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

May 15, 2007

Contents

*Thanks to J. Lewi and J. Pillow for many informative discussions.

0.1 Generalized linear models for spike trains; relationship to the standard linear Gaussian model

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

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

While in general there are many varieties of encoding models that could conceivably satisfy these three conditions, in this chapter we will focus on a particular model class known as the "generalized linear" model (GLM) (Paninski, 2004; Truccolo et al., 2005). This model class is a natural mathematical representation of the basic physiological concept of a "receptive field" and has proven useful in a wide variety of experimental preparations. In its simplest linear-nonlinear-Poisson (LNP) form, the model hypothesizes that spike trains are produced by an inhomogeneous Poisson process with rate

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

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

How do we estimate the model parameters $\theta = \vec{k}$ here? Clearly, this model is closely related to the linear-nonlinear-Bernoulli model we discussed in the classification-based setting, and as we will see the methods of parameter estimation in these two models share many similarities. We begin by writing down the likelihood $p(D|\vec{k}, \vec{x})$ of the observed spike data D given the model parameter $\vec{k}$ and the observed stimulus $\vec{x}$, and then we may employ standard likelihood optimization methods to obtain the maximum likelihood (ML) or maximum a posteriori (MAP) solutions for $\vec{k}$. It 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. This model hypothesizes that each bin of observed spike train data n_t of width dt is generated by the

formula $n_t = \vec{k} \cdot \vec{x}(t)dt + \epsilon_t$, where ϵ_t is discrete Gaussian white noise. If we write down the likelihood of this model using the log of the Gaussian probability density function we have

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

where c_1, c_2 are constants that do not depend on the parameter $\vec{k}$, X abbreviates the matrix of observed stimuli (the t-th row of X is given by $X_t = \vec{x}(t)$), and the sum in t is over all observed time bins. Maximizing this likelihood gives the usual least-squares regression solution $\vec{k}_{LS} \propto (X^t X)^{-1} X^T \vec{n}$.

Now, if we repeat the same exercise under the more plausible assumption that spike counts per bin follow a Poisson instead of Gaussian distribution (with the rate parameter of the Poisson distribution given by Eq. (1), we have

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

implying

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

or in the continuous-time limit $dt \to 0$,

$$\log p(D|X, \vec{k}) = c + \sum_{t_j} \log f(\vec{k} \cdot \vec{x}(t_j)) - \int f(\vec{x}(t) \cdot \vec{k})dt.$$

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 $\vec{k}$, since in this case the loglikelihood is just a sum of concave functions of $\vec{k}$ (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)).

The gradient and Hessian (second derivative matrix) of the likelihood in this model may be computed very easily. Define the vectors $\vec{f}^{(i)}$ and $\vec{g}^{(i)}$ as

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

and

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

Then it is not hard to see that

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

and similarly,

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

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's update[1] in this model, we find a familiar form:

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

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).

0.2 Generalized linear models for images in low-light conditions; incorporating nonnegativity constraints

While we will spend most of our time discussing point process models in one dimension (i.e., spike trains), similar models are very useful in the case of spatial (i.e., two- or three-dimensional) processes as well. For example, one standard model for images under low-light conditions is as a Poisson process with rate

$$\lambda(x,y) = I(x,y) * w(x,y) = \int I(x,y)w(x-x', y-y')dx'dy',$$

where $I(x,y)$ is the underlying true image intensity, $w(x,y)$ is the point spread function (a positive "blur" function that describes the scatter of photons traveling from the light source to the imaging apparatus), $*$ denotes spatial convolution, and $\lambda(x,y)$ is the rate at which photons hit the detector at position (x,y)[2]. Just as before, we may write down the

[1] Recall the Newton-Raphson method for optimization of the function $f(\vec{x})$, which is based on the second-order Taylor expansion of $f(\vec{x})$ about the current guess $\vec{x}_0$ at the best $\vec{x}$:

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

Thus, iterating

$$\vec{x}_{i+1} = \vec{x}_i - \left(\nabla\nabla_{\vec{x}} f(\vec{x}_i)\right)^{-1} \nabla_{\vec{x}} f(\vec{x}_i)$$

is known to lead to an efficient method for optimizing $f(\vec{x})$ (though note that this method is not always stable, due to the possible singualarity of the second derivative matrix $\nabla\nabla_{\vec{x}} f(\vec{x}_i)$).

[2] In higher light conditions this Poisson model, which is most plausible on physical grounds, is typically approximated with the somwehat simpler Gaussian model

$$n(x,y) = I(x,y) * w(x,y) + \sigma\epsilon(x,y),$$

with ϵ standard white Gaussian noise. This model is justified by the fact that for large values of λ, the $Poiss(\lambda)$ distribution is well-approximated by a Gaussian (although in this case, to simplify, we drop the dependence of the noise scale σ on λ).

loglikelihood $\log p(\{n(x,y)\}|\{I(x,y)\}, w(.)) =$

$$\sum_t n(x,y) \log \left(\sum_{x',y'} I(x,y) w(x-x', y-y') dx dy \right) - \left(\sum_{x',y'} I(x,y) w(x-x', y-y') dx dy \right),$$

with $n(x,y)$ denoting the observed photon count in the pixel located at position (x,y). Clearly the loglikelihood is concave in the vector $\{w(.,.)\}$, just as in the spike train GLM case. Thus, if we have some true test image $I(x,y)$ which can be observed directly, we may estimate the point spread function $w(.,.)$ easily by maximizing the likelihood of the observed photon counts $n(x,y)$ given the known image $I(x,y)$, under the non-negativity constraints $w(.,.) \geq 0$; this is a concave optimization problem on a convex set, mathematically exactly analogous to the spike train case considered above, if we identify the known stimulus $\vec{x}$ with the known image I, and the unknown linear filter $\vec{k}$ with the unknown linear filter w.

0.3 Sampling properties of the MLE; connections to spike-triggered averaging

In the above we have developed a class of models for which the ML estimator is especially computationally tractable. However, we have not yet said much about how good an estimator the MLE actually is: for example, does the MLE asymptotically provide the correct $\vec{k}$, given enough data (i.e., is the MLE consistent for $\vec{k}$? 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 $\vec{k}$ is considered unknown): in this (admittedly idealized) case, standard likelihood theory (Schervish, 1995) establishes that the MLE is consistent for $\vec{k}$, 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 familiar 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 have:

Proposition 1. *The MLE based on any convex and log-concave f is consistent almost surely 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 $\vec{k}$. In particular, again, the assumption on $p(\vec{x})$ holds if $\vec{x}$ is drawn from any Gaussian distribution. (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 (Chichilnisky, 2001; Paninski, 2003).

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 $\alpha\vec{k}_0$, for some $\alpha \neq 0$. We have

$$E[\log p(D|X,\theta)] = \int p(\vec{x}) \left[g(\vec{k}_0\vec{x}) \log f(\vec{k}\vec{x}+b) - f(\vec{k}\vec{x}+b) \right] d\vec{x}.$$

The key fact about this function is that it is concave in $(\vec{k}, b)$ and, after suitable change of variables (multiplication by a whitening matrix), symmetric with respect to reflection about the $\vec{k}_0$ axis. This immediately implies that a maximizer lies on this axis (i.e., is of the form $\alpha\vec{k}_0$, 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. □

As noted above, Proposition 1 bears a striking similarity to the main result for the STA (Bussgang, 1952; Chichilnisky, 2001; Paninski, 2003); the conditions ensuring their asymptotic accuracy 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

$$\frac{1}{T}\sum_i \vec{x}_i \frac{f'}{f}(\vec{k}_{MLE}\vec{x}_i + b_{MLE}) = \int_0^T f'(\vec{k}_{MLE}\vec{x}_t + b_{MLE})dt = \int p(\vec{x}) f'(\vec{k}_{MLE}\vec{x} + b_{MLE})\vec{x}, \quad (2)$$

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 $C\vec{k}_{MLE}$ (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 $\frac{f'}{f}(\vec{k}\vec{x}_i + b)$ positive and monotonically non-increasing in $\vec{k}\vec{x}_i$, 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 $\vec{k}_{WSTA}$, for weighted STA, we have that the MLE asymptotically behaves like

$$\vec{k}_{MLE} = C^{-1}\vec{k}_{WSTA};$$

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

0.4 Incorporating nonlinear terms

As we have described the GLM above, it may appear that the model is restricted to including only linear dependencies on the stimulus $\vec{x}(t)$, through the $\vec{k}\cdot\vec{x}(t)$ term. However, just as in the

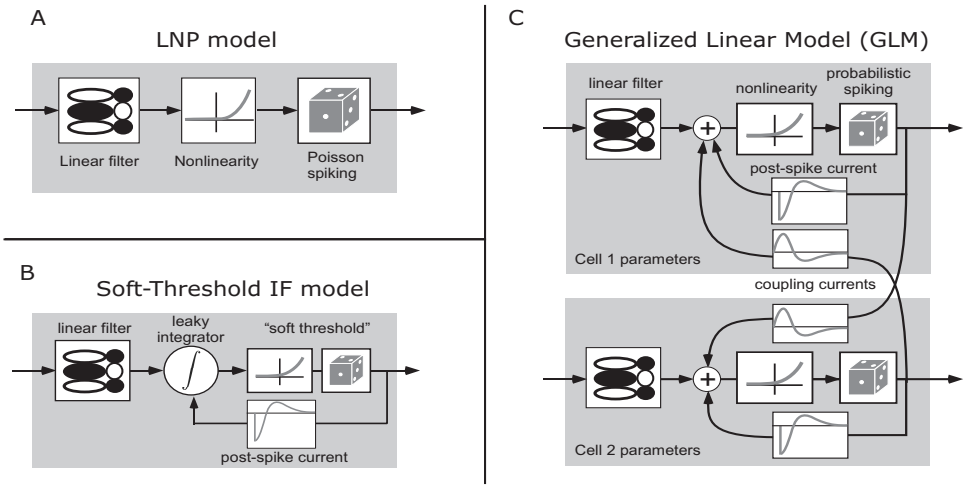

Figure 1: Schematic diagrams of some of the encoding models discussed here. **A**: the linear-nonlinear-Poisson (LNP) model is strictly feedforward, with no spike-history terms. **B**: Illustration of the connection between the GLM with spike-history terms and the integrate-and-fire cell with a probabilistic ("soft") threshold. **C**: GLM incorporating both spike-history and interneuronal coupling terms $h(.)$.

