Statistical analysis of neural data:
Generalized linear models for spike trains

Liam Paninski
Department of Statistics and Center for Theoretical Neuroscience
Columbia University
http://www.stat.columbia.edu/∼liam

September 30, 2013

Contents

In the previous chapter we focused on linear regression models with Gaussian noise. Generalized linear models substitute more appropriate nonnegative, discrete count models, and will allow us in the next chapter to develop a full point process theory for spike trains, instead of just trying to predict the spike count in one time bin at a time, as has been our focus in the regression-based chapters.

1 The Poisson regression model allows us to apply regression methods to nonnegative count data

Let's begin with the model $n_t \sim Poiss[\lambda_t dt]$, for the spike count n_t in a single time bin of length dt. Our first task, as usual, is to write down the likelihood in this model. This implies that

$$\log p(n_t) = n_t \log \lambda_t - \lambda_t dt + const. \tag{1}$$

Now we need to specify the rate λ_t. Let's assume, following our discussion of regression modeling, that λ_t is a function of some observed covariates X_t and model parameters θ. The simplest approach would seem to be to just let λ_t be a linear function $X_t\theta$, as in the linear regression model. However, this doesn't quite work, since λ_t must be nonnegative, and $X_t\theta$ may not be. We can fix this easily by including a nonlinearity to ensure that $\lambda_t \geq 0$, which leads us to the standard "Poisson regression model":

$$\lambda(t) = f(X_t\theta),$$

for some function $f(.) \geq 0$. (In the statistics literature, $f(.)$ is often chosen to be the exponential function $\exp(.)$, for reasons we will discuss further below, but it will be useful to consider more general choices for $f(.)$ below.) In the neuroscience literature this is often referred to as the "linear-nonlinear-Poisson" (LNP) model, since the covariate X_t is linearly weighted by θ, then plugged into the nonlinearity f, leading to the Poisson output n_t.

How do we estimate the model parameters θ here? We begin by writing down the likelihood $p(D|\theta, \vec{x})$ of the observed spike data D given the model parameter θ 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 θ. It is helpful to draw an analogy to standard linear regression here: imagine that we want to fit the standard linear regression model to our data. Recall that this model hypothesizes that each bin of observed spike train data n_t of width dt is generated according to $n_t = \theta \cdot \vec{x}(t)dt + \epsilon_t$, 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

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

where c_1, c_2 are constants that do not depend on the parameter θ. Maximizing this likelihood gives the usual least-squares regression solution $\theta_{LS} \propto (X^t X)^{-1} X^T \vec{n}$.

Now, if we repeat the same exercise under the more plausible assumption that spike counts per bin follow a Poisson instead of Gaussian distribution (under the LNP model), we have

$$n_t \sim Poiss[\lambda(t)dt] = Poiss[f(\theta \cdot \vec{x}(t))dt],$$

implying

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

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 (upward-curving) function of its scalar argument u, and 2) $\log f(u)$ is concave (downward-curving) in u, then the loglikelihood above is guaranteed to be a concave function of the parameter θ, since in this case the loglikelihood is just a sum of concave functions of θ (Paninski, 2004). 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. For more examples, see (Paninski, 2004); it turns out that any such function $f(u)$ must be monotonic and must decay exponentially or faster as $f(u) \to 0$. (Note that the convexity assumption on $f(.)$ appears problematic here, since convex functions can not saturate, but we know that real neurons have finite upper bounds on their firing rates enforced by the absolute refractory period. We will address this concern by incorporating spike-history effects in the firing rate in a later subsection, once the basic theory has been introduced.)

To optimize this loglikelihood, it is useful to compute the gradient and Hessian (second derivative matrix) of this function with respect to the model parameters θ. It's worth writing out these quantities here, in order to point out some useful analogies to the linear regression setting. Define the vectors $\vec{f}^{(i)}$ and $\vec{g}^{(i)}$ as

$$f_t^{(i)} = \frac{\partial^i}{\partial s^i} f(s) \bigg|_{s=\theta^T \vec{x}(t)}$$

and

$$g_t^{(i)} = n_t \frac{\partial^i}{\partial s^i} \log f(s) \bigg|_{s=\theta^T \vec{x}(t)}.$$

Then it is not hard to see that

$$\nabla_\theta \log p(D|X, \theta) = \nabla_\theta \left(c + \sum_t \left(n_t \log f(\theta \cdot \vec{x}(t)) - f(\vec{x}(t) \cdot \theta) dt \right) \right) = X^T (\vec{g}^{(1)} - \vec{f}^{(1)} dt),$$

and similarly,

$$J \equiv \nabla\nabla_\theta \log p(D|X, \theta) = X^T diag[\vec{g}^{(2)} - \vec{f}^{(2)} dt] X.$$

Note that our log-concavity and convexity condition of $f(.)$ guarantee the nonpositivity of $\vec{g}^{(2)}$ and $-\vec{f}^{(2)}$, respectively, and clearly whenever $(\vec{g}^{(2)} - \vec{f}^{(2)} dt)$ is guaranteed to be nonpositive then the Hessian J is guaranteed to be negative semidefinite; this furnishes another proof of the concavity of the log-likelihood in this model.

If we examine the Newton search direction[1] in this model, we find a familiar form:

$$J^{-1} \nabla = \left(X^T diag[w^{(2)}] X \right)^{-1} X^T w^{(1)},$$

[1] Recall the Newton-Raphson method for optimization of a smooth, concave function $f(\vec{x})$: this approach

where we have abbreviated the weight vector $\vec{w}^{(i)} = \vec{g}^{(i)} - \vec{f}^{(i)}dt$. This may be considered a "weighted" version of the familiar least-squares form $(X^TX)^{-1}(X^T\vec{n})$, where the weights $\vec{w}^{(i)}$ are recomputed iteratively, with each Newton update step. Thus Newton's method for fitting generalized linear models is often referred to as iteratively reweighted least squares (IRLS or IRWLS) (McCullagh and Nelder, 1989).

The theory of the statistical efficiency of GLM estimators mirrors the theory in the linear-Gaussian case, in many respects. In the GLM, the single-observation quadratic loglikelihood term from the linear-Gaussian model, $-\frac{1}{2\sigma^2}(X_t^T\theta - n_t)$, is replaced by the single-observation Poisson loglikelihood $n_t \log f(X_t^T\theta) - f(X_t^T\theta)dt$. Both functions are concave and both depend on θ only through the "rank-1" projection X_t^T. Thus the resulting picture and corresponding geometric intuition in terms of intersection of soft constraints are quite similar. The analog of the $\frac{1}{2\sigma^2}X^TX$ matrix, which sets the inverse covariance of the estimate θ in the linear regression case, is the "observed Fisher information" matrix, which is simply $-J$, the negative of the Hessian matrix defined above, evaluated at the MLE. We will discuss this matrix in much more depth below. Similarly, we can penalize the GLM loglikelihood with concave penalty functions to combat overfitting exactly as in the linear-Gaussian case.

Once we have obtained a model estimate, how do we decide how good the model is compared to other potential models? We will discuss model checking and goodness-of-fit in more depth after we have introduced some relevant point process theory, but for now we note that one helpful model score is the cross-validated loglikelihood. In this context it is important to normalize the loglikelihood to obtain a more meaningful number for comparisons; if the loglikelihood under the model may be written as

$$\log p(D|X,\theta) = \sum_t \log p(n_t|X_t^T\theta)$$

then it is useful to normalize by the number of time bins T/dt and to subtract off a measure of how predictable the firing rate is *a priori*, i.e., compute

$$\frac{1}{T/dt}\sum_t \log p(n_t|X_t^T\theta) - \frac{1}{T/dt}\sum_t \log p(n_t) = \frac{1}{T/dt}\sum_t \log\frac{p(n_t|X_t^T\theta)}{p(n_t)} \approx I(n_t; X_t^T\theta),$$

where $p(n_t)$ is the marginal probability of observing the response n_t in time bin t. The approximate equality follows by the law of large numbers; the right-hand side is the mutual (Shannon) information I between the response per bin n and the projected predictor $X_t^T\theta$ (Cover and Thomas, 1991). This information is measured in a standardized unit (bits), and is a natural way to quantify how predictable n_t is given X_t and the model parameters θ. (Note that I here depends on the size of the time bin dt; I is roughly proportional to dt for small enough dt.)

is based on the second-order Taylor expansion of $f(\vec{x})$ about the current guess $\vec{x}_0$ at the best $\vec{x}$:

$$\begin{aligned}\arg\max_{\vec{x}} f(\vec{x}) &\approx \arg\max_{\vec{x}} \Big(f(\vec{x}_0) + \nabla_{\vec{x}} f(\vec{x}_0)^T(\vec{x}-\vec{x}_0) + \frac{1}{2}(\vec{x}-\vec{x}_0)\nabla\nabla_{\vec{x}} f(\vec{x}_0)^T(\vec{x}-\vec{x}_0)\Big)\\ &= \vec{x}_0 - \big(\nabla\nabla_{\vec{x}} f(\vec{x}_0)\big)^{-1}\nabla_{\vec{x}} f(\vec{x}_0).\end{aligned}$$

Thus, iteratively setting $\vec{x}_{i+1}$ to be the minimizer of the objective function in the direction defined by a solution $\vec{u}$ to the linear equation

$$-\big(\nabla\nabla_{\vec{x}} f(\vec{x}_i)\big)\vec{u} = \nabla_{\vec{x}} f(\vec{x}_i)$$

is known to lead to an efficient method for optimizing $f(\vec{x})$.

2 Estimation of time-varying firing rates

With the above general comments out of the way, we'll start by focusing on a rather classical question as an important special case: how do we estimate the time-varying firing rate of a single neuron, observed over multiple trials, given repeated presentations of a fixed stimulus? The standard method for estimating a firing rate in this setting is to construct the peri-stimulus time histogram (PSTH). As we will see, we can use GLM methods to generalize and improve upon these classical techniques.

2.1 The simplest histogram binning approach can be interpreted in the context of the Poisson regression model

Assume that we observe spike counts over N trials indexed by i, each with multiple time bins indexed by t, with each spike count $n_t^{(i)}$ drawn independently from $n_t^{(i)} \sim Poiss[\lambda_t dt]$. Note that the firing rate λ_t depends on t but not i here. Then we can write the loglikelihood as

$$\begin{aligned} \log p(\{n_t^{(i)}\}) &= \sum_{it} [n_t^{(i)} \log \lambda_t - \lambda_t dt] + const. & (2) \\ &= \sum_t [\sum_i n_t^{(i)}] \log \lambda_t - N\lambda_t dt + const. & (3) \end{aligned}$$

An interesting implication of the this equation is that the PSTH $\sum_{i=1}^N n_t^{(i)}$ is a sufficient statistic for this model: data from different trials can be combined by simply adding spikes, without any loss of information (since the loglikelihood only depends on $\{n_t^{(i)}\}$ through this sum). This is a very special feature of the Poisson spiking model; it is not true of more general point process data!

To make a connection with the Poisson regression model we need to define λ_t here in terms of some covariate matrix X and a corresponding parameter vector θ. This is easy: set X_t to the t-th unit vector (with a 1 in the t-th bin and zeros everywhere else). Then the loglikelihood breaks up into an independent sum over each time bin t, which means that if we want to compute the MLE for θ we can simply optimize each element θ_t independently of the rest. We find that the MLE for λ_t can be computed analytically, and is simply the normalized PSTH,

$$\hat{\lambda}_t^{MLE} = \frac{1}{Ndt} \sum_{i=1}^N n_t^{(i)}.$$

The choice of the binwidth dt here is important, and affects the smoothness of the estimated $\hat{\lambda}_t^{MLE}$: when dt is small, the estimate tends to follow the data closely, but is very rough (due to the random noise in the sum over $n_t^{(i)}$), while when dt is large the estimate averages over more time and therefore becomes less noisy, but may incorrectly coarsen fine temporal variations in the true underlying rate λ_t. Thus binwidth selection involves a "bias versus variance" tradeoff: small dt reduces the bias of the estimate (and increases its variance) while large dt reduces the variance (but increases the bias). To obtain a consistent estimator, dt should decrease as the information available in the data grows: more informative data allows us to fit richer models. Asymptotic expressions for the bias and variance are may be easily derived here (we leave these computations as an exercise); we find that dt should not go to zero faster than $1/N$, or else we won't have a consistent estimator (since we need to average over an increasing number of spikes within each bin in order to guarantee that the variability in the estimate eventually goes to zero).

