams of some of the encoding models discussed here. **A**: the linear-nonlinear-Poisson (LNP) model is strictly feedforward, with no spike-history terms. **B**: Illustration of the connection between the GLM with spike-history terms and the integrate-and-fire cell with a probabilistic ("soft") threshold. **C**: GLM incorporating both spike-history and interneuronal coupling terms $h(.)$.

preparations. In its simplest linear-nonlinear-Poisson (LNP) form, the model hypothesizes that spike trains are produced by an inhomogeneous Poisson process with rate

$$\lambda(t) = f(\vec{k} \cdot \vec{x}(t)) \tag{1}$$

given by a cascade of two simple steps (Fig. 1a). The linear stage, $\vec{k} \cdot \vec{x}(t)$, is a linear projection of $\vec{x}(t)$, the (vector) stimulus at time t, onto the receptive field $\vec{k}$; this linear stage is then followed by a simple scalar nonlinearity $f(.)$ which shapes the output (and in particular enforces the nonnegativity of the output firing rate $\lambda(t)$). A great deal of the systems neuroscience literature concerns the quantification of these receptive field parameters $\vec{k}$.

How do we estimate the model parameters $\theta = \vec{k}$ here? We simply need to write down the likelihood $p(D|\vec{k}, \vec{x})$ of the observed spike data D given the model parameter $\vec{k}$ and the observed stimulus $\vec{x}$, and then we may employ standard likelihood optimization methods to obtain the maximum likelihood (ML) or maximum a posteriori (MAP) solutions for $\vec{k}$. It may be helpful to draw an analogy to standard linear regression here: imagine that we want to fit the standard linear regression model to our data. This model hypothesizes that each bin of observed spike train data n_t of width dt is generated by the formula $n_t = \vec{k} \cdot \vec{x}(t)dt + \epsilon_t$, where ϵ_t is discrete Gaussian white noise. If we write down the likelihood of this model using the log of the Gaussian probability density function we have

$$\log p(D|X, \vec{k}) = c_1 - c_2 \sum_t \left(n_t - (\vec{k} \cdot \vec{x}(t))dt \right)^2,$$

where c_1, c_2 are constants that do not depend on the parameter $\vec{k}$, X abbreviates the matrix of observed stimuli (the t-th row of X is given by $X_t = \vec{x}(t)$), and the sum in t is over all observed time bins. Maximizing this likelihood leads to the usual least-squares regression solution $\vec{k}_{LS} = (X^t X)^{-1} (\sum_t n_t \vec{x}(t)/dt)$. Here $X^t X$ is a scaled estimate of the covariance of $\vec{x}$, and the term on the right is proportional to the classical spike-triggered average [6, 18].

Now, if we repeat the same exercise under the more plausible assumption that spike counts per bin follow a Poisson instead of Gaussian distribution (with the rate parameter of the Poisson distribution given by Eq. (1), $\lambda(t)dt = f(\vec{k} \cdot \vec{x}(t))dt$), we have $n_t \sim Poiss[f(\vec{k} \cdot \vec{x}(t))dt]$, implying

$$\log p(D|X, \vec{k}) = c + \sum_t \left(n_t \log f(\vec{k} \cdot \vec{x}(t)) - f(\vec{x}(t) \cdot \vec{k})dt \right).$$

This likelihood no longer has a simple analytical maximizer, as in the linear regression case, but nonetheless we can numerically optimize this function quite easily if we are willing to make two assumptions about the nonlinear rectification function $f(.)$: if we assume 1) $f(u)$ is a convex function of its scalar argument u, and 2) $\log f(u)$ is concave in u, then the loglikelihood above is guaranteed to be a concave function of the parameter $\vec{k}$, since in this case the loglikelihood is just a sum of concave functions of $\vec{k}$ [19]. This implies that the likelihood has no non-global local maxima, and therefore the maximum likelihood parameter $\hat{k}_{ML}$ may be found by numerical ascent techniques. Functions $f(.)$ satisfying these two constraints are easy to think of: for example, the standard linear rectifier and the exponential function both work. Interestingly, in the exponential case $f(.) = \exp(.)$, the ML estimate $\hat{\vec{k}}_{ML}$ turns out to be closely related to the least-squares estimate $\vec{k}_{LS}$ [19], and therefore $\hat{k}_{ML}$ may be considered a generalization of classical spike-triggered average-based techniques for estimating the receptive field. See [19] for further discussion and mathematical details.

Regularization: maximum penalized likelihood

In the linear regression case it is well-known that estimates of the receptive field $\vec{k}$ based on spike-triggered averaging can be quite noisy when $\vec{k}$ has many parameters [28, 30]; the noisiness of the estimate $\vec{k}_{LS}$ is roughly proportional to the dimensionality of $\vec{k}$ (the number of parameters in $\vec{k}$ that we need to estimate from data) divided by the total number of observed samples [18]. The same "overfitting" phenomenon (noisiness increasing with number of parameters) occurs in the GLM context. A variety of methods have been introduced to "regularize" the estimated $\vec{k}$, to incorporate prior knowledge about the shape and/or magnitude of the true $\vec{k}$ to reduce the noise in $\vec{k}_{LS}$. One basic idea is to restrict $\vec{k}$ to lie within a lower-dimensional subspace; we then employ the same fitting procedure to estimate the coefficients of $\vec{k}$ within this lower-dimensional basis (model selection procedures may be employed to choose the dimensionality of this subspace [31]).