linear regression case, if we modify our input matrix X to include nonlinear transformations $\mathcal{F}_j(\vec{x})$ of the stimulus $\vec{x}$ — $X_t = \begin{pmatrix} 1 & \vec{x}(t)^T & \mathcal{F}_1(\vec{x}) & \mathcal{F}_2(\vec{x}) & \dots & \mathcal{F}_m(\vec{x}) \end{pmatrix}$ — it is clear that we may fit nonlinear models of the form $\lambda(t) = f(b + \vec{k}^T\vec{x} + \sum_j a_j \mathcal{F}_j(\vec{x}))$ easily by maximizing the loglikelihood $\log p(D|X, b, \vec{k}, \vec{a})$ with respect to the parameters $(b, \vec{k}, \vec{a})$ (Wu et al., 2006; Ahrens et al., 2006). Mathematically, the nonlinearities $\mathcal{F}_j(\vec{x})$ may take essentially arbitrary form; physiologically speaking, it is clearly wise to choose $\mathcal{F}_j(\vec{x})$ to reflect known facts about the anatomy and physiology of the system (e.g., $\mathcal{F}_j(\vec{x})$ might model inputs from a presynaptic layer whose responses are better-characterized than are those of the neuron of interest (Rust et al., 2006)).

0.5 Incorporating spike history effects and interneuronal interactions

Above we have described how to adapt standard spike-triggered averaging techniques for the GL model. However, it is not immediately obvious, from a physiological point of view, what we have gained by this exercise. More importantly, it is clear that this simple model suffers from a number of basic deficiencies: for example, the fact that we have assumed that the nonlinearity $f(.)$ is a convex function implies that the firing rate of our basic LNP model does not saturate: as we increase the magnitude of the stimulus $\vec{x}$, the rate must continue to increase at least linearly, whereas the firing rate of a real neuron will invariably saturate, leveling off after some peak discharge rate is attained. Moreover, neurons display a number of other related strongly nonlinear effects that are not captured by the model: e.g., refractory effects, burstiness and bistability of responses, and firing-rate adaptation. In other words, it seems the LNP model does not satisfy our first requirement: model (1) is insufficiently flexible to accurately model real spiking responses.

Luckily, it turns out that we may simultaneously fix these problems and greatly enhance

the GLM's flexibility, by the simple trick of enlarging our input matrix X. Recall that in the discussion above, the t-th row of this matrix consisted of the stimulus $\vec{x}(t)$. However, there is no mathematical reason why we cannot incorporate other 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, $\lambda(t) = f(\vec{k} \cdot \vec{x}(t) + b)$, which provides an important degree of flexibility in setting the threshold and baseline firing rate of the model.

More importantly, we may incorporate terms corresponding to the neuron's observed past activity n_{t-j}, to obtain models of the form $\lambda(t) = f(b + \vec{k} \cdot \vec{x}(t) + \sum_{j=1}^{J} h_j n_{t-j})$ (Fig. 1c). Depending on the shape of the "spike history filter" $\vec{h}$, the model can display all of the effects described above (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 $\log p(D|X, \theta) = \sum_t \left(n_t \log f(X_t \cdot \theta) - f(X_t \cdot \theta) dt \right)$, and compute the ML solution for the model parameters $\theta = \{b, \vec{k}, \vec{h}\}$ by a concave optimization algorithm. (Note that, while we still assume that the spike count n_t within a given short time bin is drawn from a one-dimensional $Poiss(\lambda(t)dt)$ distribution, the resulting model displays strong history effects and therefore the output of the model, considered as a vector of counts $D = \{n_t\}$, is no longer a Poisson process, unless $\vec{h} = 0$.) See Fig. 2 for an application to data from a retinal ganglion cell (Uzzell and Chichilnisky, 2004; Pillow et al., 2005b), and Fig. 3 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.

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 is summarized (Fig. 1c):

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

with $\lambda_i(t)$ denoting the instantaneous firing rate of the i-th cell at time t, $\vec{k}_i$ 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 ($\{b_i, \vec{k}_i, \vec{h}_{i',i}\}$), making fitting highly tractable.

0.6 Example: fitting time-varying terms / PSTH estimation

One simple but important example of the GLM approach is as follows:

$$\lambda(t) = f\left(b(t) + \sum_j h(t - t_j) \right).$$

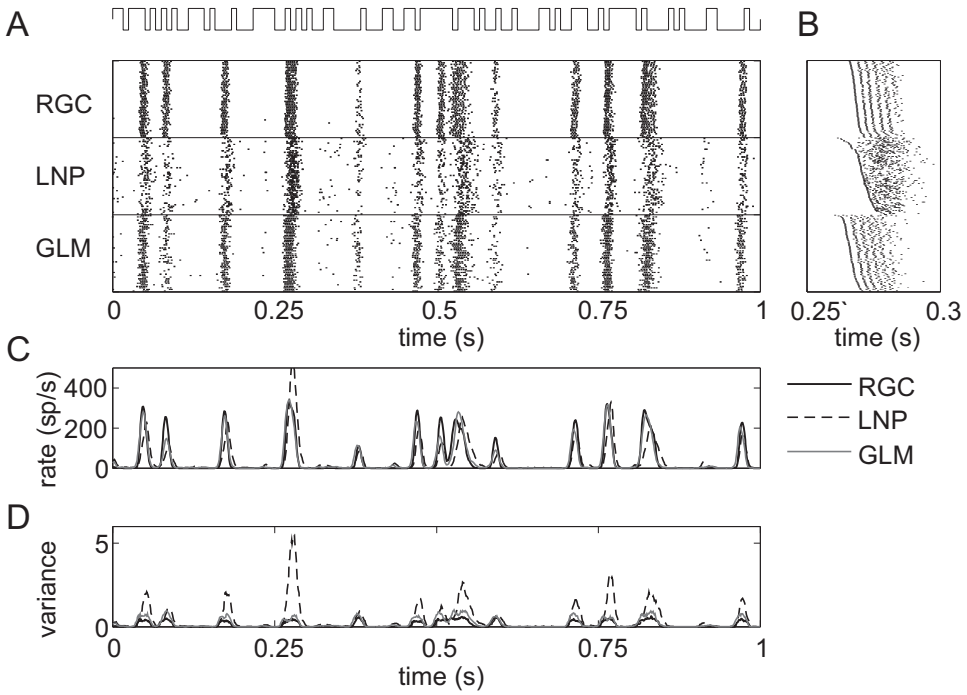

Figure 2: Example predictions of a single retinal ganglion ON-cell's activity using the GL encoding model with and without spike history terms. Conventions are as in Figs. 3 and 4 in (Pillow et al., 2005b); physiological recording details as in (Uzzell and Chichilnisky, 2004; Pillow et al., 2005b). **A**: 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.

Here the term $b(t)$ encodes the dependence of the firing rate as a function of time since the beginning of the trial; the spike history terms $h(.)$ play the same role as above. We may fit $b(.)$ and $h(.)$ simultaneously, due to the joint concavity of the loglikelihood in the parameters $(b(.), h(.))$.

0.7 Multiplicative models of spike history-dependence; connections to renewal models

In the next two sections, we will point out some connections of the above likelihood-based techniques to related models that have appeared in the literature. The first such model we will consider was developed in order to correct the main deficiency of the basic LNP model, namely its Poisson nature. We introduced an "additive" model for incorporating history dependence above (that is, the spike history effects entered in an additive manner, through the terms $\sum_j h_j n_{t-j}$); a simple alternate way to model the spike-history effects (e.g.,

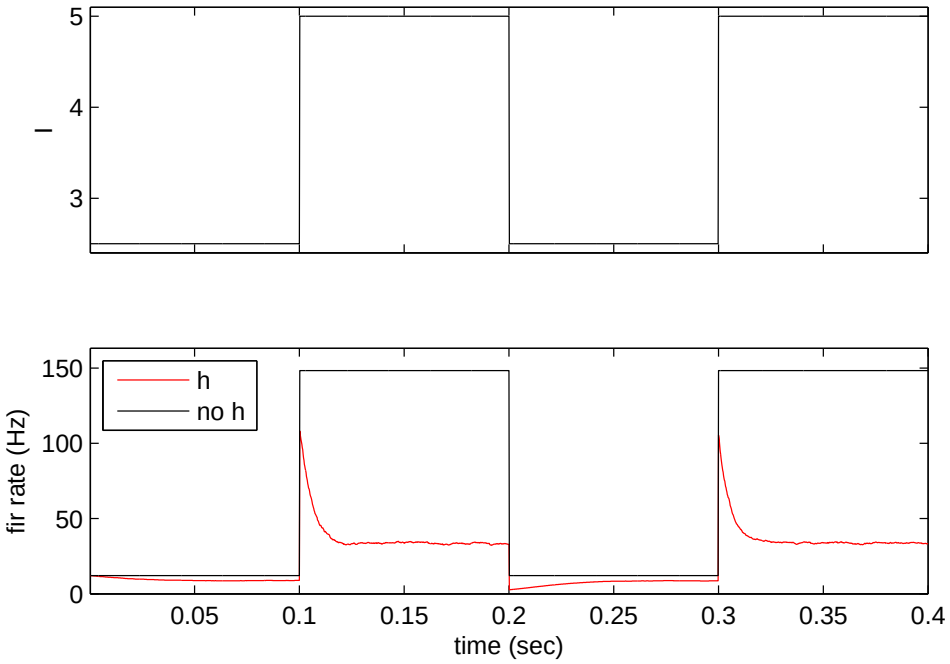

Figure 3: Mean response of a simulated GLM neuron to step inputs. Red trace shows the empirical average firing rate, $E[\lambda(t)]$ (where the expectation is taken over the random spike history effects $\sum_j h_j n_{t-j}$) given the step input $I(t)$. The spike-history function here was chosen to be $h(t) = -3\exp(-t/\tau), t > 0$, 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.

refractoriness, burstiness) that such a Poisson model ignores is to introduce a multiplicative term, dependent on the time since the last spike, to modulate the firing rate; this leads to a kind of "linear-nonlinear-recovery" model (Berry and Meister, 1998; Brown et al., 2002; Paninski, 2003) (a version of the "inhomogeneous Markov interval," or IMI, model discussed in (Kass and Ventura, 2001)). It turns out that this type of model may be solved using likelihood-based tools which are mathematically quite similar to those developed above.