2.2 Local likelihood and kernel smoothing

Histogram estimates are necessarily "blocky": they are constant within timebins and discontinuous at the timebin boundaries. Many methods are available to obtain smoother estimators. One approach that provides a conceptual bridge between likelihood-based methods (which will be our focus) and more intuitive, simple smoothing methods (such as simply applying some kind of post-hoc smoothing filter to the PSTH) is based on "local likelihood" methods. The idea is that, for each time t, we can fit a Poisson regression model $\lambda_t = f(X_t\theta_t)$ for appropriately chosen covariates X_t, just as before, but now note that we fit a different parameter θ_t for each desired time t, and we use a different objective function. Rather than using the full loglikelihood over all times t to fit θ_t, we instead optimize a locally-weighted version of the loglikelihood:

$$\hat{\theta}_t^{local} = \arg\max_{\theta_t} \sum_s \left[w(s-t) \left(\sum_i n_s^{(i)} \right) \log f(X_s\theta_t) - N f(X_s\theta_t) \right] ds; \tag{4}$$

this is almost the same as the full loglikelihood, except we have included a weighting function $w(.) \geq 0$ which is centered at $s - t = 0$ (i.e., $w(u)$ peaks at $u = 0$ and decreases to zero as $|u|$ grows). This term up-weights observations near t, and allows us to ignore observations at times s far from t.

In general it may be expensive to compute a new $\hat{\theta}_t^{local}$ for each desired value of t, but in some cases the computations simplify considerably. For example, if we set $X = 1$ and $w(.)$ integrates to one, then the resulting rate estimate can be computed analytically:

$$\hat{\lambda}_t^{local} = \frac{\int_0^T w(s-t) \frac{1}{N} \sum_i n_s^{(i)} ds}{\int_0^T w(s-t) ds}. \tag{5}$$

Here T is the length of the trial. If $0 \ll t \ll T$, then the denominator is close to one, while if $w(u)$ is symmetric around $u = 0$, then the numerator is simply a convolution of w with the normalized PSTH $\frac{1}{N}\sum_i n_s^{(i)}$; this convolution can in turn be computed efficiently using fast Fourier-domain techniques. This approach is often referred to as "kernel smoothing," with $w(.)$ the "kernel" function. (There is a distant connection between these kernels and the "kernel trick" we discussed previously, but in practice these two types of kernel approaches are fairly distinct and should not be confused.)

How to choose $w(.)$ here? It is convenient to parameterize $w(.)$ by some "bandwidth" parameter h:

$$w(u) = \frac{1}{h} w_0 \left(\frac{u}{h} \right),$$

where w_0 is a fixed nonnegative symmetric function that integrates to one and h is the "bandwidth" parameter that sets the width of $w(.)$. For example, if w_0 is chosen to be a standard Gaussian probability density function (pdf), then w corresponds to a Gaussian pdf with standard deviation h. This bandwidth parameter plays a similar role as dt in the histogram setting: small h means that we average only very locally near t, reducing bias but increasing the variance of the estimate, while while large h leads to much coarser (more biased) but smoother (less variable) estimates.

2.3 Representing time-varying firing rates in terms of a weighted sum of basis functions

So far we have considered quite simple versions of the covariate vectors X_t: letting X_t be indicator functions (equal to 1 within a timebin, and zero outside) led to the histogram estimator, and letting $X_t = 1$ in the local likelihood setting led to the kernel smoother. What other choices of X_t are convenient? Remember that we are representing the firing rate as $\lambda_t = f(X_t\theta)$ here: if we rewrite $X_t\theta = \sum_{j=1}^{p} X_t^j \theta^j$, where p is the length of the parameter vector θ, we see that $f^{-1}(\lambda_t)$ is just a weighted sum of the p time-varying functions X_t^j. Note that this representation makes the bias-variance tradeoff here quite clear: the larger the span of the basis X_t^j, the more functions λ_t we can represent exactly (i.e., the smaller the bias of our estimator), but this leads to potentially larger estimator variances.

What basis X_t^j should we choose? For computational reasons, it is useful to choose basis functions with local support: i.e., X_t^j should be zero except on a small interval. (Or more generally, we need to be able to easily transform the basis we use into such a locally-supported basis.) Recall the Newton method for optimizing the loglikelihood: we need to iteratively solve linear matrix equations of the form $Hx = \nabla$, where H is the Hessian and ∇ is the gradient of the loglikelihood. If the basis X_t^j has local support, then the basis elements can be ordered so that H is a banded matrix; in particular, the bandwidth b (the number of nonzero off-diagonal elements per row) of H will correspond to the number of basis elements X_t^j with overlapping support. Banded matrix equations $Hx = \nabla$ can be solved in time that scales as $O(b^2 p)$, whereas general matrix equations require $O(p^3)$ time; when b is small, banded Newton methods are much faster.

There are many potential smooth, locally-supported bases. “Splines” are one popular choice, with various implementations in most software packages; the basis elements here are piecewise polynomial, with the constraint that any element in the span of the spline basis must be continuous (and typically differentiable) at the “knot” points at which the piecewise polynomials are joined. The user controls the number and location of the knots: we may, for example, place more knot points in regions where we believe the function of interest may be more rapidly varying. (Again, more knots corresponds to a larger basis dimension p, which typically reduces bias and increases variance.)

Since computation time scales linearly with p in these locally-supported bases, it is common to use a large p and incorporate prior information to constrain the estimate to avoid excess variance. For example, if we believe a priori that $z_t = X_t\theta$ is a smooth function of t, then we can incorporate this information in our prior directly. A common choice is to use an (improper) log-prior of the form

$$\log p(\{z_t\}) \propto -\sum_i a_i \int_0^T \left(\frac{d^i z_t}{dt^i} \right)^2 dt,$$

for some set of coefficients a_i: larger values of a_i correspond to stronger prior constraints on the integrated variance of the i-th derivative of z_t. Thus this log-prior serves to penalize the roughness of z_t. Note that computation of the MAP estimator here remains highly tractable, since the Hessian of this log-prior remains banded. (There are also some close connections between this class of priors and “state-space” models, which we will discuss in depth later.) The prior coefficients a_i here must be chosen via some model selection criteria here, e.g., marginal likelihood, cross-validation, generalized cross-validation, etc.

It is also natural to try to adapt the basis to the observed data. For example, the firing rate might be more-or-less stationary during one part of the trial (e.g., before a sensory stimulus is presented), but vary sharply during another part (after the stimulus is presented). We can always adapt the basis by hand; the goal is to choose the span of the basis to cover the set of the firing rates λ_t that we expect to observe, while keeping the basis dimension low to reduce estimator variability and overfitting effects. There are also automated methods for basis selection; for example, the Bayesian adaptive regression spline (BARS) approach of (DiMatteo et al., 2001) uses Monte Carlo methods to place spline knots in a data-dependent manner. We will discuss these methods at more length in a later chapter, after we develop some suitable background in Monte Carlo computational methods.

Finally, similar methods may be applied to estimate firing rate maps which depend on two variables instead of just one. For example, it is often desirable to model a given neuron's activity as a function of a spatial variable instead of a function of time; see e.g. (Brown et al., 1998) for a detailed analysis of hippocampal place field activity. In the spatial case, the relevant matrices remain highly structured but are no longer simply banded; nevertheless, fast computational methods are still available in this case. See, e.g., (Rue and Held, 2005; Rahnama Rad and Paninski, 2010) for further discussion. In higher-dimensional cases (greater than three or so), the curse of dimensionality kicks in, and different techniques become necessary, as we discuss below.

3 Overdispersion, latent variables, and estimating equations*

4 Consistency of the MLE and connections to the spike-triggered average

In the above we have focused on the computational properties of GLMs. However, we have not yet said much about how good an estimator the MLE, for example, actually is: for example, does the MLE asymptotically provide the correct θ, given enough data (i.e., is the MLE "consistent" for θ? If not, how large is the asymptotic bias? We can answer both of these questions easily in the case that the "link" function f is chosen correctly (that is, when we fit the responses with a model f that corresponds exactly to the true response properties of the cell under question, and therefore only the parameter θ is considered unknown): in this (admittedly idealized) case, standard likelihood theory (Schervish, 1995) establishes that the MLE is consistent for θ, and in fact the MLE is asymptotically optimal (achieves the Cramer-Rao bound, i.e., has the smallest possible asymptotic error).

What if we do not choose the correct link function $f(.)$? It turns out consistency may be established in this case, too, under certain symmetry conditions. Assume the observed spike train is generated by a GLM with rate function g, but that we apply the MLE based on the incorrect rate function f. Our results will be stated in terms of an input probability distribution, $p(\vec{x})$, from which in the simplest case the experimenter draws independent and identically-distributed inputs $\vec{x}$, but which in general is just the "empirical distribution" of $\vec{x}$, the observed distribution of all inputs $\vec{x}$ presented to the cell during the course of the experiment. We begin with a simple but important result about the "spike-triggered average" (STA) introduced in the context of linear regression: recall that this is just proportional to the cross-correlation between the observed spikes n_t and the observed covariates X_t.

4.1 The spike-triggered average gives an unbiased estimate for the linear-nonlinear model parameters under elliptical symmetry conditions

It turns out that the STA provides a good estimator for θ under certain conditions, even though it does not require estimation of the nonlinear function $f(.)$. The notion of "elliptically symmetric" distributions $p(\vec{x})$ is key here: we say that $p(\vec{x})$ is "elliptically symmetric" if $p(\vec{x}) = q(||A\vec{x}||_2)$ for some scalar function $q(.)$, some symmetric matrix A, and the usual two-norm $||\vec{y}||_2 = (\sum_i y_i^2)^{1/2}$; that is, p is constant on the ellipses defined by fixing $||A\vec{x}||_2$. A function is "radially," or "spherically," symmetric if A is proportional to the identity, in which case the elliptic symmetries above become spherical. Multivariate Gaussian distributions with mean zero are elliptically symmetric; multivariate Gaussians with zero mean and covariance proportional to the identity are radially symmetric.

Now the key result is that if the stimulus distribution $p(\vec{x})$ is radially symmetric, then the STA is *unbiased* for θ. Recall that an estimator $\hat{\theta}$ for a parameter θ is unbiased if $E_\theta \hat{\theta} = \theta$ for all values of θ. In this case, θ is considered as a one-dimensional subspace rather than as a vector: thus, we mean that $E(\hat{\theta})$ is proportional to θ when the data are generated according to the GLM with parameter θ.) The proof of this fact is quite straightforward (Bussgang, 1952; Chichilnisky, 2001; Paninski, 2003; Schnitzer and Meister, 2003; Simoncelli et al., 2004), relying on the fact that the expectation

$$E(\hat{\theta}_{STA}) \propto \int p(\vec{x}) g(\theta^T \vec{x}) d\vec{x} \propto \theta$$

whenever $p(\vec{x})$ is radially symmetric. This may be seen by a simple symmetry argument (Chichilnisky, 2001): since the function $g(\theta^T \vec{x}) p(\vec{x})$ is symmetric around the axis θ (by the symmetry assumption on $p(\vec{x})$), the average $\int p(\vec{x}) g(\theta^T \vec{x}) d\vec{x}$ of this function must lie on the axis spanned by θ[2].