A slightly less restrictive approach is to maximize the posterior $p(\vec{k}|X, D) = (1/Z)p(D|X, \vec{k})p(\vec{k})$ (with $\vec{k}$ allowed to take values in the full original basis), instead of the likelihood $p(D|X, \vec{k})$; here $p(\vec{k})$ encodes our *a priori* beliefs about the true underlying $\vec{k}$. In the linear regression case, the computationally-simplest prior is a zero-mean Gaussian, $\log p(\vec{k}) = c - \vec{k}^t A \vec{k}/2$, where A is a positive definite matrix (the inverse covariance matrix); maximizing the corresponding posterior analytically leads directly to the regularized least-square estimator $\vec{k}_{RLS} = (X^t X + A)^{-1} \left(\sum_t n_t \vec{x}(t)/dt \right)$ [28, 30]. It is easy to incorporate this maximum *a posteriori* idea in the GLM context as well [19] (though once again we lose the nice analytic form of the optimal solution): we simply maximize

$$\log p(\vec{k}|X, D) = c + \log p(D|X, \vec{k}) + \log p(\vec{k}) = c + \sum_t \left(n_t \log f(\vec{x}(t) \cdot \vec{k}) - f(\vec{x}(t) \cdot \vec{k})dt \right) + \log p(\vec{k}).$$

Whenever $\log p(\vec{k})$ is a concave function of $\vec{k}$ (as in the Gaussian prior case described above), this "penalized" likelihood (where $\log p(\vec{k})$ acts to penalize improbable values of $\vec{k}$) is a concave function of $\vec{k}$, and ascent-based maximization may proceed (with no local maxima) as before.

Incorporating spike history effects and interneuronal interactions

Above we have described how to adapt standard spike-triggered averaging techniques for the GL model. However, it is not immediately obvious, from a physiological point of view, what we have gained by this exercise. More importantly, it is clear that this simple model suffers from a number of basic deficiencies: for example, the fact that we have assumed that the nonlinearity $f(.)$ is a convex function implies that the firing rate of our basic LNP model does not saturate: as we increase the magnitude of the stimulus $\vec{x}$, the rate must continue to increase at least linearly, whereas the firing rate of a real neuron will invariably saturate, leveling off after some peak discharge rate is attained. Moreover, neurons display a number of other related strongly nonlinear effects that are not

captured by the model: e.g., refractory effects, burstiness and bistability of responses, and firing-rate adaptation. In other words, it seems the LNP model does not satisfy our first desideratum: model (1) is insufficiently flexible to accurately model real spiking responses.

Luckily, it turns out that we may simultaneously fix these problems and greatly enhance the GLM's flexibility, by the simple trick of enlarging our input matrix X. Recall that in the discussion above, the t-th row of this matrix consisted of the stimulus $\vec{x}(t)$. However, there is no mathematical reason why we cannot incorporate other observable variables into this matrix as well. For example, appending a column of ones to X corresponds to incorporating a constant "offset" parameter b in our model, $\lambda(t) = f(\vec{k} \cdot \vec{x}(t) + b)$, which provides an important degree of flexibility in setting the threshold and baseline firing rate of the model.

More importantly, we may incorporate terms corresponding to the neuron's observed past activity n_{t-j}, to obtain models of the form $\lambda(t) = f(b + \vec{k} \cdot \vec{x}(t) + \sum_{j=1}^{J} h_j n_{t-j})$ (Fig. 1c). Depending on the shape of the "spike history filter" $\vec{h}$, the model can display all of the effects described above [22]; for example, a negative but sharply time-limited $\vec{h}$ corresponds to a refractory period (and firing rate saturation: increasing the firing rate will just increase the "hyperpolarizing" effect of the spike history terms $\sum_j h_j n_{t-j}$), while a biphasic $\vec{h}$ induces burst effects in the spike train, and a slower negative $\vec{h}$ component can enforce spike-rate adaptation. Fitting these new model parameters proceeds exactly as above: we form the (augmented) matrix X (where now $X_t = \{1 \ \vec{x}(t) \ n_{t-J} \ n_{t-J+1} \ \dots \ n_{t-1}\}$), then calculate the log-likelihood $\log p(D|X, \theta) = \sum_t (n_t \log f(X_t \cdot \theta) - f(X_t \cdot \theta)dt)$, and compute the ML or maximum *a posteriori* (MAP) solution for the model parameters $\theta = \{b, \vec{k}, \vec{h}\}$ by a concave optimization algorithm. (Note that, while we still assume that the spike count n_t within a given short time bin is drawn from a one-dimensional $Poiss(\lambda(t)dt)$ distribution, the resulting model displays strong history effects and therefore the output of the model, considered as a vector of counts $D = \{n_t\}$, is no longer a Poisson process, unless $\vec{h} = 0$.) See Fig. 2 for an application to data from a retinal ganglion cell [32, 25].

As emphasized above, while this additive form of spike-history dependence fits conveniently within our GLM framework, this choice is by no means unique. A related approach for incorporating spike-history effects is multiplicative instead [15, 3, 10]: in this multiplicative model, the firing rate at time t is given by $\lambda(t) = f(\vec{k} \cdot \vec{x}(t))r(t - t^*)$, where t^* denotes the time of the last observed spike and the function $r(.)$ encodes the spike-history effects. In general the additive (GL) and multiplicative models are distinct; for example, the basic multiplicative model only incorporates spike history information over the most recent interspike interval, whereas the additive model may incorporate information over many preceding spikes. Maximum-likelihood methods for estimating the parameters $\{\vec{k}, r(.)\}$ in this multiplicative model are discussed in [19] but are beyond the scope of this chapter; however, it is worth noting that in the case of an exponential nonlinearity $f(.) = \exp(.)$ [21, 31] the multiplicative model may be written as a one-spike modification of the GLM: $\lambda(t) = \exp(\vec{k} \cdot \vec{x}(t))r(t - t^*) = \exp[\vec{k} \cdot \vec{x}(t) + \log r(t - t^*)]$, and we may estimate $h(.) = \log r(.)$ using the likelihood optimization techniques described above, once X is defined suitably.