We begin with an arbitrary model (including but not limited to the LNP models considered above; the following results can be stated quite generally). For a given input $\vec{x}$, this model will produce a time-varying predicted firing rate $\lambda_{\vec{x}}(t)$. Now we are going to try to fix up this model by including a multiplicative "recovery" term modeling the effect of the cell's recent spike history, $r(t - t_{i-1})$, with $(t - t_{i-1})$ the time since the last spike; thus, our new model predicts a firing rate

$$\lambda'_{\vec{x}}(t) = \lambda_{\vec{x}}(t) r(t - t_{i-1}).$$

Now how to choose r? As before, a reasonable approach is maximum likelihood: the loglike-

lihood for this process, given a spike train $\{t_i\}$, is again of the form

$$\log p(D|X,\theta) = \sum_i \log(\lambda'_{\vec{x}}(t_i)) - \int \lambda'_{\vec{x}}(t)dt.$$

Rewrite this as

$$\sum_i \log(\lambda_{\vec{x}}(t_i)r(t_i - t_{i-1})) - \sum_i \int_{t_{i-1}}^{t_i} \lambda_{\vec{x}}(t)r(t - t_{i-1})dt$$

$$= K + \sum_i \left(\log r(t_i - t_{i-1}) - \int_{t_{i-1}}^{t_i} \lambda_{\vec{x}}(t)r(t - t_i)dt \right),$$

where K is constant with respect to r whenever $\lambda_{\vec{x}}$ is fixed (i.e., when we are maximizing over all possible history-dependence terms r given a fixed base model $\lambda_{\vec{x}}$).

This loglikelihood is quite convenient. First, it is concave in the recovery function r, implying that no non-global local maxima exist (since r takes values in a convex set, the set of all non-negative functions). In fact, by setting the functional gradient of the likelihood with respect to r to zero, we can find the ML estimate for r in closed form: $r_{MLE}(\tau)$ solves

$$\sum_{i:t_i - t_{i-1} = \tau} \frac{1}{r_{MLE}(\tau)} = \sum_{i:t_i - t_{i-1} \geq \tau} \lambda_{\vec{x}}(t_i + \tau),$$

or written more suggestively,

$$r_{MLE}(\tau) = \frac{p(\tau)}{\int d\vec{x} p(\vec{x}) \lambda_{\vec{x}}(\tau) \int_\tau^\infty p(t|\vec{x})dt},$$

where $p(\tau)$ denotes the observed inter-spike interval (ISI) density, $p(\lambda_{\vec{x}})$ is the observed density of $\lambda_{\vec{x}}(t_i + \tau)$, and $p(t|\lambda_{\vec{x}})$ the observed density of a spike at time t given $\lambda_{\vec{x}}$.

In the case that $p(\lambda_{\vec{x}})$ is concentrated at the point $\lambda_{\vec{x}}(t) \equiv c$ (i.e., our original model is that the spike train is a Poisson process with rate c, independent of the stimulus $\vec{x}$), we have the intuitive solution that

$$r_{MLE}(\tau) = \frac{p(\tau)}{ce^{-c\tau}},$$

the ratio of the observed ISI density $p(\tau)$ to $ce^{-c\tau}$, the ISI density we would expect of a Poisson process of rate c; thus, the MLE agrees in this very special case with the heuristic formula given in (Berry and Meister, 1998).

An obvious approach now for fitting the model parameters $r(.)$ and $\vec{k}$ now is to alternately fit $\vec{k}$ by maximum likelihood, given r (using the obvious modification of the original loglikelihood,

$$\log p(D|X,\vec{k},b,r) \sim \sum_i \left(\log f(\vec{k}\vec{x}(t_i) + b) + \log r(t_i - t_{i-1}) - \int_{t_{i-1}}^{t_i} f(\vec{k}\vec{x}(t) + b)r(t - t_{i-1})dt \right),$$

which retains its concavity in $\vec{k}$ and b for any convex and log-concave nonlinearity $f(.)$, since r is nonnegative), then fit r given $\vec{k}$ (under the model $\lambda_{\vec{x}}(t) = f(\vec{k}^T\vec{x})$), iterating until convergence; since both procedures ascend the likelihood surface, convergence to a local maximum is guaranteed (though not necessarily to a global maximum, since the loglikelihood

is only separately concave in $r(.)$ or $\vec{k}$ if one or the other is held fixed, but not necessarily jointly concave in $(r(.), \vec{k})$.).

This IMI model can also be understood in terms of renewal processes. Consider the case that the base rate is constant, $\lambda_{\vec{x}}(t) \equiv \lambda_0$. Then the resulting spike train is a renewal process with interspike interval density

$$p(t) = e^{-\int_0^t \lambda(s)ds} \lambda(t) = e^{-\lambda_0 \int_0^t r(s)ds} \lambda_0 r(t).$$

Conversely, given any renewal process with ISI density $p(t)$ we may construct an IMI model with post-spike effect $r(t)$ given by the hazard function form

$$\lambda_0 r(t) = \frac{p(t)}{1 - \int_0^t p(s)ds}$$

(note that $r(t)$ is only identifiable up to a constant in general; however, if we make the reasonable assumption that $r(t) \to 1$ for $t \to \infty$ — i.e., that the effect of a spike becomes negligible when the time since the last spike is sufficiently large — then both λ_0 and $r(t)$ may be uniquely determined).

As an application of this equivalence between renewal and IMI models, we may compute the F-I (firing rate versus input) curve for the IMI model. Assume the baseline rate is given by $\lambda_0 = f(I)$, for some montonically increasing function $f(.)$ of the input I. Then we may compute the asymptotic firing rate of the neuron using the renewal theorem:

$$\lim_{T\to\infty} \frac{1}{T} N(T)|I = \left[\int t p(t) dt\right]^{-1} = \left[f(I) \int t e^{-f(I)\int_0^t r(s)ds} r(t) dt\right]^{-1};$$

for some special cases of $r(t)$, the term on the right may be computed analytically, while more generally asymptotic approximations for large $|I|$ are feasible. (Of course we may always compute this quantity numerically.)

In general the additive (GL) and multiplicative models are distinct; for example, the basic multiplicative model only incorporates spike history information over the most recent interspike interval, whereas the additive model may incorporate information over many preceding spikes. However, it is worth noting that in the case of an exponential nonlinearity $f(.) = \exp(.)$ (Paninski et al., 2004a; Truccolo et al., 2005) the multiplicative model may be written as a one-spike modification of the GLM: $\lambda(t) = \exp(\vec{k} \cdot \vec{x}(t)) r(t - t^*) = \exp[\vec{k} \cdot \vec{x}(t) + \log r(t - t^*)]$, and we may estimate $h(.) = \log r(.)$ using the likelihood optimization techniques described above, once X is defined suitably.

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

As emphasized above, one of our key goals in constructing an encoding model is to connect the model parameters to the underlying biophysics and known physiology. Thus it is worthwhile to consider the relationship between the GLM and the more biophysically motivated models employed in studies of intracellular dynamics (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):

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

with initial value

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

namely

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

the linear convolution of the input current with a negative exponential of time constant $1/g$. Here g denotes the membrane leak conductance, $\vec{k} \cdot \vec{x}(t)$ the projection of the input signal $\vec{x}(t)$ onto the spatiotemporal linear kernel $\vec{k}$, 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 $\vec{k}_g \cdot \vec{x}(t) + \vec{h}_g \cdot \vec{n}(t)$, where $\vec{k}_g$ and $\vec{h}_g$ correspond to the original parameters $\vec{k}$ and $\vec{h}$ temporally convolved with the exponential function e^{-gt}; that is, this soft-threshold IF model is just a version of the 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 $\{\vec{k}, h, V_{reset}, b\}$, 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.)

0.9 Avoiding overfitting: training error, generalization, and cross-validation

The key thing to remember about high-dimensional data analysis is that we are looking for models that *predict* the data well, rather than *fit* the data well. For example, we typically measure the quality of a model's fit to data D by the maximum of the likelihood function,

$$L \equiv \max_{\theta \in \Theta} p(D|\theta).$$

Here Θ indexes the parameter set, corresponding to all possible models in the model class under consideration. Clearly, we can always make L larger (in principle) simply by expanding Θ, since adding elements to Θ can never decrease the maximum in the definition of L; for example, in the regression setting, we could increase L by fitting models including both linear and nonlinear terms, rather than just linear terms. Typically, by adding more and more terms in our regression we can fit any data we'd like, in the sense that L becomes arbitrarily large.

There are many problems with this approach of simply iteratively expanding the parameter space Θ to increase the fit quality L. First, of course, the higher the dimensionality of the parameter space (e.g., the more nonlinear, poorly physiologically-justified terms we include in our regression analysis), the more uninterpretable and overly complex our models become. For similar reasons, higher-dimensional models often pose greater computational difficulties than do simpler models. The most important problem with this approach from a statistical point of view, though, is that poor control over the complexity of one's model typically leads to poor predictions; this is the statistical justification for "Occam's razor," the principle that simple explanations are preferred over complex.