A similar result holds in the elliptically symmetric case, where $p(\vec{x}) = q(||A\vec{x}||_2)$ for some A that is not proportional to the identity. In this case a bit of algebra and a change of variables imply that $A^2 X^T \vec{n}$ — a simple linear transformation of the STA — is unbiased for θ (Paninski, 2003). An application of the law of large numbers is enough to establish consistency for this estimator: $A^2 (1/T) X^T \vec{n}$ converges to its expectation, which in turn is proportional to θ. (We may moreover establish rate of convergence results and a central limit theorem for this estimator; see (Paninski, 2003) for details.) More generally we have to estimate A from data, but since we typically may collect or generate an arbitrarily large number of samples

[2] A nice generalization to the multineuronal "multispike-triggered average" is described by (Schnitzer and Meister, 2003). Imagine we are observing two neurons simultaneously, and both neurons respond to the stimulus as independent linear-nonlinear encoders:

$$p(n_1 = 1, n_2 = 1 | \vec{x}) = f_1(\theta_1^T \vec{x}) f_2(\theta_2^T \vec{x}),$$

where θ_1 and θ_2 denote the receptive fields of cell 1 and 2, respectively. Now if $p(\vec{x})$ is radially symmetric, then an identical argument establishes that the multi-spike triggered average $E(\vec{x} | n_1 = 1, n_2 = 1)$ must lie in the subspace spanned by θ_1 and θ_2, i.e.,

$$E(\vec{x} | n_1 = 1, n_2 = 1) = a_1 \theta_1 + a_2 \theta_2$$

for some scalars a_1 and a_2. Thus if we measure $E(\vec{x} | n_1 = 1, n_2 = 1)$ experimentally and find significant departures from the subspace spanned by θ_1 and θ_2, we can reject the hypothesis that both neurons respond to the stimulus as independent linear-nonlinear encoders. This argument may clearly be extended to more than two neurons.

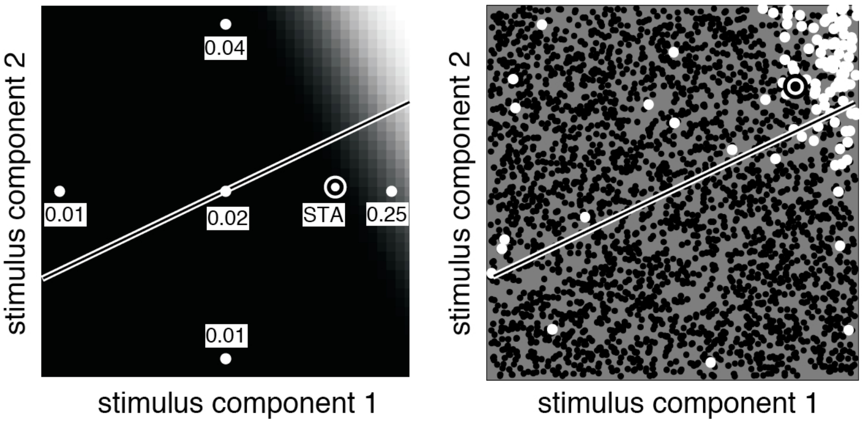

Figure 1: Simulations of an LNP model demonstrating bias in the STA for two different nonspherical stimulus distributions. The true θ is indicated by the solid line in both panels, and the firing rate function $f(.)$ is a sigmoidal nonlinearity (corresponding to the intensity of the underlying grayscale image in the left panel). In both panels, the black and white "target" indicates the recovered STA. **Left**: Simulated response to "sparse" noise. The plot shows a two-dimensional subspace of a 10-dimensional stimulus space. Each stimulus vector contains a single element with a value of ±1, while all other elements are zero. Numbers indicate the firing rate for each of the possible stimulus vectors. The STA is strongly biased toward the horizontal axis, pulled downwards by the asymmetry in $p(\vec{x})$. **Right**: Simulated response of the same model to uniformly distributed noise. The STA is now biased upwards toward the corner. Note that in both examples, the estimate will not converge to the correct answer, regardless of the amount of data collected, i.e., an asymptotic bias remains. Adapted from (Simoncelli et al., 2004).

from $p(\vec{x})$, this is straightforward: if $p(\vec{x})$ has zero mean, as we may assume without loss of generality, then the sample second moment matrix $\hat{E}(\vec{x}\vec{x}^T)$ is a consistent estimator for A^{-2}, and therefore our familiar least-squares estimator $(X^TX)^{-1}X^T\vec{n}$ is consistent for θ under the elliptical symmetry condition — without exact knowledge of $f(.)$.

But wait — we have established that the expectation of the STA is proportional to θ in the radially symmetric setting. But what if the proportionality constant is zero? In this case, the STA will converge to its expectation, and we will estimate θ to be zero — not very useful. Thus for the STA to be useful we need one additional condition:

$$\left| \int p_1(x_1) g(x_1) dx_1 \right| > 0,$$

where $x_1 = \theta^T \vec{x}$ and p_1 is the corresponding marginal distribution of this one-dimensional projection x_1. This condition guarantees that the STA will not just converge to zero.

Finally, it is interesting to note that the elliptical symmetry condition on $p(\vec{x})$ is not only sufficient for consistency of the STA estimator, but also necessary (Paninski, 2003), in the sense that if $p(\vec{x})$ is not elliptically symmetric, then there exists an $f(.)$ for which the STA estimator has a nonnegligible bias (i.e., is inconsistent) for the θ; see Fig. 1 for an illustration.

4.2 Similar symmetry conditions guarantee the consistency of the MLE, even under model misspecification

Now let's return to our discussion of the MLE. We can establish the following proposition: the MLE assuming some convex and log-concave f is consistent for any true underlying g, provided $p(\vec{x})$ is elliptically symmetric and the spike-triggered mean, $E_{p(\vec{x}|spike)}\vec{x}$, is different from zero. In other words, given our usual symmetry condition on the input distribution $p(\vec{x})$, an asymmetry condition on g, and enough data, the MLE based on f will always give us the true θ. (Note that the input distribution $p(\vec{x})$ is assumed to have mean zero, which may be enforced in general via a simple change of variables.) We present the proof here both to illustrate its simplicity and its similarity to the corresponding proof for the STA-based estimator, as discussed above.

Proof. General likelihood theory (van der Vaart, 1998) says that ML estimators, according to the law of large numbers, asymptotically maximize $E[\log p(D|X,\theta)]$, the expectation of the likelihood function under the true data distribution. We need to prove that this function has a unique maximum at $\alpha\theta_0$, for some $\alpha \neq 0$. We have

$$E[\log p(D|X,\theta)] = \int p(\vec{x}) \left[g(\theta_0 \vec{x}) \log f(\theta\vec{x}+b) - f(\theta\vec{x}+b) \right] d\vec{x}.$$

The key fact about this function is that it is concave in (θ, b) and, after suitable change of variables (multiplication by a whitening matrix), symmetric with respect to reflection about the θ_0 axis. This immediately implies that a maximizer lies on this axis (i.e., is of the form $\alpha\theta_0$, for some scalar α); the strict convexity of f or $-\log f$ implies that any such maximizer is unique. The proof that $\alpha \neq 0$ follows without too much effort; see (Paninski, 2004) for details. □

This result for the MLE bears a striking similarity to our consistency result for the STA; the conditions ensuring the asymptotic accuracy of these estimators are exactly equivalent (and by much the same symmetry argument). This leads us to study the similarities of these two methods more carefully.

We base our discussion on the solution to the equations obtained by setting the gradient of the likelihood to zero. The MLE solves

$$\frac{1}{T}\sum_i \vec{x}_i \frac{f'}{f}(\theta_{MLE}\vec{x}_i + b_{MLE}) = \int_0^T f'(\theta_{MLE}\vec{x}_t + b_{MLE})dt = \int p(\vec{x}) f'(\theta_{MLE}\vec{x} + b_{MLE})\vec{x}, \quad (6)$$

with T the length of the experiment; in the last line we have replaced an expectation over time t with an expectation over space $\vec{x}$. In the case of elliptically symmetric stimuli, as we saw in our analysis of the STA, the right hand side converges to a vector proportional to $C\theta_{MLE}$ (recall that f' is monotonically increasing), where C denotes the covariance matrix of the input distribution $p(\vec{x})$. The left hand side, on the other hand, is itself a kind of weighted STA — an average of (weighted) spike-triggered stimuli $\vec{x}$ — with the weight $\frac{f'}{f}(\theta\vec{x}_i + b)$ positive and monotonically non-increasing in $\theta\vec{x}_i$, by the log-concavity of f. (We interpret this weight as a "robustness" term, decreasing the strength of very large — possibly outlying — $\vec{x}$; see (Paninski, 2004) for more details on the robustness properties of the MLE in this model.)

Thus, denoting the left-hand side as θ_{WSTA}, for weighted STA, we have that the MLE asymptotically behaves like

$$\theta_{MLE} = C^{-1}\theta_{WSTA};$$

this is exactly analogous to $\theta_{LS} \equiv C^{-1}\theta_{STA}$, the basic correlation-corrected estimator for cascade models (Paninski, 2003). Also note that, in the case that $f(.) = \exp(.)$, the weight $\frac{f'}{f}(\theta\vec{x}_i + b)$ is constant for all $\vec{x}$. Thus, in the case of elliptically symmetric $p(\vec{x})$, θ_{LS} and the MLE under the exponential nonlinearity are asymptotically equivalent. More generally, the least-squares estimator provides a useful starting point for iterative maximization of the GLM likelihood.

4.3 Expected loglikelihood approximations can lead to much faster computation

Let's examine the loglikelihood further in the special case that $f(.) = \exp(.)$. In this case the log and the exp in the loglikelihood cancel partially, leaving us with

$$L(\theta) = \sum_t \left((X_t^T\theta)n_t - \exp(X_t^T\theta)dt \right) = \theta^T \sum_t X_t^T n_t - \sum_t \exp(x_t^T\theta)dt.$$

The first term here is easy to deal with, once we obtain $\sum_t X_t n_t$. We recognize this term as proportional to the spike-triggered average. Since the spiking data only enter into this term (not the term involving the exp(.)), we can conclude that the STA is a sufficient statistic for the spiking data in this model; in other words, once the STA has been computed, we can throw away all other details about the spike times and the likelihood will be unchanged. This simple linear structure of the first term is a special feature of the fact that we used the "canonical" link function for the Poisson model, $f(.) = \exp(.)$; our results below depend on this canonical assumption.

Evaluating $\sum_t \exp(x_t^T\theta)$ is the hard part. But recall the logic we employed in the last section: this is a big sum, independent of the spiking data, and therefore in many cases we can approximate this sum as an expectation over $p(\vec{x})$. In particular, we can define the "expected loglikelihood" (EL), denoted by $\tilde{L}(\theta)$, as an approximation to the log-likelihood that can alleviate the computational cost of the non-linear term. We invoke the law of large numbers to approximate the sum over the non-linearity by its expectation (Paninski, 2004; Field et al., 2010; Park and Pillow, 2011; Sadeghi et al., 2013):

$$\begin{aligned} L(\theta) &= \theta^T \sum_t X_t^T n_t - \sum_t \exp(x_t^T\theta)dt & (7) \\ &\approx \theta^T \sum_t X_t^T n_t - TE_{\vec{x}} \exp(x^T\theta) \equiv \tilde{L}(\theta), & (8) \end{aligned}$$

where the expectation is with respect to $p(\vec{x})$. The EL trades in the $O(KTp)$ cost of computing the nonlinear sum for the cost of computing the expectation over $\exp(x^T\theta)$ at K different values of θ, resulting in order $O(Kz)$ cost, where z denotes the cost of computing the expectation. Thus the nonlinear term of the EL can be be computed about $\frac{Tp}{z}$ times faster than the dominant term in the exact GLM log-likelihood. Similar gains are available in computing the gradient and Hessian of these terms with respect to θ.