Finally, we may expand the definition of X to include observations of other spike trains, and therefore develop GL models not just of single spike trains, but network models of how populations of neurons encode information jointly [5, 21, 21, 31, 24]. The resulting model is summarized (Fig. 1c):

$$n_{i,t} \sim Poiss(\lambda_i(t)dt); \quad \lambda_i(t) = f\left(b + \vec{k}_i \cdot \vec{x}(t) + \sum_{i',j} h_{i',j} n_{i',t-j}\right),$$

with $\lambda_i(t)$ denoting the instantaneous firing rate of the i-th cell at time t, $\vec{k}_i$ the cell's linear receptive field, and $h_{i',j}$ a post-spike effect from the i'-th observed neuron in the population of cells from which we are recording simultaneously; these terms are summed over all past spike activity $n_{i',t-j}$. The $h_{i,j}$ terms (corresponding to the i-th cell's own past activity) plays the role of $\vec{h}$ above; the $h_{i',j}$ terms from the other cells in the population correspond to interneuronal interaction effects.
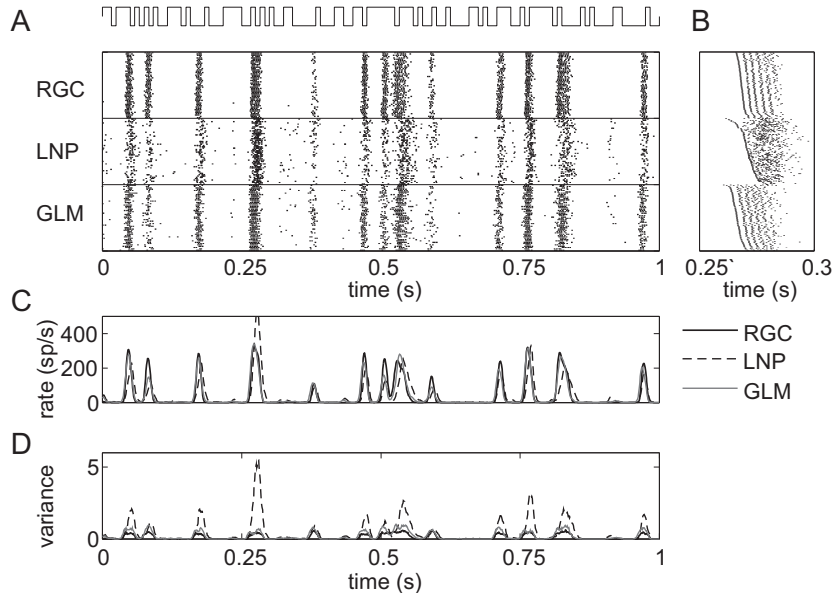
Figure 2: Example predictions of retinal ganglion ON-cell activity using the GL encoding model with and without spike history terms. Conventions are as in Figs. 3 and 4 in [25]; physiological recording details as in [32, 25]. **A**: Recorded responses to repeated full-field light stimulus (top) of true ON-cell ("RGC"), simulated LNP model (no spike history terms; "LNP"), and GL model including spike-history terms ("GLM"). Each row corresponds to the response during a single stimulus presentation. Peristimulus rate and variance histograms are shown in panels **C** and **D**, respectively. **B**: Magnified sections of rasters, with rows sorted in order of first spike time within the window in order to show spike timing details. Note that the predictions of the model including spike history terms are in each case more accurate than those of the Poisson (LNP) model. The predictions of the GLM with spike history terms are comparable in accuracy to those of the noisy integrate-and-fire model presented in [25] (PSTH variance accounted for: 91% in each case, compared to 39% for the LNP model; IF data not shown here); predictions of the multiplicative model [3, 18] are significantly less accurate (78% v.a.f.). All data shown here are cross-validated "test" data (that is, the estimated model parameters $\hat{\theta}_{ML}$ were in each case computed based on a nonoverlapping "training" data set not shown here).

Connection to biophysical models: soft-threshold integrate-and-fire models

As emphasized above, one of our key goals in constructing an encoding model is to connect the model parameters to the underlying biophysics and known physiology. Thus it is worthwhile to consider the relationship between the GLM and the more biophysically motivated models employed in studies of intracellular dynamics. One connection is provided by the following model (Fig. 1b): consider the inhomogeneous Poisson process with rate given by $f(V(t) + b)$, where f is a convex, increasing, log-concave scalar function, b is a scalar, and $V(t)$ is the solution of the "leaky-integrator" differential equation $dV/dt = -gV(t) + \vec{k} \cdot \vec{x}(t) + \sum_j h_j n_{t-j}$, with initial value $V(t_{i-1}) = V_{reset}$. Here g denotes the membrane leak conductance; as usual, in the absence of input, V decays back to 0 with time constant $1/g$. This model is conceptually identical to a simple version of the "escape-rate" approximation to the noisy integrate-and-fire (IF) model described in [8]. Since this differential equation is linear, $V(t)$ here may be written in the form $\vec{k}_g \cdot \vec{x}(t) + \vec{h}_g \cdot \vec{n}(t)$, where $\vec{k}_g$ and $\vec{h}_g$ correspond to the original parameters $\vec{k}$ and $\vec{h}$ temporally convolved with the exponential function e^{-gt}; that is, this soft-threshold IF model is just a version of the GLM, and therefore the GLM parameters may be indirectly interpreted in biophysical terms. (It is worth noting, but beyond the scope of this article,

6

that many of the same helpful concavity properties apply in the traditional, hard-threshold IF case, where noise is induced by an additive, unobserved intracellular noise current [22].)

Extensions

It should be clear that the GLM encoding framework described here can be extended in a number of important directions. We briefly describe two such directions here. First, as we have described the GLM above, it may appear that the model is restricted to including only linear dependencies on the stimulus $\vec{x}(t)$, through the $\vec{k} \cdot \vec{x}(t)$ term. However, if we modify our input matrix X once again, to include nonlinear transformations $\mathcal{F}_j(\vec{x})$ of the stimulus $\vec{x}$, it is clear that we may fit nonlinear models of the form $\lambda(t) = f(\sum_j a_j \mathcal{F}_j(\vec{x}))$ easily by maximizing the loglikelihood $\log p(D|X, \vec{a})$ with respect to the weight parameter $\vec{a}$ [33, 1]. Mathematically, the nonlinearities $\mathcal{F}_j(\vec{x})$ may take essentially arbitrary form; physiologically speaking, it is clearly wise to choose $\mathcal{F}_j(\vec{x})$ to reflect known facts about the anatomy and physiology of the system (e.g., $\mathcal{F}_j(\vec{x})$ might model inputs from a presynaptic layer whose responses are better-characterized than are those of the neuron of interest [16]).