A classical example of this phenomenon is shown in Fig. 4. We observe data generated by a smooth curve $g(.)$ (a sum of a few low-fequency sinusoids, in this case) plus i.i.d. Gaussian noise: thus, the i-th sample was given by

$$n_i = g(x_i) + \sigma\epsilon_i,$$

where ϵ_i is i.i.d. standard Gaussian noise. Then we fit a series of models of monotonically increasing complexity to this data: the p-th model class, Θ_p, is the set of all sums of sinusoids of integer frequency less than p. We see, as expected, that the *training error*

$$\frac{1}{N}\sum_{i=1}^{N} Err(g_p(x_i), n_i) = \frac{1}{N}\sum_{i=1}^{N} [\hat{g}_p(x_i) - n_i]^2$$

(with the sum taken over the observed samples n_i and the estimate $\hat{g}_p(.)$ constructed by linear least squares from sinusoids of maximal frequency p) decreases monotonically with the model complexity p, while the *generalization error*

$$E\left[Err(g_p(x), n)\right] = E\left[\hat{g}_p(x) - n\right]^2 = E_{x,\epsilon}\left[\hat{g}_p(x) - (g(x) + \sigma\epsilon)\right]^2$$

(where the expectation is now taken over the true underlying distribution of the data x and noise ϵ, instead of the observed samples) reaches its minimum at the true p (5 Hz in this case) and then increases for larger p. The explanation is that models with larger p fit the data very well at the observed sample points x_i at the expense of large oscillations where no data are observed (i.e., the *noise* has been fit well, not the underlying true function g). Thus the training error curve is completely misleading if we want to understand how well our estimator is actually generalizing, rather than just fitting the data.

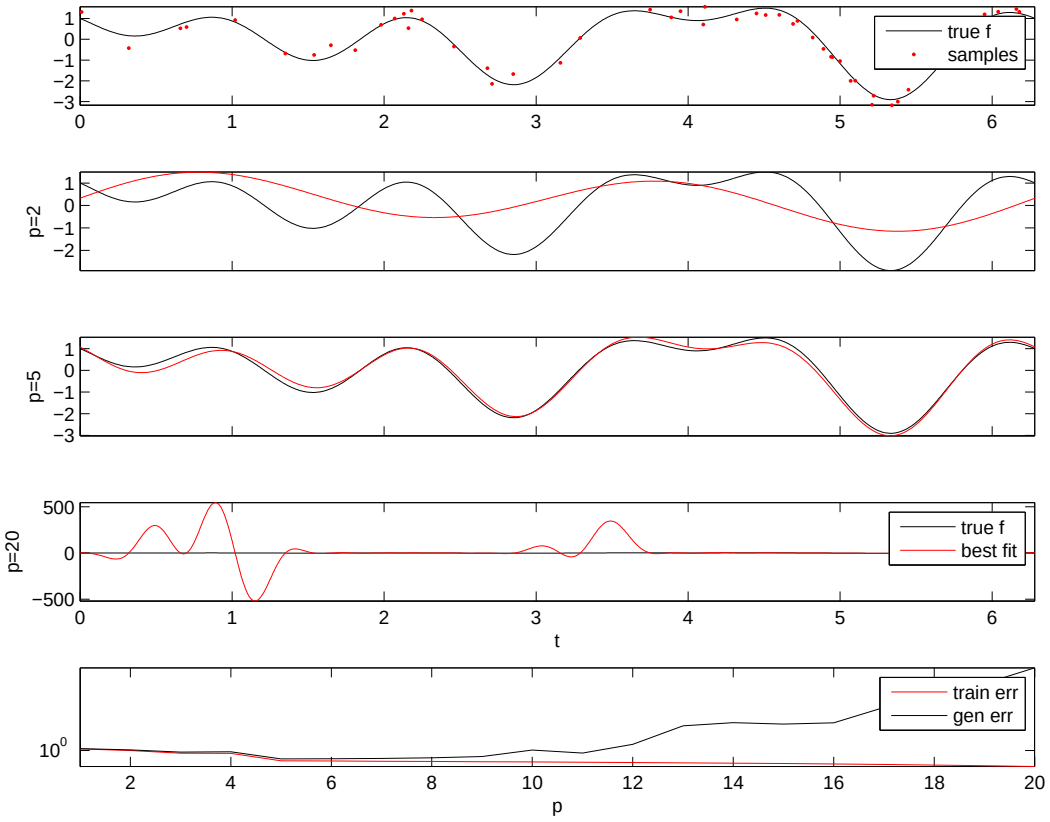

Figure 4: Overfitting demonstration: test vs. generalization error. **Top**: the true function $g(.)$ (solid black) was a sum of three sinusoids, of frequency 2, 3, and 5 Hz. We observe 50 noisy (zero-mean Gaussian; sd=0.3) samples from this true function (red dots). **Middle**: best-fitting function $\hat{g}_p(.)$ (red trace) versus true function $g(.)$ (black), using all sines and cosines of nonnegative integer frequencies, up to and including a maximum frequency $p = 2$, 5, and 20 Hz. Note that the estimate is oversmoothed when the maximal frequency $p = 2$ Hz, and badly overfit for $p = 20$. **Bottom**: training and generalization error as a function of maximal frequency p. Note that the generalization error achieves a minimum at the true value $p = 5$, and increases for higher p (overfitting). The training error decreases monotonically, as it must. (Error curves averaged over 100 i.i.d. experiments; note log axis.)

A geometric analysis of this phenomenon is useful; as usual, we start with the linear regression case to build our intuition. Linear regression may be understood as an intersection of soft constraints, in the following sense. The loglikelihood is simply the quadratic form

$$\log(D|X, \vec{k}) = c - \frac{1}{2\sigma^2} \sum_{i=1}^{N} \left(n_i - \vec{k}^t \vec{x}_i \right)^2,$$

which may be rewritten as

$$-\vec{k}^T A \vec{k} + \vec{b}^T \vec{k} + c,$$

where

$$A \propto \sum_i \vec{x}_i \vec{x}_i^T = X^T X,$$

$$\vec{b} \propto 2 \sum_i n_i \vec{x}_i,$$

and c is a constant which does not affect the location of the optimal $\vec{k}$. Note that each observation $\vec{x}_i$ contributes a rank-one matrix to the resulting $X^T X$ matrix. The geometric interpretation of this sum of rank-one matrices is illustrated in Fig. 5: each sample contributes a term $(\vec{k}^T \vec{x}_i - n_i)^2$ to the overall cost function which serves to constrain the optimal $\vec{k}$ along a single direction in $\vec{k}$-space (because the matrix $\vec{x}_i \vec{x}_i^T$ is of rank one), and we obtain the optimal solution by forming a weighted intersection of these constraints. The directions of low curvature of the resulting quadratic surface are the directions for which $\vec{k}$ is poorly constrained, and which will result in an estimate of $\vec{k}$ which will be highly variable in these directions (in the sense that slight changes in $\vec{n}$ will cause large variations in the least-squares estimate $\hat{k}_{LS}$). Unconstrained directions correspond to zero curvature — infinitely long flat valleys in the cost function. The overfitting phenomenon we observed in Fig. 4 results in exactly this unconstrained case (e.g., when the dimension of $\vec{k}$ is large compared to the number of available samples).

We can translate this geometric intuition into algebra fairly easily: $\hat{k}_{LS}$ is defined as $\hat{k}_{LS} = (X^T X)^{-1}(X^T \vec{n})$, so if $cov(\vec{n}) = \sigma^2 I$, then

$$Cov(\hat{k}_{LS}) = (X^T X)^{-1} X^T \sigma^2 I X (X^T X)^{-1} = \sigma^2 (X^T X)^{-1}.$$

If we express this covariance matrix in terms of its eigenvectors (principal components), $(X^T X)^{-1} = O D^{-1} O^T$, then we see again that the directions of low curvature (small eigenvalues of $X^T X$, i.e., small values of the diagonal matrix D) will correspond exactly to directions of large variance.

The above discussion has been in terms of the linear regression model with Gaussian noise. In the case of our GLM model, the single-observation quadratic loglikelihood term $-\frac{1}{2\sigma^2}(X_t^T \vec{k} - n_t)$ is replaced by the single-observation Poisson loglikelihood $n_t \log f(X_t^T \vec{k}) - f(X_t^T \vec{k}) dt$. Both functions are concave and both depend on $\vec{k}$ 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 $X^T X$ matrix, which sets the inverse covariance of the estimate $\vec{k}$ in the linear regression case, is the "Fisher information" matrix, which we will define and discuss in much more depth below.

These arguments help to explain how overfitting arises, and give us some intuition into what is going on. But how can we avoid overfitting? We describe several methods below. But the major concept to keep in mind (c.f. Fig. 4) is that the training error is misleading; when fitting a model to data we need to quantify the performance of the model in predicting data that was not used to train the model. This practice of quantifying the model's performance on a "test" set of data which is held completely distinct and independent of the "training" set is called "cross-validation."