How hard is it to compute this integral in practice? I.e., how large is z? First, note that because $\exp(x^T\theta)$ only depends on the projection of x onto θ, calculating this expectation only requires the computation of a one-dimensional integral:

$$E(\exp(x^T\theta)) \;\; = \;\; \int \exp(x^T\theta)p(\vec{x})d\vec{x} = \int \exp(q)\zeta_\theta(q)dq, \tag{9}$$

where ζ_θ is the (θ-dependent) distribution of the one-dimensional variable $q = x^T\theta$. If ζ_θ is available analytically, then we can simply apply standard unidimensional numerical integration methods to evaluate the expectation.

In certain cases this integral can be performed analytically. For example, if $p(x)$ is Gaussian with mean zero and covariance C, then

$$\int \exp(x^T\theta)\frac{1}{(2\pi)^{\frac{p}{2}}|C|^{\frac{1}{2}}} \exp\Big(-x^TC^{-1}x/2\Big)dx = \exp\Big(\frac{\theta^TC\theta}{2}\Big), \tag{10}$$

where we have recognized the moment-generating function of the multivariate Gaussian distribution.

Note that in the Gaussian case, the expectation turns out to depend only on $\theta^TC\theta$. This will always be the case if $p(x)$ is elliptically symmetric, by logic similar to that described in the last section; see (Ramirez and Paninski, 2013) for details. Thus we only need to compute this integral once for all values of $||\theta'||_2^2 = \theta^TC\theta$, up to some desired accuracy. This can be precomputed off-line and stored in a one-dimensional lookup table before any EL computations are required, making the amortized cost z very small.

What if $p(x)$ is non-elliptical and we cannot compute ζ_θ easily? We can still compute the integral approximately in most cases with an appeal to the central limit theorem (Sadeghi et al., 2013): we approximate $q = x^T\theta$ as Gaussian, with mean $E(\theta^Tx) = \theta^TE(x) = 0$ and variance $\text{var}(\theta^Tx) = \theta^TC\theta$. This approximation can be justified by the classic results of (Diaconis and Freedman, 1984), which imply that under certain conditions, if d is sufficiently large, then ζ_θ is approximately Gaussian for most projections θ. (Of course in practice this approximation is most accurate when the vector x consists of many weakly-dependent, light-tailed random variables and θ has large support, so that q is a weighted sum of many weakly-dependent, light-tailed random variables.) Thus, again, we can precompute a lookup function for $E(\exp(x^T\theta))$, this time over the two-dimensional table of all desired values of the mean and variance of q. Numerically, this approximation often works quite well; see (Ramirez and Paninski, 2013) for further discussion.

Further speedups are available when we think about optimizing the expected loglikelihood (as an approximation to the MLE) or the penalized expected loglikelihood (as an approximation to the MAP estimate). Somewhat surprisingly, the maximum expected loglikelihood estimator (MELE) can be computed analytically for this model (Park and Pillow, 2011) if $p(x)$ is Gaussian and we modify the model slightly to include an offset term so that the Poisson rate in the t-th time bin is given by

$$\lambda_t = \exp(\theta_0 + x_t^T\theta), \tag{11}$$

with the likelihood and EL modified appropriately. The details are provided in (Park and Pillow, 2011); the key result is that if one first analytically optimizes the EL with respect to the offset θ_0 and then substitutes the optimal θ_0 back into the EL, the resulting "profile" expected log-likelihood $\max_{\theta_0} \tilde{L}(\theta, \theta_0)$ is a quadratic function of θ, which can be optimized easily to obtain the MELE:

$$\arg\max_{\theta} \left(\max_{\theta_0} \tilde{L}(\theta, \theta_0) \right) = \arg\max_{\theta} \left(\theta^T X^T n - \sum_{t=1}^{T} n_t \frac{\theta^T C \theta}{2} \right) \quad (12)$$

$$= \left((\sum_t n_t) C \right)^{-1} X^T n. \quad (13)$$

Note that this is essentially the same quadratic problem that arises in the context of least-squares estimation, with $(\sum_t n_t)C$ replacing $X^T X$. In many cases C is a highly structured matrix (e.g., diagonal in a computationally-tractable basis), and can therefore be inverted easily. Thus in many cases it is even easier to optimize the EL than it is to compute the least-squares estimator; see (Park and Pillow, 2011; Sadeghi et al., 2013; Ramirez and Paninski, 2013) for further discussion. (Ramirez and Paninski, 2013) further notes that the statistical accuracy of the MLE and maximum EL estimators are often comparable; in cases where this is not true, the EL can be used to speed up the computation of the MLE, by providing a good initialization and preconditioner for the likelihood optimization.

5 Incorporating nonlinear terms; connections to the spike-triggered covariance

We have seen above that standard GLM estimators are effective if the stimulus distribution $p(\vec{x})$ is elliptically symmetric, and if the spike-triggered average (or weighted STA in the case that $f(.) \neq \exp(.)$) is nonzero. However, it is easy to think of cases where the STA will converge to zero. This occurs, for example, whenever the true nonlinearity $f(.)$ is symmetric with respect to its argument, i.e., when the neuron is sensitive only to the magnitude of the stimulus, not the sign, as is the case for complex cells in the primary visual cortex (Simoncelli and Adelson, 1996), for example. In this case, STA-based techniques fail to recover any meaningful information at all (since the STA converges to zero).

We might also consider the following simple generalization of the LN model: $E(n|\vec{x}) = f(K^T \vec{x})$, where K is an m-by-d matrix and $f(.)$ is now an m-dimensional nonlinearity. Clearly in this case the STA fails to capture all of the information in K; even in the radially symmetric case, it is easy to see (by a direct generalization of our symmetry argument above) that the expectation of the STA estimate now falls within the subspace spanned by the columns of K. But is there a way to capture all the columns of K, instead of just a single linear combination?

Clearly we can incorporate nonlinear covariate terms in the Poisson regression model, just as in the standard regression model. This is one powerful approach to make the GLM methodology much more flexible, though as discussed in the previous chapter, choosing the nonlinear covariate terms appropriately is often a challenging task. One example where a good deal of the theory has been worked out is the second-order Volterra series case, where the nonlinear covariates are chosen to be all possible quadratic interactions of the vector $\vec{x}$, so that the log-firing rate is taken to be

$$\log \lambda_t = X_t A X_t^T + X_t \theta + b$$

for a suitable matrix A, vector θ, and scalar offset b. (Recall we discussed this class of quadratic models, along with the associated rank-penalizing priors, in the linear regression context the previous chapter.) There are close links between this quadratic GLM and the method of spike-triggered covariance (STC) (Brenner et al., 2001; Simoncelli et al., 2004; Rust et al., 2005), which we'll describe in a bit more detail now.

The basic idea behind STC approaches is that, if the first moment of the conditional distribution $p(\vec{x}|n)$ (the STA) is not different from that of the prior distribution $p(\vec{x})$, then perhaps an analysis of the second moment matrix will be more revealing. In the simplest case, assume that $\vec{x}$ has a multivariate Gaussian distribution and (after a standardizing transformation of $\vec{x} \rightarrow Cov(\vec{x})^{-1/2}[\vec{x} - E(\vec{x})]$, if necessary) that $Cov(\vec{x}) = I$. Then it is easy to see that the posterior second moment matrix $C_{post} \equiv E(\vec{x}\vec{x}^T|n = 1)$ can in general only differ from I in directions spanned by the columns of K. To see this, break $\vec{x}$ into two orthogonal components $\vec{x} = \vec{x}_K + \vec{x}_{\backslash K}$, where $\vec{x}_K$ lies within the subspace spanned by K, and $\vec{x}_{\backslash K}$ in the orthogonal subspace, and note that

$$\begin{aligned}
p(\vec{x}_{\backslash K}|\vec{x}_K, n) &= \frac{p(\vec{x}_{\backslash K}, \vec{x}_K, n)}{p(\vec{x}_K, n)} \\
&= \frac{p(\vec{x}_{\backslash K}, \vec{x}_K)p(n|\vec{x})}{p(\vec{x}_K, n)} \\
&= \frac{p(\vec{x}_{\backslash K}, \vec{x}_K)p(n|\vec{x}_K)}{p(\vec{x}_K, n)} \\
&= \frac{p(\vec{x}_{\backslash K})p(\vec{x}_K)p(n|\vec{x}_K)}{p(\vec{x}_K, n)} \\
&= \frac{p(\vec{x}_{\backslash K})p(\vec{x}_K, n)}{p(\vec{x}_K, n)} \\
&= p(\vec{x}_{\backslash K});
\end{aligned}$$

i.e., the conditional distribution in the orthogonal subspace is equal to the prior distribution, and therefore any conditional moments in this subspace must be equal to the prior moments. The key equality here — namely, $p(\vec{x}_{\backslash K}, \vec{x}_K) = p(\vec{x}_{\backslash K})p(\vec{x}_K)$ — is due to the Gaussian assumption on $p(\vec{x})$.

This suggests a straightforward principal components-based estimator for K: we compute the eigenvectors of the matrix formed by the difference $\hat{C}_{post} - I$, where $\hat{C}_{post}$ is a consistent estimator of C_{post} (typically the sample second moment matrix). Our estimator for the subspace spanned by K is now

$$\hat{K} = eig(\hat{C}_{post} - I),$$

where the operator $eig(A)$ extracts the eigenvectors of the matrix A which are significantly different from zero[3]. Constructing a bona fide significance test in this setting is a slightly more difficult proposition; see, e.g., (Rust et al., 2005) for a bootstrap-based analysis. In general we prefer to consider this STC analysis an exploratory method, useful for identifying subspaces K in which the neuron is tuned, and therefore more qualitative definitions of significance — i.e., choose all eigenvectors corresponding to eigenvalues which appear qualitatively different in magnitude from the "bulk spectrum" (Johnstone, 2000), i.e., the remaining eigenvalues — are sufficient for our purposes. It is straightforward to prove that this simple approach

[3] Note that we can only hope to identify the column space of K, not K itself, since any nonsingular linear transformation of K may be absorbed in the definition of the nonlinear function $f(.)$, just as in the one-dimensional θ case. It is also worth remembering that the eigenvectors of any symmetric matrix (e.g., $C_{post} - I$) must be orthogonal; thus the estimate $\hat{K}$ always has orthogonal columns.

provides a consistent estimator for K in this case of a standard Gaussian $p(\vec{x})$ (Paninski, 2003); a simple extension of the approach provides a consistent estimator in the more general case of elliptically symmetric stimuli (Samengo and Gollisch, 2013).

It is worth noting that estimating the matrix C_{post} requires a fair amount of data. It is known that the usual empirical estimate of C_{post} (the sample second-order matrix) is unbiased, but if $d = \dim(\vec{x})$ is on the same order as N, the number of spikes, then the corresponding empirical eigenvalue spectrum is in fact strongly biased (Fig. 3). Dealing with this bias is an active research area in random matrix theory (Johnstone, 2000; Ledoit and Wolf, 2004; Schafer and Strimmer, 2005; El Karoui, 2007), though to date few of these more recent methods have been applied in the context of neural data.

(Park and Pillow, 2011) provide a detailed discussion of the links between this STC analysis and the GLM with quadratic interaction terms. The basic idea is to begin with the loglikelihood, as usual:

$$L(A, \theta, b) = \sum_t n_t \left(X_t A X_t^T + X_t \theta + b \right) - \sum_t \exp\left(X_t A X_t^T + X_t \theta + b \right) dt,$$

and then to note that the sum on the left may be written in terms of the STA and STC; just as the STA is a sufficient statistic for the spiking data in the standard GLM, here we see that the STA, STC, and total spike count form a sufficient statistic for the spiking data in this quadratic GLM. (Park and Pillow, 2011) further show that the maximum expected loglikelihood estimator for this model can be interpreted in terms of the STC and STA. More importantly, once the STC components are incorporated in a proper likelihood function we can bring Bayesian methods to bear and use matrix penalizers to improve the estimation of the model parameters here. See (Park and Pillow, 2011) for further discussion.