Second, in many cases it is reasonable to include terms in X that we may not be able to observe or calculate directly (e.g., intracellular noise, or the dynamical state of the network); fitting the model parameters in this case requires that we perform a kind of average over these "latent," unobserved variables, e.g. via the expectation maximization algorithm [29, 22, 11]. While inference in the presence of these hidden parameters is beyond the scope of this chapter, it is worth noting that this type of model may be fit tractably using generalizations of the methods described here, at the cost of increased computational complexity, but the benefit of enhanced model flexibility and realism.

2   Optimal model-based spike train decoding

"Decoding" refers to the problem of how to "read out" the information contained in a set of neural spike trains, and has both theoretical and practical implications for the study of neural coding [26, 7]. A variety of statistical techniques have been applied to this problem [4]; in this section, we focus specifically on decoding methods that rely on Bayesian "inversion" of the GL encoding model discussed above. That is, we apply Bayes' rule to obtain the posterior probability of the stimulus, conditional on the observed response: $p(\vec{x}|D) = (1/Z)p(D|\vec{x})p(\vec{x})$, where $p(\vec{x})$ is the prior stimulus probability, and Z is a normalizing constant independent of $\vec{x}$[1]. The primary appeal of such Bayesian decoding methods is that they are optimal if we assume that the encoding model $p(D|\vec{x})$ is correct. Decoding therefore serves as a means for probing which aspects of the stimulus are preserved by the response, and also as a tool for comparing different encoding models. For example, we can decode a spike train using different models (e.g., including vs. ignoring spike-history effects) and examine which encoding model allows us to best decode the true stimulus [25]. Such a test may in principle give a different outcome than a comparison which focuses the encoding model's ability to predict spike train statistics. In the following, we illustrate how to decode using the stimulus which maximizes the posterior distribution $p(\vec{x}|D)$, and show how a simple approximation to this posterior allows us to estimate how much information the spike train response carries about the stimulus.

Maximum *a posteriori* decoding

The MAP estimate is the stimulus $\vec{x}$ that is most probable given the observed spike response D, i.e., the $\vec{x}$ that maximizes $p(\vec{x}|D)$. Computing the MAP estimate for $\vec{x}$ once again requires that we search a high-dimensional space (the space of all possible stimuli $\vec{x}$) to find the maximizer of a

---

[1]We used a similar idea above when we incorporated prior knowledge to regularize our estimates of the encoding model parameter θ; here we are assuming that θ, or equivalently $p(D|\vec{x})$, is known (or at least estimated to a reasonable degree of precision) and now we want use our prior knowledge of the stimulus $\vec{x}$.
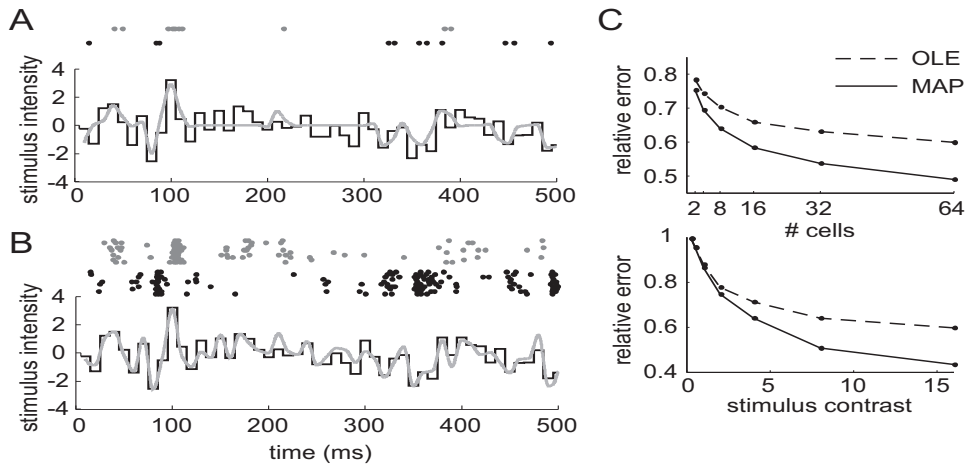
Figure 3: Illustration of MAP decoding. **A:** Simulated spike trains from a single pair of simulated ON and OFF retinal ganglion cells (above, gray and block dots) were used to compute the MAP estimate (gray) of a 500-ms Gaussian white noise stimulus (black), sampled at 100 Hz. **B:** Spike trains from 10 identical, independent ON and OFF cells in response to the same stimulus, with the associated MAP estimate of the stimulus, illustrating convergence to the true stimulus as the responses of more cells are observed. **C:** Comparison of the optimal linear estimate (OLE) and MAP estimate on simulated data, as a function of the number of observed cells (top) and stimulus contrast (variance; bottom). For each data point, the parameters of the OLE were estimated using a long run of simulated data. "Relative error" denotes the average RMS error between the true and estimated stimulus, averaged over 100 trials, divided by the RMS amplitude of the true stimulus.

nonlinear function, $p(\vec{x}|D)$. Luckily, in the GLM, the stimulus $\vec{x}$ interacts linearly with the model parameters θ, implying that concavity of the log-likelihood with respect to $\vec{x}$ holds under exactly the same conditions as does concavity in θ [19]. Moreover, the sum of two concave functions is concave, so the log-posterior, $\log p(\vec{x}|D) = \log p(D|\vec{x}) + \log p(\vec{x}) + c$, is concave as long as the stimulus log-prior $\log p(\vec{x})$ is itself a concave function of $\vec{x}$ (e.g. Gaussian). In this case, again, we may easily compute $\hat{x}_{MAP}$ by numerically ascending the function $\log p(\vec{x}|D)$.