Thus, to compare the performance of two distinct models, we might fit both models on the same training set, then compute the likelihood of a completely distinct test data set under each of the two models: the model with the higher test likelihood may be considered a better model, in that it is able to predict the responses of the neuron to novel stimuli more

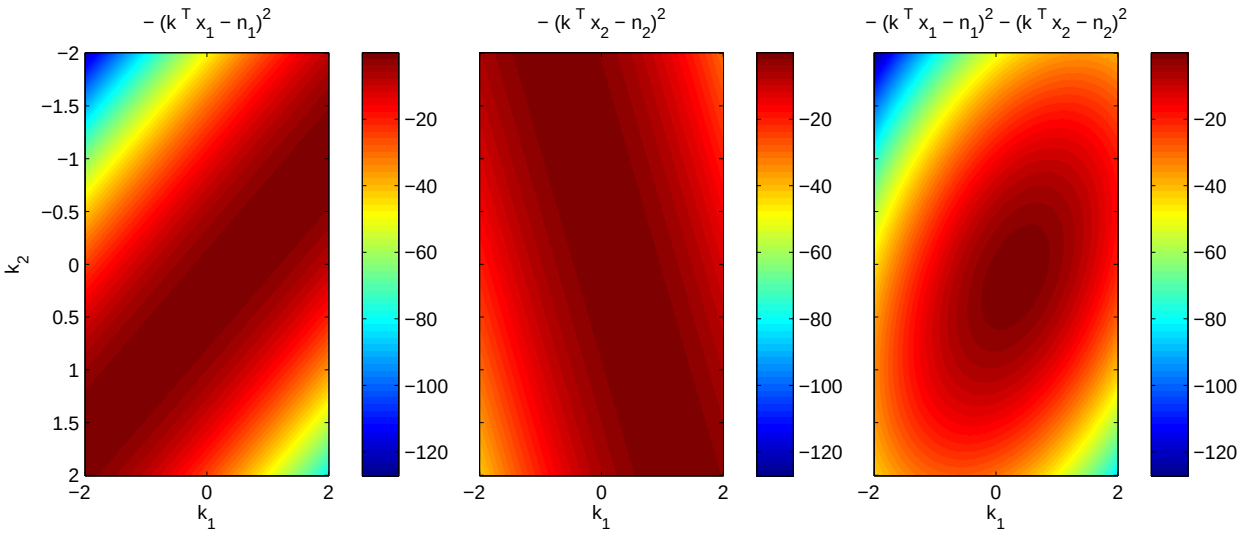

Figure 5: Geometry of least-squares: soft constraints. The left two panels show the cost function $-(\vec{k}^T\vec{x}_i - n_i)^2$ as a function of $\vec{k}$ given two observed data points, $(\vec{x}_1, n_1)$ and $(\vec{x}_2, n_2)$, while the right panel shows the sum of these two terms. Each individual sample $(\vec{x}_i, n_i)$ acts as a soft consraint, restricting $\vec{k}$ in the direction parallel to $\vec{x}_i$ (but not restricting $\vec{k}$ at all in any of the $\dim(\vec{k}) - 1$ other directions); by combining the available data, (i.e., forming what amounts to the weighted intersection of these soft constraints), we obtain a well-defined minimum. Note that the resulting confidence ellipse is tilted in the direction of the first constraint, which was stronger in this case; despite the fact that the constraints were nonorthogonal, the principal axes of the resulting confidence ellipse are orthogonal (as they must be, since these correspond to he eigenvectors of the symmetric matrix $X^T X$).

accurately. It is useful to normalize the loglikelihoods 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.)

0.10 Reducing the number of parameters can increase the prediction accuracy; basis choice

As discussed above, in the linear regression case it is well-known that estimates of the receptive field $\vec{k}$ based on spike-triggered averaging can be quite noisy when $\vec{k}$ has many parameters (Sahani and Linden, 2003; Smyth et al., 2003); the noisiness of the estimate $\vec{k}_{LS}$ is roughly proportional to the dimensionality of $\vec{k}$ (the number of parameters in $\vec{k}$ that we need to estimate from data) divided by the total number of observed samples (Paninski, 2003). The same "overfitting" phenomenon (noisiness increasing with number of parameters) occurs in the GLM context. A variety of methods have been introduced to "regularize" the estimated $\vec{k}$, to incorporate prior knowledge about the shape and/or magnitude of the true $\vec{k}$ to reduce the noise in $\vec{k}_{LS}$. In each case, the goal is to reduce the variance associated with estimating a large number of parameters, at the possible expense of an increase in bias due to a reduction in the flexibility of the model.

One basic idea is to restrict $\vec{k}$ to lie within a lower-dimensional subspace,

$$\vec{k} = \sum_l a_l \vec{k}_l,$$

where $\vec{k}_l$ denotes the l-th basis element (fixed *a priori*); we then employ the same fitting procedure to estimate the coefficients a_l of $\vec{k}$ within this lower-dimensional basis. Plugging in this formula for $\vec{k}$, we have

$$\vec{k}^T \vec{x}_t = (\sum_l a_l \vec{k}_l)^T \vec{x}_t = \sum_l a_l (\vec{k}_l^T \vec{x}_t) = \vec{a}^T \vec{y}_t,$$

where

$$\vec{y}_l = \vec{k}_l^T \vec{x}.$$

Thus fitting these new parameters $\vec{a}$ proceeds in exactly the same fashion as before: we set up a design matrix, $X_t = \vec{y}_t$, and optimize the loglikelihood with respect to the parameters $\theta = \vec{a}$. The goal is to choose a basis whose span contains the "shapes" we might expect $\vec{k}$ to take (in order to minimize the bias associated with restricting our attention to a lower-dimensional subspace of possible $\vec{k}$), with as few basis elements (smallest dimensionality) as possible (since the variance of the estimated $\vec{k}$ is roughly proportional to the dimensionality). Of course, this restriction also increases the computational efficiency of the fitting procedure, since computation time increases with the dimensionality of X.

Some examples:

- 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).

- 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 Markov processes.

- We often have good a priori knowledge about the smoothness of the filter $\vec{k}$: in this case, it is common to represent $\vec{k}$ in a Fourier basis, with the very high- (and/or low-) frequency elements removed from the basis.

- We may take a similar approach using an orthogonal basis defined by principal components analysis (PCA) instead of Fourier analysis (in some cases the two approaches coincide, e.g. in the case of shift-stationary stimuli $\vec{x}$): we perform PCA on $\vec{x}$ and represent $\vec{k}$ in a basis defined by the eigenvectors which capture, say, 90% of the covariance of $\vec{x}$ (Schwartz et al., 2002).

- Other useful bases include the Hermite basis (Victor et al., 2006), the basis introduced by Keat et al. for the representation of temporal receptive fields (Keat et al., 2001; Pillow et al., 2005b), and the Zernike basis for receptive fields defined on a circle (Barbieri et al., 2004).

As discussed above, there are close connections between the regression estimator $\hat{\theta}_{LS} = (X^T X)^{-1}(X^T \vec{n})$ and the GLM maximum-likelihood estimate $\hat{\theta}_{ML}$. Thus it is helpful to choose the basis in such a way that the resulting design matrix X (expressed in the new basis) is close to orthogonal, i.e., that the matrix $(X^T X)$ has a small condition number. This increases the numerical stability as well as interpretability of the resulting estimate $\hat{\theta}_{ML}$.

0.11 Low-rank approximations provide a useful method for reducing the number of parameters

One common case that leads to a very large number of parameters involves the estimation of a spatiotemporal receptive field in vision (Sharpee et al., 2006) or somatosensory studies, or a spectrotemporal receptive field in audition (Sahani and Linden, 2003; Gill et al., 2006). In each case, it is often a reasonable approximation to represent the STRF as a "separable" function of space and time, that is, the product form

$$k(x, y, t) = k_s(x, y)k_t(t).$$

The gain here is that the number of parameters is reduced from ST to $S + T$, where S and T denote the number of parameters required to describe the spatial component $k_s(x, y)$ and the temporal component $k_t(t)$, respectively; typically $ST \ll S + T$.

To fit such a separable model, we may proceed by a simple alternating maximization strategy: if we hold $\vec{k}_t$ fixed, then the log-likelihood is concave with respect to $\vec{k}_s$, and vice versa. Unfortunately, we are no longer guaranteed to find a global maximum using this strategy, despite the fact that the loglikelihood is concave as a function of $\vec{k}$ since the class of receptive fields of this separable form does not form a convex set (the sum of two separable functions is typically not separable). In addition, we must place restrictions on the model to ensure identifiability, since clearly $\left(c\vec{k}_t(t), \frac{1}{c}\vec{k}_s(x,y)\right)$ specifies the same model as $\left(\vec{k}_t(t), \vec{k}_s(x,y)\right)$, for any $c \neq 0$; see (Ahrens et al., 2006) for details.

This separable receptive field idea can be quite useful. However, in many cases the receptive field is highly non-separable. (For example, consider a motion-detecting receptive field,

$$k(x,t) = g(x - vt),$$

for some function $g(.)$ and velocity $v \neq 0$.) In this case we may generalize the separable idea slightly. Consider a one-dimensional spatial variable for simplicity (we may always concatenate the x and y variables in general): we may represent $k(x,t)$ as a matrix K. A separable receptive field corresponds to a matrix of rank one:

$$K = \vec{k}_s \vec{k}_t^T.$$

A natural generalization is to consider a matrix of higher rank r:

$$K = K_s K_t^T,$$

where the vectors $\vec{k}_s$ and $\vec{k}_t$ above have been replaced by the matrices K_s and K_t, of size $S \times r$ and $T \times r$, respectively. As in the separable case, we may fit the model parameters by straightforward alternating maximization: with K_t held fixed, the loglikelihood is concave in K_s, and vice versa. (Once again, certain restrictions on the model are necessary to ensure identifiability.) If $(S+T)r \ll ST$, clearly we achieve a reduction in the number of parameters, often at no great loss of accuracy in the resulting model.

Because of the loss of our convergence guarantees in this low-rank model, the choice of initialization of the parameter search becomes somewhat more important. One useful approach is to take a preliminary estimate of the full-rank K and then to perform a singular value decomposition of the matrix $K = UWV$; then the first r rows of U serve to initialize K_s, while the first r columns (multiplied by the first r singular values in W) serve to initialize K_t.