6 Fully semiparametric estimators give correct estimates more generally than do the STA or STC estimators

We have seen that STA- and STC-based methods can work well under certain assumptions, e.g., elliptical symmetry of the stimulus distribution $p(\vec{x})$. It is natural to seek an estimator for K which is guaranteed to be consistent more generally. Several such estimators have been constructed (Weisberg and Welsh, 1994; Paninski, 2003; Sharpee et al., 2004); however, the gains in generality are offset by the fact that the resulting estimators are much less computationally tractable than the STA or STC techniques, which require only simple linear algebraic operations and convex optimizations.

Several approaches have been proposed. The first (Weisberg and Welsh, 1994) is to use maximum likelihood to simultaneously fit both K and the function $f(.)$, where $f(.)$ is represented in a finite dimensional basis which is allowed to become richer with increasing sample size N. Note that if K is known, then we can use low-dimensional parametric or nonparametric methods to estimate the firing rate as a function of the low-dimensional projection $K^T\vec{x}$. Standard results for the consistency of maximum likelihood estimators may be adapted to establish the consistency of the resulting sequence of estimators for K; however, computationally solving this likelihood optimization problem is difficult (and becomes more difficult as the data length T increases, as the dimensionality of the space in which we must search for an optimal $f(.)$ increases).

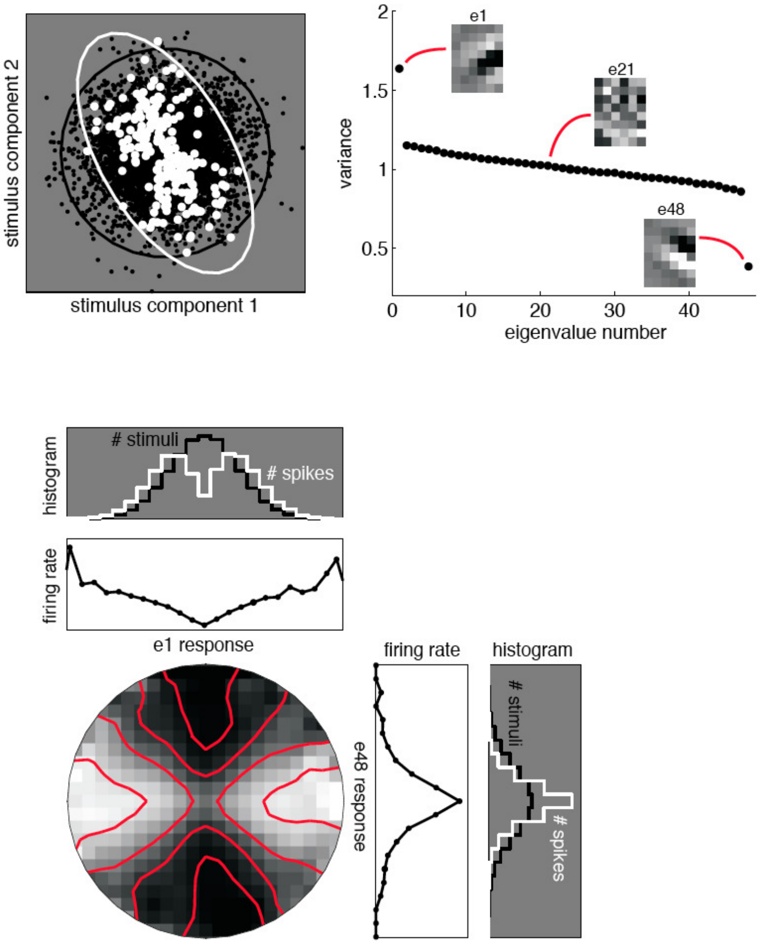

Figure 2: Simulated characterization of a two-dimensional linear-nonlinear cascade model via spike-triggered covariance (STC). In this model, the spike generator is driven by the squared response of one linear filter divided by the sum of squares of its own response and the response of another filter. Both filters are 6×8, and thus live in a 48-dimensional space. The simulation is based on a sequence of $200,000$ raw stimuli, with $8,000$ spikes. **Top, left**: simulated raw and spike-triggered stimulus ensembles, viewed in the two-dimensional subspace spanned by the filters θ_1 and θ_2. The covariance of these ensembles within this two-dimensional space is represented geometrically by an ellipse that is three standard deviations from the origin in all directions. The raw stimulus ensemble has equal variance in all directions, as indicated by the black circle. The spike-triggered ensemble is elongated in one direction and compressed in another direction (white ellipse). **Top, right**: Eigenvalue analysis of the simulated data. The principal axes of the covariance ellipse correspond to the eigenvectors of the spike-triggered covariance matrix, and the associated eigenvalues indicate the variance of the spike-triggered stimulus ensemble along each of these axes. The plot shows the full set of 48 eigenvalues, sorted in descending order. Two of these are substantially different the others (one significantly larger and one significantly smaller), indicating the presence of two axes in the stimulus space along which the model is differentially responsive. Also shown are three example corresponding eigenvectors. **Bottom, one-dimensional plots**: Spike-triggered and raw histograms of responses along the two distinguished eigenvectors, along with the nonlinear firing rate functions estimated from their quotient. **Bottom, two-dimensional plot**: the quotient of the two-dimensional spike-triggered and raw histograms provides an estimate of the two-dimensional nonlinear ring firate function $f(.)$. This is shown as a circular-cropped grayscale image, where intensity is proportional to firing rate. Superimposed contours (red) indicate four different response levels. Adapted from (Simoncelli et al., 2004).

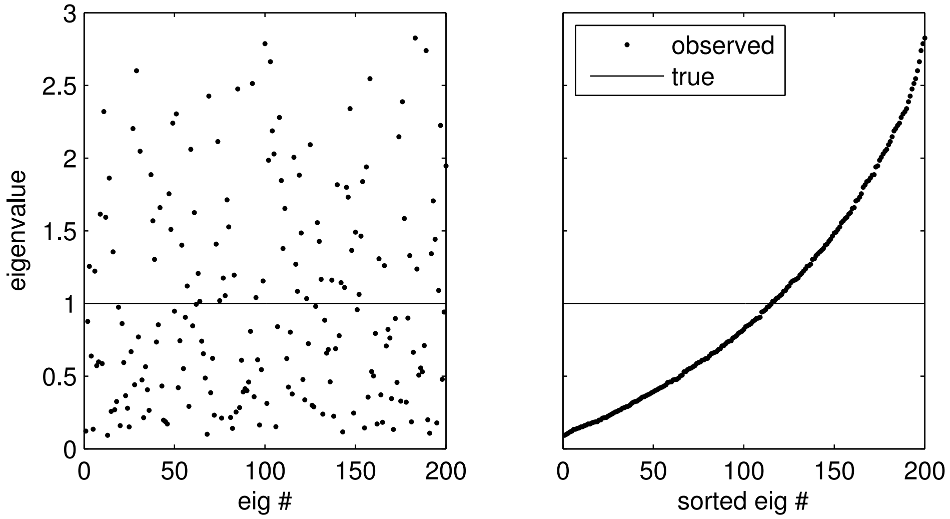

Figure 3: Illustration of the bias in a sparsely-sampled covariance spectrum (Johnstone, 2000). 400 samples were drawn from a 200-dimensional standard normal distribution (mean zero, identity covariance), and then the eigenvalues of the empirical covariance matrix were computed. These eigenvalues display a good deal of variability around their true value (identically 1 in this case), and this variability is converted into systematic bias when the eigenvalues are sorted; unsorted eigenvalues shown on the **left**, and sorted eigenvalues shown on the **right**.

A second approach has been employed more recently in the machine learning and statistics literature, where models with LN form are often referred to as "multi-index models" (or "single-index models" in the case that K contains just one row, i.e., $K\vec{x}$ is one-dimensional). The idea is to maximize a semi-parametric likelihood, as before, but include a penalty on the rank of the Hessian of an estimate of the firing rate as a function of $\vec{x}$; this approach exploits the fact that if the firing rate is well-described by a function of LN form $f(K\vec{x})$, then the Hessian of this function will have rank equal to the rank of K, rather than $\dim(\vec{x})$. This method can exploit recent sophisticated methods for optimization of rank-penalizing functions, but constructing the Hessian of the estimated firing rate function can be computationally challenging. See, e.g., (Hemant and Cevher, 2012) for further details.

Another approach is to construct an objective function, $M(V)$ (here V denotes a matrix of the same size as K), with the property that $M(V)$ obtains its optimum if and only if $V = K$. Then we construct our estimator by maximizing some empirical estimator $\hat{M}(V)$. The idea is that if $\hat{M}(V)$ is a good estimator for $M(V)$, then given enough data the maximizer of $\hat{M}(V)$ should be close to the maximizer of $M(V)$, i.e., K. See, e.g., (van der Vaart, 1998) for more details on this argument.

The objective function chosen is based on the so-called "data processing inequality" from information theory (Cover and Thomas, 1991). The basic idea is that $V^T\vec{x}$ is equivalent to $K^T\vec{x}$ plus some "noise" term that does not affect the spike process (more precisely, this noise term is conditionally independent of n given $K^T\vec{x}$); this noise term is obviously 0 for $V = K$. Thus if we optimize the objective function

$$M(V) \equiv I(V^T\vec{x}; n),$$

with $I(X;Y)$ denoting the mutual information between the random variables X and Y,

$$I(X;Y) = \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy,$$

the data processing inequality (Cover and Thomas, 1991) says that $M(V)$ obtains its maximum at the point $V = K$. Another way to put it is that $M(V)$ measures how strongly $V^T\vec{x}$ modulates the firing rate of the cell: for V near K, the conditional measures $p(n = i|V^T\vec{x})$ are on average very different from the prior measure $p(n = i)$, and $M(V)$ is designed to detect exactly these differences; conversely, for V orthogonal to K, the conditional distributions $p(n = i|V^T\vec{x})$ will appear relatively "unmodulated" (that is, $p(n = i|V^T\vec{x})$ will tend to be much nearer the average $p(n = i)$), and $M(V)$ will be comparatively small.

Under weak conditions (e.g., that the support of $p(\vec{x})$ covers every possible point $\vec{x}$ — i.e., all points are sampled with some probability, roughly speaking), this maximum of $M(V)$ at $V = K$ is unique (up to equivalence of column spaces), by the converse to the data processing inequality (Cover and Thomas, 1991). Thus if we choose some uniformly consistent estimator $\hat{M}(V)$ of $M(V)$ (Beirlant et al., 1997), it is easy to show that the estimator

$$\hat{K} = \arg\max_V \hat{M}(V)$$

is consistent for K under weak conditions (Paninski, 2003; Sharpee et al., 2004). However, as emphasized above, maximizing $\hat{M}(V)$ is often fairly computationally difficult, since $\hat{M}(V)$ is typically non-convex and many local maxima may occur in general. In fact, it turns out that an equivalence may be drawn between the semiparametric maximum likelihood approach and the information-theoretic objection function approach; see (Williamson et al., 2013) for a nice discussion of these issues.

7 We may also easily incorporate 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 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 requirement of any good encoding model: the LNP model 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 observables into this matrix as well. For example, as usual, appending a column of ones to X corresponds to incorporating a constant "offset" parameter b in our model, $\lambda(t) = f(\theta \cdot \vec{x}(t) + b)$, which provides an important degree of flexibility in setting the threshold and baseline firing rate of the model.