As an empirical test of the MAP estimate, we can compare its performace with that of the optimal linear estimate (OLE), the best linear estimate of the stimulus as a function of the observed spiking data D [26]. (Note that the MAP estimate, on the other hand, is in general a *nonlinear function of D.*) Parameters of the OLE can be obtained using standard least-squares regression of the spiking data onto the stimulus $\vec{x}$ (recall, conversely, that we discussed regressing the stimulus onto the spiking data in section 1). Figure 3 shows a comparison of the two decoding techniques, given responses D generated by a GLM encoding model with known parameters, as a function of stimulus contrast (variance) and size of the neuronal population. The MAP clearly outperforms the OLE at high contrasts or large population sizes.

Assessing decoding uncertainty

In addition to providing a reliable estimate of the stimulus underlying a set of spike responses, computing the MAP estimate gives us easy access to several important quantities for analyzing the neural code. In particular, the variance of the posterior distribution around $\hat{x}_{MAP}$ tells us something about which stimulus features are best encoded by the response D. For example, along stimulus axes where the posterior has small variance (i.e. the posterior declines rapidly as we move away from $\hat{x}_{MAP}$), we have relatively high certainty that the true $\vec{x}$ is close to $\hat{x}_{MAP}$. Conversely, if the posterior

has large variance along an axis, we have relatively low certainty about this stimulus feature.

We can measure the scale of the posterior distribution along an arbitrary axis in a fairly simple manner: since we know (by the above concavity arguments) that the posterior is characterized by a single "bump," and the position of the peak of this bump is already characterized by $\hat{x}_{MAP}$, it is enough to measure the curvature of this bump at the peak $\hat{x}_{MAP}$. Mathematically, we measure this curvature by computing the "Hessian" matrix A of second-derivatives of the log-posterior, $A_{ij} = -\partial^2 \log p(x_i|D)/\partial x_i \partial x_j$. Moreover, the eigendecomposition of this matrix A tells us exactly which axes of stimulus space correspond to the "best" and "worst" encoded features of the neural response: small eigenvalues of A correspond to directions of small curvature, where the observed data D poorly constrains the posterior distribution $p(\vec{x}|D)$ (and therefore the posterior variance will be relatively large in this direction), while conversely large eigenvalues in A imply relatively precise knowledge of $\vec{x}$, i.e., small posterior variance [9] (for this reason A is referred to as the "observed Fisher information matrix" in the statistics literature).

We can furthermore use this Hessian to construct a useful approximation to the posterior $p(\vec{x}|D)$. The idea is simply to approximate this log-concave bump with a Gaussian function, where the parameters of the Gaussian are chosen to exactly match the peak and curvature of the true posterior; this Gaussian approximation makes it much easier to compute various quantities that are quite difficult to compute for general distributions $p(\vec{x}|D)$ (this approximation is quite common in the physics and statistics literature [4, 26]). Specifically,

$$p(\vec{x}|D) \approx (1/Z)e^{-(\vec{x}-\hat{x}_{MAP})^T A(\vec{x}-\hat{x}_{MAP})/2}. \tag{2}$$

Figure 4 shows a comparison of the true posterior with the Gaussian approximation, for the examples illustrated in Fig. 3; the approximation is quite accurate here.

Computing mutual information

A number of previous authors have drawn attention to the connections between the decoding problem and the problem of estimating how much information a population spike train carries about the stimulus [26, 2]. Computing mutual information is quite difficult in general, as (roughly speaking) this requires estimating joint probability distributions over $\vec{x}$ and D, which is intractable in high dimensions. However, in the case that our forward model of $p(D|\vec{x})$ is sufficiently accurate, several model-based methods make the problem tractable. We express the mutual information as:

$$I(\vec{x}; D) = H(\vec{x}) - \langle H(\vec{x}|D) \rangle, \tag{3}$$

where $H(\vec{x})$ is the entropy of the raw stimulus and $H(\vec{x}|D)$ is the conditional entropy of the stimulus given the observed spiking data D, and $\langle . \rangle$ denotes averaging over D. The first term depends only on the prior stimulus distribution $p(\vec{x})$, and represents our uncertainty about the stimulus before any responses have been observed. The second term represents the average residual uncertainty about the stimulus *after* an observation of response data D. Mutual information therefore can be seen as the average amount we learn about $\vec{x}$ given D.

The most common approach to this information-estimation problem is not to estimate the true information at all, but rather to estimate a lower bound on this quantity [26]. The idea is to take a large number of stimulus-response pairs, $\{\vec{x}_i, D_i\}$, and compute the optimal linear estimate $\hat{x}_{OLE_i}$ from each response. We then approximate the distribution of the residuals, $p(\vec{x} - \vec{x}_{OLE}|D)$, as Gaussian with mean zero and whose covariance $\hat{C}_{OLE}$ we estimate with the covariance of the OLE residuals, $\hat{C}_{OLE} = \text{cov}(\vec{x}_i - \hat{x}_{OLE_i})$. This Gaussian distribution (whose entropy is given by the formula $(1/2)\log|\hat{C}_{OLE}|$, where $|.|$ is the matrix determinant) gives a maximum-entropy approximation to $p(\vec{x} - \hat{x}_{OLE}|D)$: its entropy provides an upper bound on $\langle H(\vec{x}|D) \rangle$, and therefore a *lower* bound on $I(\vec{x}; D)$ via equation (3).
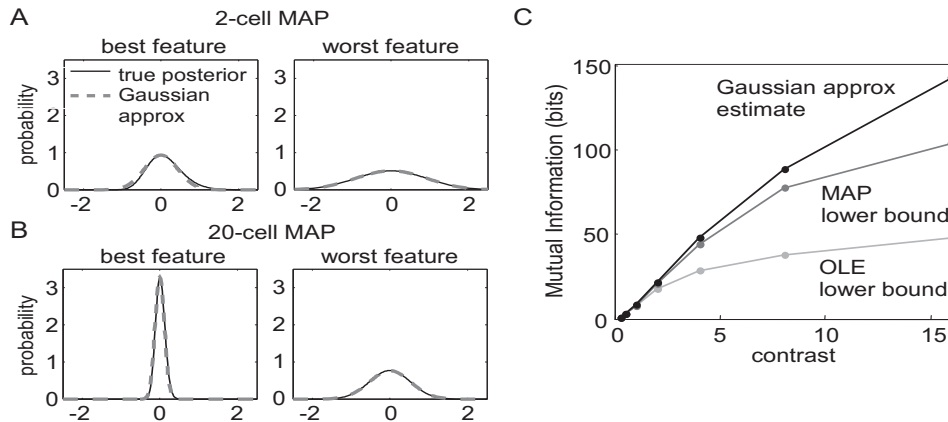
Figure 4: Comparison of the Gaussian approximation and true posterior. **A:** Slices through the true posterior $P(\vec{x}|D)$ (solid) and Gaussian approximation to the posterior (dotted), centered around the MAP estimate computed with two neural spike trains (Fig. 3a). Slices were taken along the principal axes of the posterior distribution with lowest (left) and highest (right) variance, which correspond to the "best" and "worst" encoded stimulus features (largest and smallest eigenvalues of the Hessian A, respectively). **B:** Similar plots for the true posterior and Gaussian approximation around the MAP estimate computed with 20 cells (Fig. 3b). Note that the uncertainty (i.e. variance) of the distribution is greatly reduced relative to the 2-cell MAP estimate; the Gaussian approximation is quite accurate in each case. **C:** Comparison of mutual information lower bounds computed with OLE and MAP versus the information estimated directly from the Gaussian approximation, as a function of stimulus contrast (responses generated from a 20-cell GLM). The lower bounds appear to be tight for low stimulus contrast, but significantly underestimate information at higher contrasts.