Another example arises in the multicell case. Let's imagine that we'd like to fit a GLM including interneuronal coupling terms, and that we have reason to believe that all the interneuronal terms $h_{i',i}(t)$ may be formed by taking a weighted sum of a few simple candidate "quasi-synaptic" functions $q_l(t)$, i.e., $h_{i',i}(t) = \sum_l a_{i',i,l} q_l(t)$ for some set of weights $\{a_{i',i,l}\}$. Furthermore, we would prefer not to assume any *a priori* form for $q_l(t)$, but would rather learn these functions directly from the data. If we consider one target neuron i at a time, for simplicity, we need only fit the matrix parameters Q and A in the low-rank model

$$H = QA^T,$$

with $H(t,i') = h_{i',i}(t)$, $Q(t,l) = q_l(t)$, and $A(i',l) = a_{i',i,l}$. If we would like to fit all the terms i simultaneously (i.e., if each cell i receives the same quasisynaptic input shapes $q_l(t)$),

we need only concatenate the i', i terms to form an augmented H and A matrix, and then optimize the full loglikelihood

$$\log p(\{n_i(t)\}|X,\theta) = \sum_i \log p(\{n_i(t)\}_i|X,\theta);$$

conveniently, the likelihood for A decomposes into a set of independent problems which may be solved in parallel, though the optimization in L does not decompose (since the elements of the L matrix affect all cells i; conversely, the elements of the matrix A are not shared between cells i).

We may also apply this low-rank idea to the Volterra series analysis discussed above. The second-order Volterra generalized linear model is

$$\lambda(t) = f\left(b + \vec{k}^T\vec{x} + \vec{x}^T A\vec{x}\right),$$

for some matrix A. As we discussed previously, we require a good deal of data to fit all the elements of the matrix A accurately; instead, we may assume that A is of low rank, $A = A_l A_r$, where A_l and A_r are of size $\dim(x) \times r$ and $r \times \dim(x)$, respectively. Once again, the loglikelihood is concave in the parameters $(b, \vec{k}, A_l)$ with A_r held fixed, or in $(b, \vec{k}, A_r)$ with A_l held fixed, and so an alternating maximization approach is natural. (One small note: since

$$\vec{x}^T A_l A_r \vec{x} = \vec{x}^T A_r A_l \vec{x} = \vec{x}^T \left(\frac{A_l A_r + A_r A_l}{2}\right)\vec{x},$$

we are actually fitting a rank-$2r$ model for A here, instead of the usual rank-r model, since the matrix $(A_l A_r + A_r A_l)/2$ has rank $2r$.)

One key example of this low-rank Volterra model is the classical "energy model" for complex cells in primary visual cortex (Adelson and Bergen, 1985; Okajima and Imaoka, 2001). In this model the firing rate is given by

$$\lambda(t) = f\left((\vec{k}_1^T\vec{x})^2 + (\vec{k}_1^T\vec{x})^2\right),$$

where the linear filters $\vec{k}_1$ and $\vec{k}_2$ are in quadrature pair (and therefore the response of the model is invariant with respect to the phase of the stimulus $\vec{x}$). This may be rewritten in more standard Volterra form as

$$f\left((\vec{k}_1^T\vec{x})^2 + (\vec{k}_1^T\vec{x})^2\right) = f(\vec{x}^T A\vec{x} + b^T\vec{x}),$$

with $A = \vec{k}_1\vec{k}_1^T + \vec{k}_2\vec{k}_2^T$ and $b = 0$; thus the energy model may be considered a rank-two Volterra model.

A final useful application appears in (Ahrens et al., 2006). In many cases we might not know a neuron's "preferred units" a priori: for example, it might make more sense to represent $\vec{x}$ in logarithmic instead of linear units, or perhaps the neuron's response is invariant with respect to the sign of $\vec{x}$; we would like to learn this representation directly from data. Thus we might fit an "input nonlinearity" model, of the form

$$\lambda(t) = f\left[\sum_i a_i g(x(t-i))\right],$$

where $x(t)$ is the scalar input at time t, $g(.)$ is an unknown nonlinearity which transforms this input (for example, $g(.)$ could apply a logarithmic or squaring transformation), $\vec{a}$ is a temporal filter, and $f(.)$, as usual, is a convex and log-concave scalar function. To fit this model we represent the input nonlinearity function $g(.)$ as a weighted sum of some set of known functions $g_l(.)$,

$$g(u) = \sum_l b_l g_l(u),$$

and rewrite

$$\lambda(t) = f\left[\sum_i a_i g(x(t-i))\right] = f\left[\sum_i a_i \sum_l b_l g_l(x(t-i))\right] = f\left[\sum_{il} a_i b_l g_l(x(t-i))\right].$$

Now if we think of the fixed (known) stimulus terms $g_l(x(t))$ as elements of a matrix indexed by t and l, we may reinterpret the double sum

$$\sum_{il} a_{t-i} b_l g_l(x(t)) = \sum_{il} K_{il} g_l(x(t))$$

as a sum over a rank-one matrix $K = \vec{a}\vec{b}^T$, just as in the examples given above. Of course, it is now straightforward to generalize further, to let the matrix K be of rank r, for example, or to use the same trick to infer more complex linear-nonlinear-linear-etc. cascade models. See (Ahrens et al., 2006) for details.

0.12 Regularization; maximum penalized likelihood; maximum a posteriori estimation

Above we discussed one tractable way to avoid overfitting, by restricting our attention to a linear subspace (or, in the low-rank case, a submanifold) of the full parameter space. This corresponds to enforcing "hard" constraints on the acceptable parameter values. A slightly less restrictive approach is to use "soft" constraints instead — that is, to penalize some parameters but not to disallow them entirely. This penalization may be interpreted easily in Bayesian terms: instead of maximizing the loglikelihood $\log p(D|X, \vec{k})$ directly (which can lead to overfitting), we maximize the logarithm of the *posterior*

$$\log p(\vec{k}|X, D) = c + \log p(D|X, \vec{k}) + \log p(\vec{k})$$

(with $\vec{k}$ allowed to take values in the full original parameter space, i.e., no hard constraints have been imposed); here $p(\vec{k})$ encodes our *a priori* beliefs about the true underlying $\vec{k}$, and if we set $-Q(\vec{k}) = \log p(\vec{k})$, we see that $Q(\vec{k})$ acts as a kind of "penalty function," encoding our preferences in more *a priori* probable values of $\vec{k}$ (or equivalently, penalizing less probable values).

In the linear regression case, the computationally-simplest prior is a zero-mean Gaussian,

$$\log p(\vec{k}) = c - \vec{k}^T A \vec{k}/2,$$

where A is a positive definite matrix (the inverse prior covariance matrix); maximizing the corresponding log-posterior

$$\log p(\vec{k}|X, D) = c + \log p(D|X, \vec{k}) + \log p(\vec{k}) = c - \frac{1}{2\sigma^2}||X^T\vec{k} - \vec{n}||_2^2 - \frac{1}{2}\vec{k}^T A \vec{k}$$

analytically leads directly to the regularized least-square estimator

$$\vec{k}_{RLS} = (X^T X + \sigma^2 A)^{-1} X^T \vec{n}$$

(Sahani and Linden, 2003; Smyth et al., 2003). It is easy to incorporate this maximum *a posteriori* idea in the GLM context as well (Paninski, 2004) (though once again we lose the nice analytic form of the optimal solution): we simply maximize

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

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

Thus we are free to choose the penalty term Q as we like within the class of smooth convex functions, bounded below. One of the most common penalties acts to smooth the resulting estimate (Smyth et al., 2003). For example, our prior might express that smooth $\vec{k}$ are more common than rapidly changing or highly fluctuating $\vec{k}$. If we express this prior in the Gaussian form described above, $\log p(\vec{k}) = c - \vec{k}^T A \vec{k}/2$, then we might choose the matrix A such that

$$\vec{k}^T A \vec{k} = \sum_i [k(i) - k(i+1)]^2 = ||D\vec{k}||_2^2 = \vec{k}^T D^T D \vec{k},$$

with D denoting the discrete difference matrix, i.e., large changes between adjacent elements $\vec{k}(i)$ of $\vec{k}$ are penalized. Clearly A here may be written as $A = D^T D$, which turns out to correspond to the symmetric second difference matrix. (Of course it is also possible to penalize higher-order derivatives.)

It is also worth mentioning how to implement this penalty in the case that $\vec{k}$ is expressed in some alternate basis, $\vec{k} = \sum_l a_l \vec{k}_l = K\vec{a}$ for a suitable basis matrix K. If we write out the penalty in this case,

$$\vec{k}^T A \vec{k} = \vec{k}^T D^T D \vec{k} = \vec{a}^T K^T D^T D K \vec{a} = \vec{a}^T B \vec{a},$$

where the elements of the matrix $B = K^T D^T D K$ are given by the inner product of the differenced basis elements,

$$B_{l,l'} = (D\vec{k}_l)^T (D\vec{k}_{l'}).$$

Note that precomputing B and maximizing the posterior with respect to the parameters $\vec{a}$ is typically more computationally efficient than recomputing $\vec{k} = K\vec{a}$ on each iteration.

More generally, if Q is minimized (i.e., if $\log p(\vec{k})$ is maximized) at the point $\vec{k} = \vec{0}$, the MAP estimator will basically be a more "conservative" version of the MLE, with the chosen coefficients shrunk nonlinearly towards zero. This type of "shrinkage" estimator has been extremely well-studied, from a variety of viewpoints (James and Stein, 1960; Donoho et al., 1995; Klinger, 1998; Tipping, 2001; Ng, 2004), and is known, for example, to perform strictly better than the MLE in certain contexts: again, because this shrinkage can effect a large decrease in the variance of our estimator, at the expense of a small increase in the bias. See (Sahani and Linden, 2003; Machens et al., 2003; Harris et al., 2003) for some illustrations of this effect.