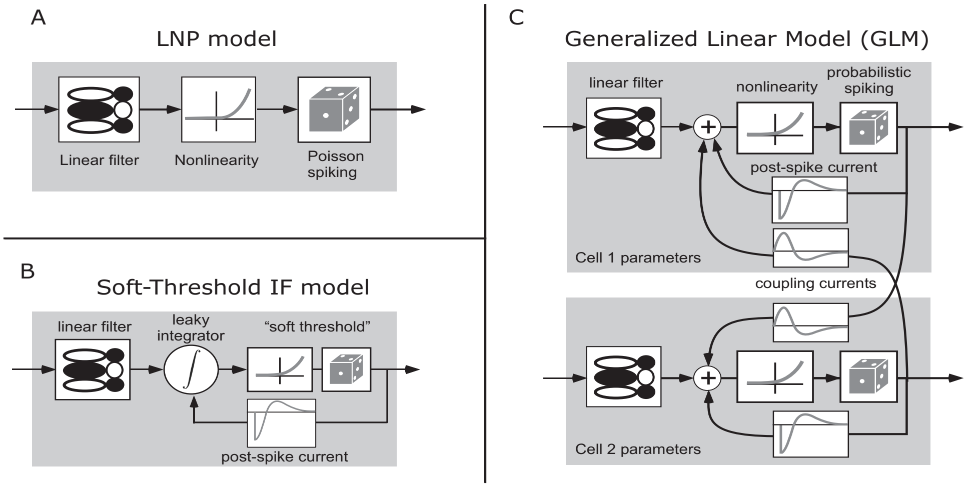

Figure 4: Schematic diagrams of some of the encoding models discussed here. **A**: the linear-nonlinear-Poisson (LNP) model is strictly feedforward, with no spike-history terms. **B**: Illustration of the connection between the GLM with spike-history terms and the integrate-and-fire cell with a probabilistic ("soft") threshold. **C**: GLM incorporating both spike-history and interneuronal coupling terms $h(.)$.

More importantly, we may incorporate terms corresponding to the neuron's observed past activity n_{t-j}, to obtain models of the form $\lambda(t) = f(b + \theta \cdot \vec{x}(t) + \sum_{j=1}^{J} h_j n_{t-j})$ (Fig. 4c). Depending on the shape of the "spike history filter" $\vec{h}$, the model can display all of the effects described above (Paninski et al., 2004c); 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 $X_t = \begin{pmatrix} 1 & \vec{x}(t) & n_{t-J} & n_{t-J+1} & \dots & n_{t-1} \end{pmatrix}$), 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 solution for the model parameters $\theta = \{b, \theta, \vec{h}\}$ 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. 5 for an application to data from a retinal ganglion cell (Uzzell and Chichilnisky, 2004; Pillow et al., 2005b), and Fig. 6 for an illustration in model data of how these spike-history effects induce interesting dynamic effects in the response of the neuron to a simple step current.

In the beginning of this chapter, we focused on the problem of "PSTH smoothing," in which we wanted to estimate the firing rate of a single neuron observed over multiple repeated trials, given fixed stimulus conditions. In this context we emphasized the importance of finding computational methods that scaled linearly with the trial length T, in order to keep the computations tractable. In particular, by enforcing bandedness of the loglikelihood Hessian we obtained methods with computational complexity that scaled linearly with T. These

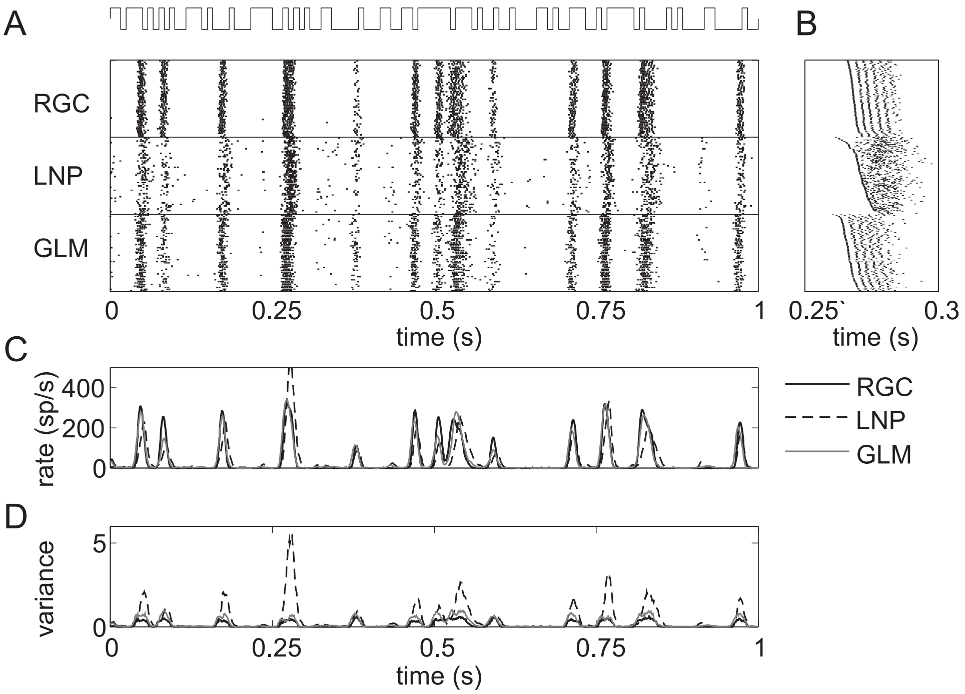

Figure 5: Example predictions of a single retinal ganglion ON-cell's activity using the GL encoding model with and without spike history terms. Conventions are as in Figs. 3 and 4 in (Pillow et al., 2005b); physiological recording details as in (Uzzell and Chichilnisky, 2004; Pillow et al., 2005b). **A**: 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 neuron's spike train 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.

methods can still be used if we incorporate spike history effects. The key trick is to split the parameter vector θ into the terms θ_1 we had before (e.g., the spline coefficients, if we are modeling the log-firing rate in a spline basis) and new terms θ_2 involving the spike-history effects. The new Hessian, involving the full vector θ, is not banded in general, since typically many of the cross-terms involving both θ_1 and θ_2 will be nonzero. However, the block of the Hessian involving just θ_1 terms will still be banded, and Schur decomposition methods can be used in this case to recover our previous $O(T)$ scaling.

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 (Chornoboy et al., 1988; Paninski et al., 2004a; Paninski et al., 2004a; Truccolo et al., 2005; Pillow et al., 2005a). The resulting model

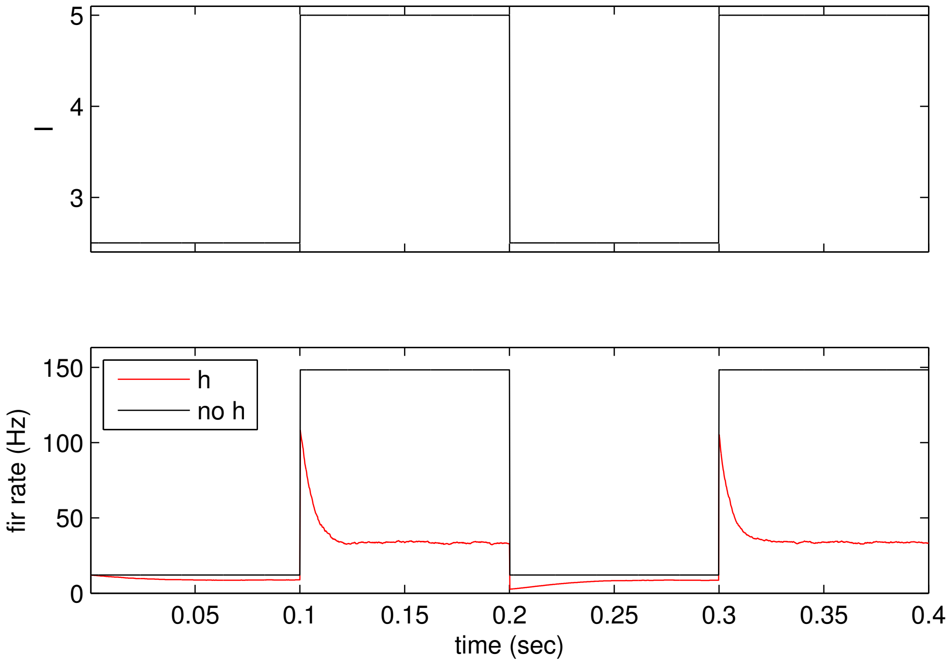

Figure 6: Mean response of a simulated GLM neuron to step inputs. Red trace shows the empirical average firing rate, $E[\lambda(t)]$ (where the expectation is taken over the random spike history effects $\sum_j h_j n_{t-j}$) given the step input $I(t)$. The spike-history function here was chosen to be $h(t) = -3\exp(-t/\tau), t > 0$, with $\tau = 20$ ms; the firing rate was given by $\lambda(t) = \exp(I(t) + \sum_j h_j n_{t-j})$. Response of the corresponding inhomogeneous Poisson neuron (with h set to zero) shown for comparison. Note that the inhibitory spike-history term here increases the transience of the response and decreases the overall spike count; we will discuss analytical methods for computing this time-varying mean firing rate $E[\lambda(t)]$ in a later chapter.

is summarized (Fig. 4c):

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

with $\lambda_i(t)$ denoting the instantaneous firing rate of the i-th cell at time t, θ_i the cell's linear receptive field, and $\vec{h}_{i',i}$ 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 $n_{i',t-j}$ in the network. The $\vec{h}_{i,i}$ terms (corresponding to the i-th cell's own past activity) plays the role of $\vec{h}$ above; the $\vec{h}_{i',i}$ terms from the other cells in the population correspond to interneuronal interaction effects. Once again, the loglikelihood remains jointly concave in all the parameters ($\{b_i, \vec{k}_i, \vec{h}_{i',i}\}$). In addition, the loglikelihood has a nicely separable form: we can break up this objective function into a sum of N terms, each of which involve only the parameters governing the firing of the i-th neuron, and which can therefore be solved in parallel. This means that the full loglikelihood — which appears forbidding, since it involves a highly nonlinear, recurrently connected network — turns out to be highly tractable.

What basis should we choose to represent these spike-history effects? The simplest basis is the “histogram” basis composed of nonoverlapping step functions (functions which are zero everywhere except on a convex set, where the function is constant). In some cases it makes sense for some of the basis functions to be of different widths: for example, to represent the spike history function $h(.)$, it makes sense to put more narrow basis functions at time delays near zero (where we expect $h(.)$ to vary quickly) and fewer, wider basis functions at large time delays (where $h(.)$ should vary more slowly). However, note that it is important not to “oversample” here: if we use basis functions that are narrower than the smallest observed interspike interval, then the MLE for the corresponding basis coefficient will be pushed towards negative infinity, in order to enforce the observed zero firing rate on short intervals following a spike. This results in unstable algorithms and highly variable estimators.

The step basis is discontinuous, so the inferred kernel or spike-history function will not be smooth. A smoother basis is composed of “spline” functions. Splines are piecewise polynomial (i.e., smooth) functions; these piecewise polynomials are defined over nonoverlapping intervals which meet at a single point, the “knot”; the spline is constrained to be continuous and differentiable at these knots. The more knots are used, the higher the dimensionality of the basis (and the less smooth the resulting representation, in general). Splines may be used to represent both post-spike effects and effects that depend on the time since the beginning of the trial (PSTH effects) (Kass and Ventura, 2001). In the former case, as in the step basis setting, it is a good idea to place more knots near the spike time (to allow more flexibility in representing $h(.)$ here), since we may expect $h(.)$ to be more smoothly varying for larger t.

Another useful basis is composed of decaying exponentials (with logarithmically-spaced time constants), for example to represent the spike history function $h(.)$: again, this basis varies more quickly for small t and smoothly approaches zero for large times. This exponential basis has some mathematical advantages that we will discuss later, in the context of Markovian models of spike trains.