Our encoding model and MAP decoding framework suggest two approaches to improving this estimate of mutual information. First, we can simply substitute the MAP for the OLE in the above procedure. Since the MAP provides a better estimate of the stimulus (recall Fig. 3c), we can obtain a tighter upper bound on the conditional entropy with $\langle H(\vec{x}|D)\rangle = (1/2)\log|\hat{C}_{MAP}|$, where $\hat{C}_{MAP}$ is the covariance estimated from the residuals $\{x_i - \hat{x}_{MAP_i}\}$. Second, we can directly estimate the mutual information (as opposed to lower bounding it), by making use of our Gaussian approximation to $p(\vec{x}|D_i)$ for every observed response D_i [2] (recall that this approximation is quite accurate here; Fig. 4). That is, instead of estimating information with the assumption that the posterior is independent of D (i.e., that all the residuals come from the same Gaussian distribution), we can estimate the conditional entropy for every D_i, as $H(\vec{x}|D_i) = (1/2)\log|A_{D_i}^{-1}| = -(1/2)\log|A_{D_i}|$, where A_{D_i} is the Hessian of the log-posterior computed with data D_i (note that A plays the role of the inverse covariance matrix here, as in equation (2)). We can then average this over trials i to obtain an estimate for the information, $\hat{I}(\vec{x};D) = H(\vec{x}) + \langle(1/2)\log|A_{D_i}|\rangle$. Figure 4c shows a comparison of these three information estimates as a function of stimulus contrast. At low contrasts, the lower bounds derived from $\hat{x}_{OLE}$ and $\hat{x}_{MAP}$ are tight, while at higher contrasts the estimate derived from the Gaussian approximation suggests that the lower bounds significantly underestimate the true information.

Thus, the GLM encoding model enables tractable model-based decoding by simple MAP computations, and a straightforward Gaussian approximation has applications both for assessing uncertainty and estimating information-theoretic quantities. Although beyond the scope of this chapter, this approximation is also useful for testing hypotheses about changes in stimulus statistics (a.k.a. "change point detection," such as detecting a change in stimulus contrast), which requires computing integrals over the posterior $p(\vec{x}|D)$; see [23] for details. In addition, we can extend these techniques to employ more general "fully Bayesian" methods, in which we compute these integrals exactly by

Monte Carlo techniques [27] instead of using the (computationally cheaper) Gaussian approximation.

3   Optimal model-based closed-loop stimulus design

In the previous sections we have developed robust and tractable approaches to understanding neural encoding and decoding based on GL models. The framework we have developed is ultimately data-driven; both our encoding and decoding methods fail if the observed data do not sufficiently constrain our encoding model parameters θ. Therefore we will close by describing how to take advantage of the properties of the GLM to optimize our experiments: the objective is to select, in an online, closed-loop manner, the stimuli that will most efficiently characterize the neuron's response properties (Fig. 5a).

An important property of GL models is that not all stimuli will provide the same amount of information about the unknown coefficients $\vec{k}$. As a concrete example, we can typically learn much more about a visual neuron's response properties if we place stimulus energy within the receptive field, rather than "wasting" stimulus energy outside the receptive field. To make this idea more rigorous and generally applicable, we need a well-defined objective function that will rank any given stimulus according to its potential informativeness. Numerous objective functions have been proposed for quantifying the utility of different stimuli [14, 17, 13]. When the goal is estimating the unknown parameters of a model, it makes sense to choose stimuli $\vec{x}(t)$ which will on average reduce the uncertainty in the parameters θ as quickly as possible (as in the game of 20 questions), given $D = \{\vec{x}(s), n_s\}_{s<t}$, the observed data up to the current trial. If we use the entropy of the posterior distribution on the model parameters $p(\theta|\vec{x}(t), D)$ to quantify this uncertainty, we arrive at the objective function $I(\theta; n_t|\vec{x}(t), D)$, the mutual information between the response n_t and the model parameters θ given the stimulus and past data [14, 20].

So to choose the optimal stimulus $\vec{x}(t)$ at time t, we need to do two things. First, we need to compute the objective function $I(\theta; n_t|\vec{x}(t), D)$, and then we need to optimize this function with respect to the stimulus $\vec{x}(t)$. In general, both of these problems are quite difficult: computing $I(\theta; n_t|\vec{x}(t), D)$ requires an integration over the parameter space, a task whose complexity in general scales exponentially with the number of parameters, $\dim(\theta)$. Then we need to compute this difficult objective function repeatedly as we search for the optimal $\vec{x}(t)$ (a search whose difficulty, again, scales exponentially with $\dim(\vec{x})$).