The simplest version of this shrinkage idea is to choose the matrix A in the quadratic form for Q to be proportional to the identity. Thus we are penalizing the magnitude $\vec{k}^T\vec{k}$ directly instead of the magnitude of $D\vec{k}$, as in the smoothing case. This form of direct "shrinkage" has been studied extensively and is also known as "ridge regression" or Tikhonov regularization, depending on the literature. Note that the eigenstructure of $X^TX + \lambda I$ is easily derived from that of X^TX: the eigenvectors are exactly the same, and the eigenvalues are merely changed by the constant value λ. The key fact is that any zero eigenvalues in X^TX which might have caused problems in computing the inverse $(X^TX)^{-1}$ are now strictly positive, making the inverse much more stable and insensitive to noise.

Another very common penalizer is based on the L_1 norm of $\vec{k}$, $Q(\vec{k}) = \sum_i |\vec{k}(i)|$, instead of the L_2 norms we have discussed above[3]. This L_1 penalty is often used as a "sparseness" penalty: in many cases, we might believe that many of the elements of $\vec{k}$ are exactly zero, i.e., that only a "sparse" subset of $\vec{k}$ are actually active. One reasonable penalty to enforce sparseness would be the so-called L_0 norm,

$$||\vec{k}||_0 = \sum_i \delta(\vec{k}_i) = \lim_{p \to 0} ||\vec{k}||_p$$

(where as usual the Dirac delta function $\delta(.)$ is one at zero and zero everywhere else); unfortunately, this L_0 norm is nonconvex and the resulting minimization problem is often plagued by multiple local optima. The absolute value function $|x|$ is in a sense the closest convex function to the discontinuous function $1(x = 0)$, and so we often use the L_1 norm to impose sparseness. A great deal of recent research has focused on the properties of this L_1 penalty (also known as the "LASSO" in the statistics literature (Donoho et al., 1995; Tibshirani, 1996)); for example, it has recently been established that, under certain circumstances, the L_0- and L_1-penalized regression problems have exactly the same solution, i.e., the L_1 term really does serve to sparsen (Donoho and Elad, 2003).

It is worth comparing the L_1 versus L_2 penalization in a simple one-dimensional case, in order to gain some intuition into the behavior of these penalizers. As always, we turn to the quadratic-loss case for simplicity: if our original loglikelihood can be written in the form

$$-\frac{a}{2}\theta^2 + b\theta + c$$

for some coefficients (a, b, c), with $a > 0$, then adding an L_2 penalty term on θ, $-(a_0/2)\theta^2$, corresponds to changing the objective function to

$$-\frac{a + a_0}{2}\theta^2 + b\theta + c,$$

with the penalized optimum $\theta = b/(a + a_0)$, as compared to the unpenalized optimal $\theta = b/a$. Thus we see that the L_2 penalty term simply shrinks the optimal θ by a multiplicative factor of $a/(a + a_0)$ — and therefore, the larger the optimal original θ is, the larger the absolute shrinkage will be.

[3]The L_p norm of a vector $\vec{k}$ is a measure of the magnitude of $\vec{k}$, defined as

$$||\vec{k}||_p = \left(\sum_i |k(i)|^p\right)^{1/p}.$$

For $p \geq 1$, this is a convex function of $\vec{k}$.

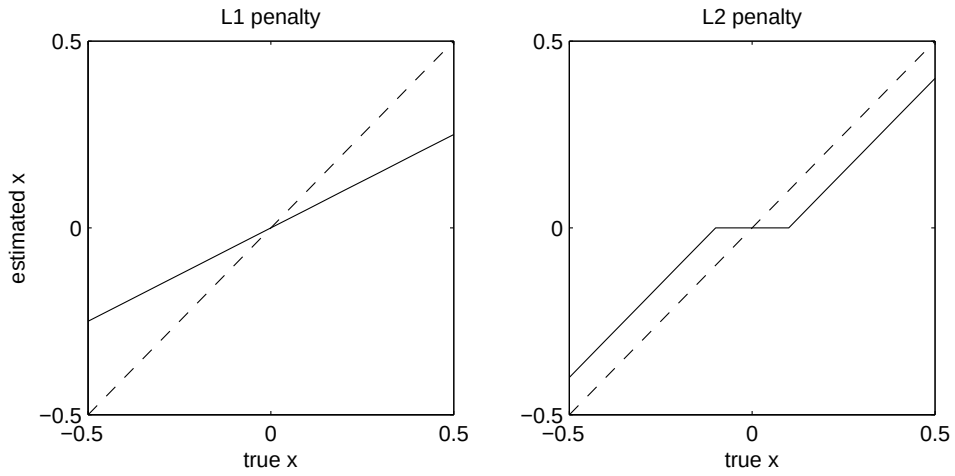

Figure 6: Comparison of L_2 and L_1 penalties for one-dimensional observations x. Note that the L_2 penalty changes the slope of the line $\hat{x}(x)$ (i.e., larger x are shrunk more), while the L_1 penalty leads to threshold behavior in $\hat{x}$ (estimates corresponding to small x are set to zero, but medium and large x are shrunk equally). Identity line (dashed trace) shown for comparison.

The L_1 penalty term behaves differently in two respects. First, instead of a multiplicative shrinkage we have an additive shrinkage: the total shrinkage does not increase as a function of the absolute value of the unpenalized optimizer. Second, and related, it is easy to see that the L_1 optimizer has a threshold nature: if the unpenalized θ has small enough magnitude, θ will be shrunk all the way to zero. (Clearly the L_2 penalty will never set θ to zero exactly, unless the original unpenalized θ is itself zero.)

In the case of multidimensional $\vec{k}$, the geometric interpretation of the two penalties is helpful and fairly natural. See Fig. 7 for an illustration: the key point is that the L_2 is radially symmetric — no elements of $\vec{k}$ are preferred, and thus in a sense all directions are shrunk equally. In the L_1 case, this symmetry no longer holds: it is clear that sparse solutions are favored, since the L_1 penalty is smaller along the coordinate axes (where some components of $\vec{k}$ are set to zero) than along other directions.

One final regularizer is most useful in the case of a matrix $\vec{k}$, and combines two of the useful properties of the L_1 and L_2 penalties. Imagine for concreteness that we are fitting a model for the firing rate of cell i, in which we are incorporating inputs from other cells i'. We might believe that the connectivity from neurons i' to i is sparse in the sense that only a few cells i' are connected to i. We might fit a matrix $h_i(i', \tau)$ of inputs, indexed by time delay τ and cell i'. Enforcing sparseness in this matrix by an L_1 penalty $Q(h) = \sum_{i',\tau} |h_i(i', \tau)|$ does not give us exactly what we want, unfortunately: this might lead to a sparse $h_i(.,.)$ matrix overall (in the sense that only a few values $h(i', \tau)$ are nonzero), but may leave many cells i' connected to i (albeit at only a sparse subset of delays τ). A better penalty in this case is

$$Q(h) = \sum_{i'} ||h_i(i', .)||_2 = \sum_{i'} \left(\int |h_i(i', t)|^2 dt \right)^{1/2} :$$

i.e., enforce sparseness on the number of i' terms for which $h_i(i', .)$ is nonzero at any time t

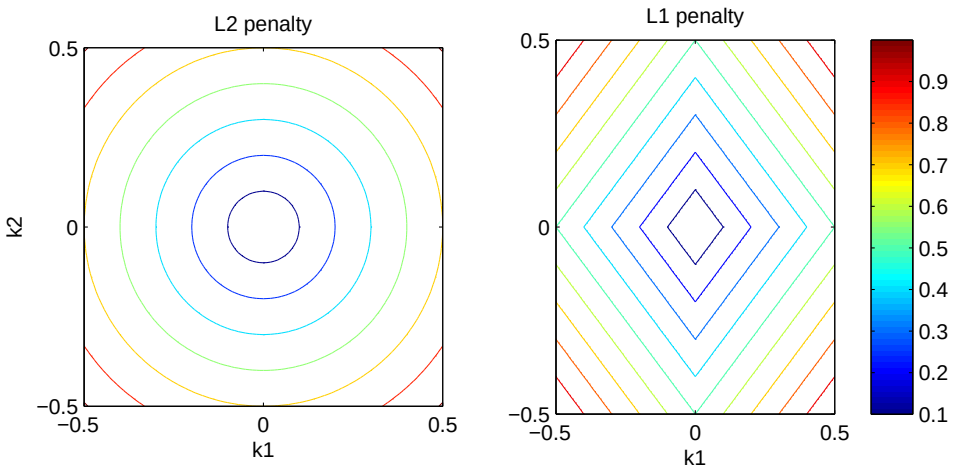

Figure 7: Comparison of L_2 and L_1 penalties for a two-dimensional $\vec{k}$. Note the radial symmetry of the L_2 penalty and the preference for the coordinate axes (the "corners") in the L_1 case.

(the L_2 norm in this case provides a nice, radially-symmetric way to detect departures from zero in the vectors $h_i(i', .)$). Because this penalty is just a sum of convex functions (the p-norm is convex for any $p \geq 1$), the penalty remains convex. This technique is referred as a "groupwise LASSO" in the statistics literature.

This regularization approach does have a somewhat unpleasant side effect. While a smoothing penalty can greatly improve the shape of the resulting estimates, and the L_1 approach can perform feature selection (by "turning off" those features $k(i)$ that do not contribute any predictive power), both of these approaches also result in a "shrunk" estimate: the overall magnitude of the estimate is often reduced. We can often achieve better generalization performance if we undo this shrinkage, while maintaining the shape or the "correct" features obtained by the regularization approach. (Of course, in the simplest L_2 case, undoing this shrinkage is a bad idea, since it undoes the symmetric shrinkage which was the whole point of the L_2 penalty.) Luckily, undoing this unwanted shrinkage is often fairly straightforward. For example, in the context of a smoothing penalty, we simply optimize the likelihood along a one-dimensional line corresponding to a magnitude $\alpha > 0$: that is, we choose $\hat{k}_0$ as the optimizer of the log-posterior $\log p(\vec{k}|X, D)$, but then choose our final estimate $\hat{k} = \alpha_{opt}\hat{k}_0$, where a_{opt} is the solution to the one-dimensional concave optimization problem

$$\alpha_{opt} = \arg\min_{\alpha>0} \log p(D|X, \alpha\hat{k}),$$

where note we are performing this last linesearch over the unpenalized likelihood, not the full posterior. This retains the smooth shape of the estimate but does not result in a reduced magnitude. Similarly, if we have used an L_1 penalty to choose a predictive subset of features $k(i)$, we may undo the L_1 shrinkage by performing an unpenalized "post-fit" on the subspace spanned by this reduced subset of features.