8 The point-process GLM is closely connected to biophysical models such as the soft-threshold integrate-and-fire model

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 (Reich et al., 1998; Gerstner and Kistler, 2002). One connection is provided by the following model: consider the inhomogeneous Poisson process with rate given by $f(V(t)+b)$, where f is a convex, log-concave scalar function, b is a constant offset term, and $V(t)$ is the solution of the “leaky integrate-and-reset” differential equation model for the dynamics of the intracellular voltage V (we'll discuss this model in much more depth in a later chapter):

$$\frac{\partial V}{\partial t} = -gV(t) + \theta \cdot \vec{x}(t) + \sum_{j=0}^{i-1} h(t - t_j),$$

with initial value

$$V(t_{i-1}) = V_{reset},$$

namely

$$V(t) = V_{reset}e^{-g(t-t_{i-1})} + \int_{t_{i-1}}^{t} \left(\theta \cdot \vec{x}(s) + \sum_{j=0}^{i-1} h(s-t_j) \right) e^{-g(t-s)} ds,$$

the linear convolution of the input current with a negative exponential of time constant $1/g$. Here g denotes the membrane leak conductance, $\theta \cdot \vec{x}(t)$ the projection of the input signal $\vec{x}(t)$ onto the spatiotemporal linear kernel θ, and h is a post-spike current waveform which is summed over the previously observed spikes. As usual, in the absence of input, V decays back to 0 with time constant $1/g$. We allow h, in turn, to take values in some low-dimensional vector space; this allows the shape and magnitude of h to be varied to fit different intrinsic spiking patterns (including burstiness, adaptation, saturation, etc.) (Gerstner and Kistler, 2002; Paninski et al., 2004c). $V(t)$ in the above is the subthreshold, and therefore unobserved ("hidden"), solution of the usual leaky integrate and fire (LIF) equation (Dayan and Abbott, 2001), for which the voltage resets to V_{reset} after the spike is emitted at t_i.

We have written the above equations to emphasize the similarity to the form of the "spike-response model" introduced by Gerstner and colleagues (Gerstner and Kistler, 2002; Jolivet et al., 2003) and employed in (Paninski et al., 2004c; Paninski et al., 2004b) to model extracellularly-recorded spike train responses. The combined IF-GL model described above is conceptually identical to a simple version of the "escape-rate" approximation to the noisy LIF-type model given in (Plesser and Gerstner, 2000; Gerstner and Kistler, 2002) (see also (Stevens and Zador, 1996)); this escape-rate approximation, in turn, was introduced to partially alleviate the difficulties associated with computing the passage time density and firing rate of the LIF model driven by noise (again, see (Paninski et al., 2004c) for more details).

Thus this "IF-GLM" can be seen as a direct approximation to the noisy LIF model developed in (Paninski et al., 2004c) (which in turn is a tractable approximation to more detailed biophysical models for spiking dynamics). Indeed, since this differential equation is linear, $V(t)$ here may be written in the form $\theta_g \cdot \vec{x}(t) + \vec{h}_g \cdot \vec{n}(t)$, where θ_g and $\vec{h}_g$ correspond to the original parameters θ and $\vec{h}$ temporally convolved with the exponential function e^{-gt}; that is, this soft-threshold IF model is just a version of the generalized linear spike train model we have been considering above, and therefore the GLM parameters may be indirectly interpreted in biophysical terms.

The main result of interest here is that the loglikelihood for the IF-GLM is jointly concave in the parameters $\{\theta, h, V_{reset}, b\}$, for any data $\{\vec{x}, t_i\}$, ensuring the tractability of the MLE. (The loglikelihood is not necessarily concave in g here, but one-dimensional maximizations are eminently tractable; it is worth noting that the convolution kernel e^{-gt} can be generalized as well, to possibly nonstationary kernels (Stevens and Zador, 1998; Jolivet et al., 2003), at the expense of the possible addition of a few more parameters.)

9 The observed Fisher information matrix is useful for large-sample approximations of confidence ellipses and errorbars

Once we have obtained an estimate for the parameters θ in the GLM, how can we quantify our uncertainty about this estimate?

As we have discussed previously, we can measure the scale of the posterior distribution along an arbitrary axis in a fairly simple manner: since we know (by the concavity of the log-posterior, under the appropriate assumptions) that the posterior is characterized by a single "bump," and the position of the peak of this bump is already characterized by θ_{MAP}, it is enough to measure the curvature of this bump at the peak θ_{MAP}. One way to measure this curvature is to compute the negative Hessian matrix J of second-derivatives of the log-posterior, $J_{ij} = -\partial^2 \log p(\theta|X,D)/\partial\theta_i\partial\theta_j$. We have already seen one example of this: in the linear case, where the posterior was of the form

$$\log p(\theta|X,D) = -\frac{1}{2}\theta^T A\theta + b^T\theta + const.$$

for some matrix A and vector b, we have that the negative Hessian is simply $J = A$. Moreover, just as in the linear case, the eigendecomposition of this matrix J tells us exactly which axes of parameter space we are most uncertain about: small eigenvalues of J correspond to directions of small curvature, where the observed data D poorly constrains the posterior distribution $p(\theta|X,D)$ (and therefore the posterior variance will be relatively large in this direction), while conversely large eigenvalues in J imply relatively precise knowledge of θ, i.e., small posterior variance. Note that the Hessian is the sum of two terms, one from the log-prior and one from the loglikelihood; recall that the negative Hessian of the loglikelihood evaluated at the MLE is the "observed Fisher information" matrix. When the likelihood term is strong compared to the prior (e.g., if T is very large), this Fisher information will be the dominant component of J.

We can furthermore use this Hessian to construct a useful approximation to the posterior $p(\theta|X,D)$. 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 $p(\theta|X,D)$ (de Ruyter van Steveninck and Bialek, 1988; Kass and Raftery, 1995; Brown et al., 1998). Specifically,

$$p(\theta|X,D) \approx \left(\frac{|J|}{2\pi}\right)^{\dim(\vec{\theta})/2} \exp\left(-\frac{1}{2}(\vec{\theta} - \theta_{MAP})^T J(\theta - \theta_{MAP})\right), \qquad (14)$$

where J here plays the role of the inverse covariance matrix: $cov(\theta) \approx J^{-1}$. This Gaussian approximation is typically referred to as the "Laplace approximation" (confusingly!) in the statistics and machine learning literature.

Now errorbars around a single parameter may be formed by computing the marginal standard deviation under this Gaussian model[4]:

$$\hat{\sigma}_i = \left[\left(cov(\vec{\theta}|X,D)\right)_{ii}\right]^{1/2} \approx \left[\left(J^{-1}\right)_{ii}\right]^{1/2}.$$

[4] A common mistake is to take J_{ii}^{-1} as the approximate variance here, instead of the correct value $\left(J^{-1}\right)_{ii}$. In fact, it is possible to show that $J_{ii}^{-1} \leq \left(J^{-1}\right)_{ii}$ (i.e., the mistaken value is biased systematically downwards from the correct value); this follows from an application of Schur complements to the symmetric positive semidefinite matrix J. In particular (assuming $i = 1$, without loss of generality), we may write

$$J = \begin{pmatrix} J_{11} & B \\ B^T & C \end{pmatrix}$$

for some matrix B and symmetric positive semidefinite matrix C. Now by using the Schur complement (Strang, 1988) we have

$$\left(J^{-1}\right)_{11} = \left(J_{11} - BC^{-1}B^T\right)^{-1};$$

since $BC^{-1}B^T \geq 0$, clearly $J_{ii}^{-1} \leq \left(J^{-1}\right)_{ii}$.

It worth noting that it is often not necessary to compute the full inverse of J; specialized algorithms are available if only a few diagonal (or near-diagonal) elements of J^{-1} inverse are needed, for example.

A useful application of this Gaussian approximation is as follows. Suppose that the matrix X is very large — large enough that it would be useful to split it up into several (say, M) chunks, which may be processed in parallel on M independent machines, or perhaps only because matrices of $size(X)/M$ fit more comfortably into our computer's memory than do matrices of size X. It would be nice to solve the MAP problem

$$\max_{\theta} \log p(D|X,\theta) + \log p(\theta)$$

using only these chunks of X. This Gaussian approximation gives us a good way to combine these chunks in a principled manner:

$$\begin{aligned}
\log p(\theta|\{X_1, X_2, \ldots X_M\}, \{D_1, D_2, \ldots D_M\}) &= c + \log p(\{D_1, D_2, \ldots D_M\}|\{X_1, X_2, \ldots X_M\}, \theta) + \log p(\theta) \\
&= c + \left(\sum_{i=1}^{M} \log p(D_i|X_i, \theta) \right) + \log p(\theta) \\
&= c + \sum_{i=1}^{M} \left(\log p(D_i|X_i, \theta) + \frac{1}{M} \log p(\theta) \right) \\
&\approx c + \sum_{i=1}^{M} \left(-\frac{1}{2} (\vec{\theta} - \theta_{MAP,i})^T J_i (\theta - \theta_{MAP_i}) \right) \\
&= c - \frac{1}{2} (\vec{\theta} - \theta^*_{MAP})^T J^* (\theta - \theta^*_{MAP}),
\end{aligned}$$

where we have used the conditional independence of the spiking data D_i given the stimulus X and the parameter θ in the second line, our Gaussian approximation in the fourth line, and made the abbreviations

$$J \approx J^* = \sum_{i=1}^{M} J_i$$

and

$$\theta_{MAP} \approx \theta^*_{MAP} = (J^*)^{-1} \left(\sum_{i=1}^{M} J_i \theta_{MAP,i} \right).$$

Note in the special case that all the J_i matrices are the same (this will be approximately true asymptotically under certain conditions), we obtain the simple and intuitive result that

$$\theta^*_{MAP} = \frac{1}{M} \sum_{i=1}^{M} \theta_{MAP,i},$$

i.e., we simply take the average of the chunked MAP estimates $\theta_{MAP,i}$.

How do we compute errorbars on other quantities of interest, such as $(X\theta)_t$? This is straightforward — just use the formula for linear transformations of covariances, $Cov(X\theta) = XCov(\theta)X^T$. If we have used the Laplace approximation to estimate $Cov(\theta) \approx J^{-1}$, we then obtain a posterior confidence interval of size proportional to $\sqrt{[XJ^{-1}X']_{tt}}$.

What about errorbars on the estimated firing rate $\lambda = f(X_t\theta)$? We could compute the first two moments $E(\lambda) = \int p(X_t\theta)f(X_t\theta)$ and $E(\lambda^2) = \int p(X_t\theta)f(X_t\theta)^2$, using the Gaussian approximation for $p(X_t\theta)$; this would provide an estimate of the mean and variance of λ. However, if $f(.) = \exp(.)$ and $X_t\theta$ is Gaussian, then $f(X_t\theta)$ will have fairly heavy tails, and it is more appropriate to compute approximate quantiles for λ: since $f(.)$ is assumed to be a monotonically increasing function, the a-th quantile for $f(X_t\theta)$ is just $f(.)$ evaluated at the a-th quantile for $X_t\theta$.

10 Bootstrap techniques provide another method for constructing confidence intervals*

confidence intervals can also be computed via MCMC methods; we will discuss these later.

11 The observed Fisher information matrix may be used to help select stimuli adaptively

In many experimental settings we have a good deal of control over what stimulus $\vec{x}$ is presented at time t. "Optimal experimental design" is a branch of statistics ("active learning" is the relevant branch of machine learning) that studies the question of how to choose stimuli $\vec{x}$ optimally, in some sense (Fedorov, 1972; Mackay, 1992; Chaloner and Verdinelli, 1995; Mascaro and Bradley, 2002; Paninski, 2005). Classical experimental design, for example, typically focuses how to construct the design matrix X optimally in a standard linear regression setting. Our objective is to select, in an online, closed-loop manner, the stimuli that will most efficiently characterize the neuron's response properties (Fig. 7a).

An important property of GL models is that not all stimuli will provide the same amount of information about the unknown coefficients θ. 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 (Mackay, 1992; Nelken et al., 1994; Machens, 2002). 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 (Cover and Thomas, 1991) of the posterior distribution on the model parameters $p(\theta|\vec{x}(t), D)$ to quantify this uncertainty, we arrive at the objective function

$$I(\theta, D|\vec{x}) = \int p(\theta, D|\vec{x}) \log \frac{p(\theta, D|\vec{x})}{p(\theta|\vec{x})p(D|\vec{x})} d\theta dD,$$

the mutual (Shannon) information between the response n_t and the model parameters θ given the stimulus and past data (Mackay, 1992; Paninski, 2005).