Here the special structure of the GLM comes into play. We saw in the last section how a Gaussian approximation of a posterior distribution can greatly simplify the computation of the information; we use the same trick here, approximating $p(\theta|\vec{x}(t), D)$ by a Gaussian in this case instead of $p(\vec{x}|D, \theta)$ as before. (This Gaussian approximation may be justified by the same log-concavity arguments as before; moreover, asymptotic theory guarantees that this Gaussian approximation will be accurate — and moreover the MAP estimate $\vec{k}_{MAP}$ will converge to the true underlying parameter $\vec{k}$ — given a sufficiently long observation time t [20].) Computing the entropy of this posterior has therefore been reduced from an intractable integration problem to the much more tractable computation of an average log-determinant of a Hessian matrix. (The geometric interpretation is that we want to minimize the volume of the confidence ellipsoid corresponding to this posterior Gaussian.)

While much simpler than the original integration problem, the determinant computation is in general still too slow for our goal of online, closed-loop stimulus optimization. Thus we make use of one more key feature of the GLM: the log-likelihood $\log p(n_t|\vec{k}, \vec{x}(t)) = c + n_t \log f(\vec{k} \cdot \vec{x}(t)) - f(\vec{k} \cdot \vec{x}(t))dt$ depends on the $\dim(\vec{x})$-dimensional vector $\vec{k}$ only through the projection $\vec{k} \cdot \vec{x}(t)$. This effectively one-dimensional nature of the log-likelihood implies that the Hessian A_t of the log-posterior distribution given t observations is simply a rank-one perturbation of the Hessian A_{t-1} after $t-1$ observations:

$$A_t = -\partial_\theta^2 \log p(\theta|D_t) = -\partial_\theta^2 \left[\log p(\theta|D_{t-1}) + \log p(n_t|\theta, \vec{x}(t))\right] = A_{t-1} - \partial_\theta^2 \log p(n_t|\theta, \vec{x}(t)),$$

where the last term is a matrix of rank one. (The equalities above are simple manipulations with
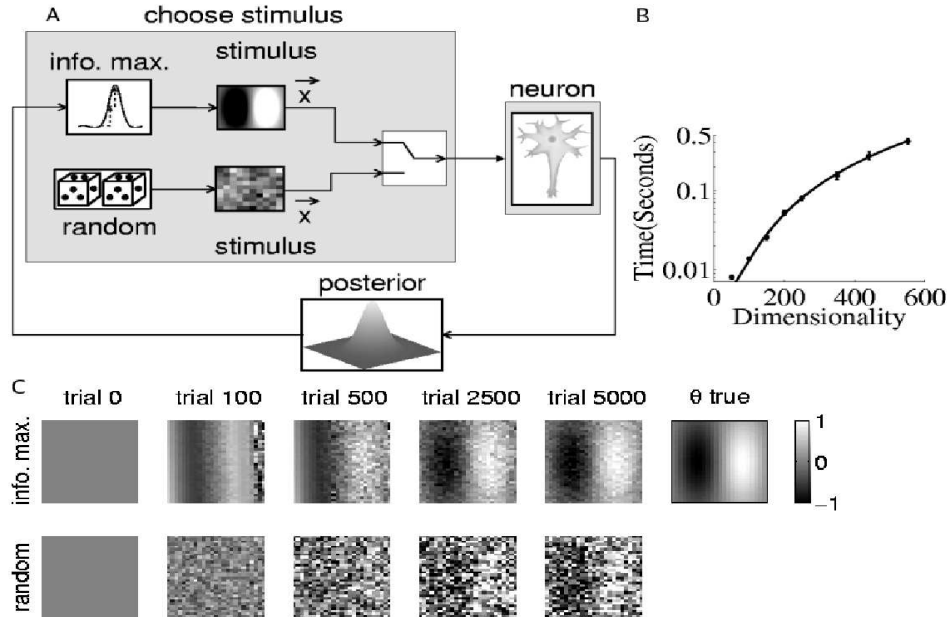
Figure 5: A) Closed-loop vs. open-loop stimulus design. B) Plot of the total running time on a desktop computer for each iteration of the model-based stimulus optimization algorithm, as a function of the dimensionality of the stimulus $\vec{x}$. A quadratic polynomial ($O(\dim(\vec{x})^2)$) fits the data quite well; note that < 15 ms are necessary to optimize a 100-dimensional stimulus. C) Plots of the estimated receptive field for a simulated visual neuron whose responses were generated by a GLM. The neuron's true receptive field $\vec{\theta}$ has the Gabor structure shown in the last panel; the nonlinearity $f(.)$ was assumed known *a priori* and the spike-history terms were assumed to be zero, for simplicity. Individual panels show $\vec{k}_{MAP}$ after observing t stimulus-response pairs (the prior $p(\vec{k})$ was taken to be Gaussian with mean zero), comparing the accuracy of the estimates using information-maximizing vs. random stimuli (all stimuli were constrained to have unit norm, $||\vec{x}||_2 = 1$ here); the closed-loop approach is an order of magnitude more efficient in this case.

Bayes rule and the definition of the Hessian.) This one-dimensional structure makes possible a very efficient recursive computation of the posterior log determinant; after making a few more simple approximations it turns out to be possible to reduce the full $\dim(\vec{x})$-dimensional optimization problem to a simple one-dimensional optimization, and this one-dimensional optimization problem can be solved numerically rapidly enough to be used online. (See [12] for a full derivation.) The entire optimization process — updating the posterior distribution, solving the one-dimensional optimiation, and choosing the corresponding optimal stimulus — is quite fast (Fig. 5b), with the running time growing only as $O(\dim(\vec{x})^2)$ (as opposed to the exponential growth in the general, non-model-based case). Figure 5c shows that the closed-loop optimization procedure leads to much more efficient experiments than does the standard open-loop approach of stimulating the cell with randomly-chosen stimuli that are not optimized adaptively for the neuron under study.