Finally, it is quite common to solve a slightly more general problem: instead of maximizing the log-posterior we might instead minimize

$$\log p(D|X, \vec{k}) - \lambda Q(\vec{k}),$$

where $\lambda > 0$ is a free "regularization parameter": for λ large, we penalize strongly, while for $\lambda \to 0$ we recover the unregularized maximum likelihood solution, and $\lambda = 1$ gives us the original MAP solution. Varying this parameter λ gives us some extra flexibility, but of course we need some way of selecting the best value of this parameter: this may be done either by cross-validation (Machens et al., 2003; Smyth et al., 2003) or by an approach known as "evidence optimization" (Tipping, 2001; Sahani and Linden, 2003), a somewhat more involved technique based on integrating out hyperparameters in a Bayesian hierarchical framework; see e.g. (Sahani and Linden, 2003) for further details.

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

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

We can measure the scale of the posterior distribution along an arbitrary axis in a fairly simple manner: since we know (by the above concavity arguments) that the posterior is characterized by a single "bump," and the position of the peak of this bump is already characterized by 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 "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 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 (Huys et al., 2006; Lewi et al., 2006) (for this reason J is referred to as the "observed Fisher information matrix" in the statistics literature (Schervish, 1995)).

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,

$$p(\theta|X, D) \approx \left(\frac{|J|}{2\pi}\right)^{\dim(\vec{\theta})/2} \exp\left(-\frac{1}{2}(\vec{\theta} - \theta_{MAP})^T J(\theta - \theta_{MAP})\right), \tag{3}$$

where J here plays the role of the inverse covariance matrix: $cov(\vec{k}) \approx J^{-1}$.

Now errorbars around a single parameter may be formed by computing the marginal

standard deviation under this Gaussian model[4]:

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

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

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

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

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

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

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

[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 $J_{ii}^{-1} \leq \left(J^{-1}\right)_{ii}$.

Note in the special case that all the J_i matrices are the same, 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 $\theta_{MAP,i}$.

0.14 The 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) (the classical experimental design setting concerns how to construct the design matrix X optimally in a standard linear regression setting). The objective is to select, in an online, closed-loop manner, the stimuli that will most efficiently characterize the neuron's response properties (Fig. 8a).

An important property of GL models is that not all stimuli will provide the same amount of information about the unknown coefficients $\vec{k}$. As a concrete example, we can typically learn much more about a visual neuron's response properties if we place stimulus energy within the receptive field, rather than "wasting" stimulus energy outside the receptive field. To make this idea more rigorous and generally applicable, we need a well-defined objective function that will rank any given stimulus according to its potential informativeness. Numerous objective functions have been proposed for quantifying the utility of different stimuli (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 $\vec{k}$ — 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 minimizing the determinant is equivalent to minimizing the posterior entropy, and minimizing the average entropy (maximizing the information) can be expected in many cases to roughly minimize the average determinant.

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 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. 8b), with the running time growing only as $O[\dim(\theta)^2]$ (as opposed to the 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. 8c); see (Lewi et al., 2006) for details.

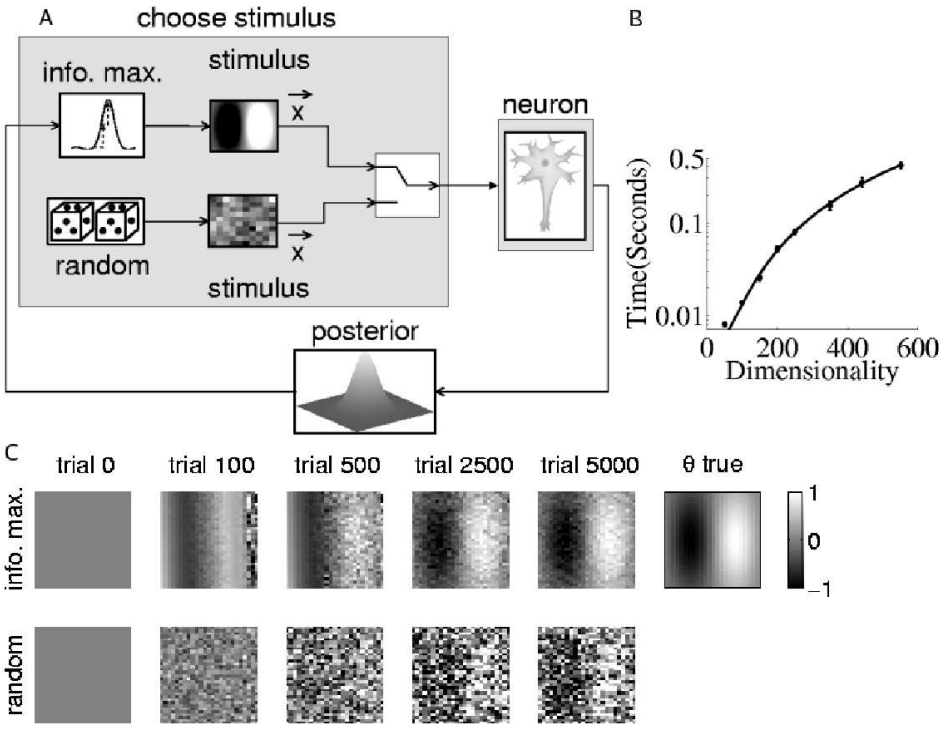

Figure 8: A) Closed-loop vs. open-loop stimulus design. B) Plot of the total running time on a desktop computer for each iteration of the model-based stimulus optimization algorithm, as a function of the dimensionality of the stimulus $\vec{x}$. A quadratic polynomial ($O[\dim(\vec{k})^2]$) fits the data quite well; note that < 15 ms are necessary to optimize a 100-dimensional stimulus. C) Plots of the estimated receptive field for a simulated visual neuron whose responses were generated by a GLM. The neuron's true receptive field $\vec{k}$ has the Gabor structure shown in the last panel; the nonlinearity $f(.)$ was assumed known *a priori* and the spike-history terms were assumed to be zero, for simplicity. Individual panels show $\vec{k}_{MAP}$ after observing t stimulus-response pairs (the prior $p(\vec{k})$ was taken to be Gaussian with mean zero), comparing the accuracy of the estimates using information-maximizing vs. random stimuli (all stimuli were constrained to have unit norm, $||\vec{x}||_2 = 1$ here); the closed-loop approach is an order of magnitude more efficient in this case. See (Lewi et al., 2006) for details.

0.15 Extensions: latent variable models

It should be clear that the GLM encoding framework described here can be extended in a number of important directions. One example we will return to in some detail is as follows. In many cases it is reasonable to include terms in X that we may not be able to observe or calculate directly (e.g., intracellular noise, or the dynamical state of the network); fitting the model parameters in this case requires that we perform a kind of average over these "latent," unobserved variables, e.g. via the expectation maximization algorithm (Smith and Brown, 2003; Paninski et al., 2004c; Kulkarni and Paninski, 2006). We will return to this topic after building some additional analytic tools.

References

Adelson, E. and Bergen, J. (1985). Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2:284–299.

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

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

Berry, M. and Meister, M. (1998). Refractoriness and neural precision. *J. Neurosci.*, 18:2200–2211.

Brown, E., Barbieri, R., Ventura, V., Kass, R., and Frank, L. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14:325–346.

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.

Donoho, D. and Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via L^1 minimization. *PNAS*, 100:2197–2202.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995). Wavelet shrinkage: Asymptopia? *J. R. Statist. Soc. B.*, 57(2):301–337.

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

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

Gill, P., Zhang, J., Woolley, S., Fremouw, T., and Theunissen, F. (2006). Sound representation methods for spectro-temporal receptive field estimation. *Journal of Computational Neuroscience*, In press.

Harris, K., Csicsvari, J., Hirase, H., Dragoi, G., and Buzsaki, G. (2003). Organization of cell assemblies in the hippocampus. *Nature*, 424:552–556.

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

James, W. and Stein, C. (1960). Estimation with quadratic loss. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, 1:361–379.

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.

Keat, J., Reinagel, P., Reid, R., and Meister, M. (2001). Predicting every spike: a model for the responses of visual neurons. *Neuron*, 30:803–817.

Klinger, A. (1998). *High-dimensional generalized linear models*. PhD thesis, University of Munich.

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

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.

Machens, C., Wehr, M., and Zador, A. (2003). Spectro-temporal receptive fields of subthreshold responses in auditory cortex. *NIPS*.

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.

Ng, A. (2004). Feature selection, L_1 vs. L_2 regularization, and rotational invariance. *ICML*, 21.

Okajima, K. and Imaoka, H. (2001). A Complex Cell-Like Receptive Field Obtained by Information Maximization. *Neural Computation*, 13(3):547–562.

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.

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.

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.

Rust, N., Mante, V., Simoncelli, E., and Movshon, J. (2006). How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, 11:1421–1431.

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

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

Schwartz, O., Chichilnisky, E., and Simoncelli, E. (2002). Characterizing neural gain control using spike-triggered covariance. *NIPS*, 14.

Sharpee, T., Sugihara, H., Kurgansky, A., Rebrik, S., Stryker, M., and Miller, K. (2006). Adaptive filtering enhances information transmission in visual cortex. *Nature*, 439:936–942.

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

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

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.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58:267–288.

Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

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.

Victor, J., Mechler, F., Repucci, M., Purpura, K., and Sharpee, T. (2006). Responses of V1 neurons to two-dimensional hermite functions. *Journal of Neurophysiology*, 95:379–400.

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