Unfortunately, it is quite difficult to perform this optimization in real time, for two reasons: 1) computing the information $I(\theta, D|\vec{x})$ for any given value of $\vec{x}$ requires an integration over θ and D, each of which may be very high dimensional; 2) $I(\theta, D|\vec{x})$ may in general have many local optima as a function of the high-dimensional variable $\vec{x}$.

Here the special structure of the GLM comes into play. We saw in the last section how a Gaussian approximation of the posterior distribution $p(\theta|X, D)$ can greatly simplify various computations in this model. (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 $\hat{\theta}_{MAP}$ will converge to the true underlying parameter θ — given a sufficiently long observation time T (Paninski, 2005).)

As discussed above, the observed Fisher information describes the uncertainty in our estimate of θ: for example, the determinant of this matrix measures the volume of the confidence ellipsoid under the Gaussian approximation. Our goal in collecting data is to make this ellipsoid as small as possible, and thus it is reasonable to choose $\vec{x}$ to make this determinant as small as possible on average (where the average is taken over all possible responses D); this is known as "D-optimal" design (Fedorov, 1972). (Of course, the determinant is only one such measurement of the overall size of this confidence ellipsoid; other choices are possible. For example, "A-optimal" design corresponds to minimizing the trace of this matrix.) Another connection is furnished by the Gaussian approximation: the entropy of a Gaussian distribution with inverse covariance marix J is given by the log-determinant

$$H = -\frac{1}{2} \log |J| + const.,$$

so maximizing the determinant of J is equivalent to minimizing the posterior entropy. Computing the information has therefore been reduced from an intractable integration problem to the much more tractable computation of an average log-determinant of a Hessian matrix; i.e., we will attempt to optimize the observed Fisher information instead of the Shannon information.

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

$$\log p(n_t|\theta, \vec{x}_t) = c + n_t \log f(\theta \cdot \vec{x}_t) - f(\theta \cdot \vec{x}_t)dt$$

depends on θ only through the one-dimensional projection $\theta \cdot \vec{x}_t$. This effectively one-dimensional nature of the log-likelihood implies that the Hessian J_t of the log-posterior distribution given t observations is simply a rank-one perturbation of the Hessian J_{t-1} after $t-1$ observations:

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

where the last term is a matrix of rank one. (The equalities above are simple manipulations with Bayes rule and the definition of the Hessian.) This one-dimensional structure makes possible a very efficient recursive computation of the posterior log determinant (using the Woodbury lemma for rank-one matrix updates); after making a few more simple approximations it turns out to be possible to reduce the full d-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 (Lewi et al., 2006) 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. 7b), with the running time growing only as $O[\dim(\theta)^2]$ (as opposed to the

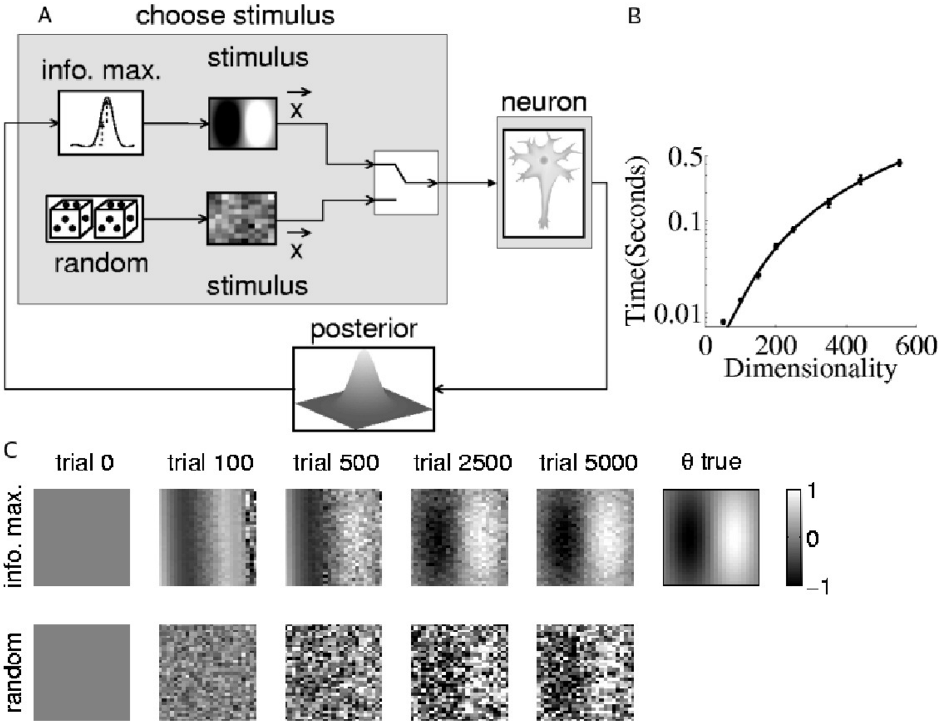

Figure 7: A) Closed-loop vs. open-loop stimulus design. B) Plot of the total running time on a desktop computer for each iteration of the model-based stimulus optimization algorithm, as a function of the dimensionality of the stimulus $\vec{x}$. A quadratic polynomial ($O[\dim(\theta)^2]$) fits the data 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 θ 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 θ_{MAP} after observing t stimulus-response pairs (the prior $p(\theta)$ 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. See (Lewi et al., 2006) for details.

exponential growth in the general, non-model-based case). Moreover, despite the approximations in the derivation, 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 (Fig. 7c); see (Lewi et al., 2006) for details.

References

Beirlant, J., Dudewicz, E., Gyorfi, L., and van der Meulen, E. (1997). Nonparametric entropy estimation: an overview. *International Journal of the Mathematical Statistics Sciences*, 6:17–39.

Brenner, N., Bialek, W., and de Ruyter van Steveninck, R. (2001). Adaptive rescaling optimizes information transmission. *Neuron*, 26:695–702.

Brown, E., Frank, L., Tang, D., Quirk, M., and Wilson, M. (1998). 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.

Bussgang, J. (1952). Crosscorrelation functions of amplitude-distorted Gaussian signals. *RLE Technical Reports*, 216.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: a review. *Statistical Science*, 10:273–304.

Chichilnisky, E. (2001). A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12:199–213.

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

Cover, T. and Thomas, J. (1991). *Elements of information theory.* Wiley, New York.

Dayan, P. and Abbott, L. (2001). *Theoretical Neuroscience.* MIT Press.

de Ruyter van Steveninck, R. and Bialek, W. (1988). Real-time performance of a movement-senstivive neuron in the blowfly visual system: coding and information transmission in short spike sequences. *Proc. R. Soc. Lond. B*, 234:379–414.

Diaconis, P. and Freedman, D. (1984). Asymptotics of graphical projection pursuit. *Annals of Statistics*, 12:793–815.

DiMatteo, I., Genovese, C., and Kass, R. (2001). Bayesian curve fitting with free-knot splines. *Biometrika*, 88:1055–1073.

El Karoui, N. (2007). Spectrum estimation for large dimensional covariance matrices using random matrix theory. *arXiv:math/0609418v1.*

Fedorov, V. (1972). *Theory of Optimal Experiments.* Academic Press, New York.

Field, G. D., Gauthier, J. L., Sher, A., Greschner, M., Machado, T. A., Jepson, L. H., Shlens, J., Gunning, D. E., Mathieson, K., Dabrowski, W., Paninski, L., Litke, A. M., and Chichilnisky, E. J. (2010). Functional connectivity in the retina at the resolution of photoreceptors. *Nature*, 467(7316):673–7.

Gerstner, W. and Kistler, W. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity.* Cambridge University Press.

Hemant, T. and Cevher, V. (2012). Active learning of multi-index function models. In Bartlett, P., Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 1475–1483.

Johnstone, I. (2000). On the distribution of the largest principal component. Technical Report 2000-27, Stanford.

Jolivet, R., Lewis, T., and Gerstner, W. (2003). The spike response model: a framework to predict neuronal spike trains. *Springer Lecture notes in computer science*, 2714:846–853.

Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90:773–795.

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

Ledoit, O. and Wolf, M. (2004). Honey, i shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30:110–119.

Lewi, J., Butera, R., and Paninski, L. (2006). Real-time adaptive information-theoretic optimization of neurophysiological experiments. *NIPS*.

Machens, C. (2002). Adaptive sampling by information maximization. *Physical Review Letters*, 88:228104–228107.

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

Mascaro, M. and Bradley, D. (2002). Optimized neuronal tuning algorithm for multichannel recording. Unpublished abstract at http://www.compscipreprints.com/.

McCullagh, P. and Nelder, J. (1989). *Generalized linear models.* Chapman and Hall, London.

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

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

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

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

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

Paninski, L., Pillow, J., and Simoncelli, E. (2004b). Comparing integrate-and-fire-like models estimated using intracellular and extracellular data. *Neurocomputing*, 65:379–385.

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

Park, I. M. and Pillow, J. W. (2011). Bayesian spike-triggered covariance analysis. *NIPS*, 24.

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

Pillow, J., Paninski, L., Uzzell, V., Simoncelli, E., and Chichilnisky, E. (2005b). Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25:11003–11013.

Plesser, H. and Gerstner, W. (2000). Noise in integrate-and-fire neurons: From stochastic input to escape rates. *Neural Computation*, 12:367–384.

Rahnama Rad, K. and Paninski, L. (2010). Efficient estimation of two-dimensional firing rate surfaces via Gaussian process methods. *Network*, 21:142–168.

Ramirez, A. and Paninski, L. (2013). Fast inference in generalized linear models via expected log-likelihoods. *J. Comput. Neurosci.*

Reich, D., Victor, J., and Knight, B. (1998). The power ratio and the interval map: Spiking models and extracellular recordings. *The Journal of Neuroscience*, 18:10090–10104.

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications.* CRC Press.

Rust, N., Schwartz, O., Movshon, A., and Simoncelli, E. (2005). Spatiotemporal elements of macaque V1 receptive fields. *Neuron*, 46:945–956.

Sadeghi, K., Gauthier, J., Greschner, M., Agne, M., Chichilnisky, E. J., and Paninski, L. (2013). Monte carlo methods for localization of cones given multielectrode retinal ganglion cell recordings. *Network*, 24:27–51.

Samengo, I. and Gollisch, T. (2013). Spike-triggered covariance: geometric proof, symmetry properties, and extension beyond gaussian stimuli. *Journal of Computational Neuroscience*, 34(1):137–161.

Schafer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4:32.

Schervish, M. (1995). *Theory of statistics.* Springer-Verlag, New York.

Schnitzer, M. and Meister, M. (2003). Multineuronal firing patterns in the signal from eye to brain. *Neuron*, 37:499–511.

Sharpee, T., Rust, N., and Bialek, W. (2004). Analyzing neural responses to natural signals: Maximally informative dimensions. *Neural Computation*, 16:223–250.

Simoncelli, E., Paninski, L., Pillow, J., and Schwartz, O. (2004). Characterization of neural responses with stochastic stimuli. In *The Cognitive Neurosciences.* MIT Press, 3rd edition.

Simoncelli, E. P. and Adelson, E. H. (1996). Noise removal via Bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, volume I, pages 379–382, Lausanne. IEEE Sig Proc Society.

Stevens, C. and Zador, A. (1996). When is an integrate-and-fire neuron like a Poisson neuron? *NIPS*, 8:103–109.

Stevens, C. and Zador, A. (1998). Novel integrate-and-fire-like model of repetitive firing in cortical neurons. *Proc. 5th joint symp. neural computation, UCSD.*

Strang, G. (1988). *Linear algebra and its applications.* Harcourt Brace, New York.

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.

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

van der Vaart, A. (1998). *Asymptotic statistics.* Cambridge University Press, Cambridge.

Weisberg, S. and Welsh, A. (1994). Adapting for the missing link. *Annals of Statistics*, 22:1674–1700.

Williamson, R. S., Sahani, M., and Pillow, J. W. (2013). Equating information-theoretic and likelihood-based methods for neural dimensionality reduction. *arXiv*, page 1308.3542.