A common argument against online stimulus optimization is that neurons are highly adaptive: a stimulus which might be optimal for a given neuron in a quiescent state may quickly become suboptimal due to adaptation (in the form of short- and long-term synaptic plasticity, slow network dynamics, etc.). Including spike-history terms in the GLM allows us to incorporate some forms of adaptation (particularly those due to intrinsic processes including, e.g., sodium channel inactivation and calcium-activated potassium channels), and these spike history effects may be easily incorporated into the derivation of the optimal stimulus [12]. However, extending our results to models with more

profound sources of adaptation is an important open research direction.

In addition to fast changes due to adaptation and spike-history effects, spiking properties typically change slowly and non-systematically over the course of an experiment due to changes in the health, arousal, or attentive state of the preparation. We may handle these nonstationarities fairly easily using a Kalman filter-based approach: the idea is to incorporate the additional uncertainty due to these nonstationarities into our recursive updates of the posterior $\log p(\theta|D)$; again, see [12] for details. Future work will continue to improve the robustness and efficiency of these GLM-based methods, with the goal of applications to real-time optimized, adaptive neurophysiology experiments.

References

[1] M. Ahrens, L. Paninski, R. Petersen, and M. Sahani. Input nonlinearity models of barrel cortex responses. *Computational Neuroscience Meeting, Edinburgh*, 2006.

[2] R. Barbieri, L. Frank, D. Nguyen, M. Quirk, V. Solo, M. Wilson, and E. Brown. Dynamic analyses of information encoding in neural ensembles. *Neural Computation*, 16:277–307, 2004.

[3] M. Berry and M. Meister. Refractoriness and neural precision. *J. Neurosci.*, 18:2200–2211, 1998.

[4] E. Brown, L. Frank, D. Tang, M. Quirk, and M. Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18:7411–7425, 1998.

[5] E. Chornoboy, L. Schramm, and A. Karr. Maximum likelihood identification of neural point process systems. *Biological Cybernetics*, 59:265–275, 1988.

[6] E. de Boer and P. Kuyper. Triggered correlation. *IEEE Transactions on Biomedical Engineering*, 15:159–179, 1968.

[7] J. Donoghue. Connecting cortex to machines: recent advances in brain interfaces. *Nature Neuroscience*, 5:1085–1088, 2002.

[8] W. Gerstner and W. Kistler. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press, 2002.

[9] Q. Huys, M. Ahrens, and L. Paninski. Efficient estimation of detailed single-neuron models. *Journal of Neurophysiology*, 96:872–890, 2006.

[10] R. Kass and V. Ventura. A spike-train probability model. *Neural Comp.*, 13:1713–1720, 2001.

[11] J. Kulkarni and L. Paninski. Common-input models for multiple neural spike-train data. *COSYNE'06*, 2006.

[12] J. Lewi, R. Butera, and L. Paninski. Real-time adaptive information-theoretic optimization of neurophysiological experiments. *EMBS-06*, 2006.

[13] C. Machens. Adaptive sampling by information maximization. *Physical Review Letters*, 88:228104–228107, 2002.

[14] D. Mackay. Information-based objective functions for active data selection. *Neural Computation*, 4:589–603, 1992.

[15] M. Miller and K. Mark. A statistical study of cochlear nerve discharge patterns in response to complex speech stimuli. *Journal of the Acoustical Society of America*, 92:202–209, 1992.

[16] J. Movshon, N. Rust, M. V., and E. Simoncelli. How MT cells analyze the motion of visual patterns. *ECVP*, 2006.

[17] I. Nelken, Y. Prut, E. Vaadia, and M. Abeles. In search of the best stimulus: an optimization procedure for finding efficient stimuli in the cat auditory cortex. *Hearing Res.*, 72:237–253, 1994.

[18] L. Paninski. Convergence properties of some spike-triggered analysis techniques. *Network: Computation in Neural Systems*, 14:437–464, 2003.

[19] L. Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15:243–262, 2004.

[20] L. Paninski. Asymptotic theory of information-theoretic experimental design. *Neural Computation*, 17:1480–1507, 2005.

[21] L. Paninski, M. Fellows, S. Shoham, N. Hatsopoulos, and J. Donoghue. Superlinear population encoding of dynamic hand trajectory in primary motor cortex. *J. Neurosci.*, 24:8551–8561, 2004.

[22] L. Paninski, J. Pillow, and E. Simoncelli. Maximum likelihood estimation of a stochastic integrate-and-fire neural model. *Neural Computation*, 16:2533–2561, 2004.

[23] J. Pillow and L. Paninski. Model-based optimal decoding of neural spike trains. *CNS*06 Meeting, Edinburgh*, 2006.

[24] J. Pillow, L. Paninski, J. Shlens, E. Simoncelli, and E. Chichilnisky. Modeling multi-neuronal responses in primate retinal ganglion cells. *Comp. Sys. Neur. '05*, 2005.

[25] J. Pillow, L. Paninski, V. Uzzell, E. Simoncelli, and E. Chichilnisky. Accounting for timing and variability of retinal ganglion cell light responses with a stochastic integrate-and-fire model. *Journal of Neuroscience*, 25:11003–11013, 2005.

[26] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes: Exploring the neural code*. MIT Press, Cambridge, 1997.

[27] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2005.

[28] M. Sahani and J. Linden. Evidence optimization techniques for estimating stimulus-response functions. *NIPS*, 15, 2003.

[29] A. Smith and E. Brown. Estimating a state-space model from point process observations. *Neural Computation*, 15:965–991, 2003.

[30] D. Smyth, B. Willmore, G. Baker, I. Thompson, and D. Tolhurst. The receptive-field organization of simple cells in primary visual cortex of ferrets under natural scene stimulation. *Journal of Neuroscience*, 23:4746–4759, 2003.

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

[32] V. Uzzell and E. Chichilnisky. Precision of spike trains in primate retinal ganglion cells. *Journal of Neurophysiology*, 92:780–789, 2004.

[33] M. Wu, S. David, and J. Gallant. Complete functional characterization of sensory neurons by system identification. *Annual Review of Neuroscience*, 29(1):477–505, 2006.