Published in: Bayesian Brain: Probabilistic Approaches to Neural Coding
eds. K Doya, S Ishii, A Pouget & R Rao. MIT press: 53-70 (2007).

3 Likelihood-Based Approaches to Modeling the Neural Code

Jonathan Pillow

One of the central problems in systems neuroscience is that of characterizing the functional relationship between sensory stimuli and neural spike responses. Investigators call this the *neural coding problem*, because the spike trains of neurons can be considered a code by which the brain represents information about the state of the external world. One approach to understanding this code is to build mathematical models of the mapping between stimuli and spike responses; the code can then be interpreted by using the model to predict the neural response to a stimulus, or to decode the stimulus that gave rise to a particular response. In this chapter, we will examine *likelihood-based* approaches, which use the explicit probability of response to a given stimulus for both fitting the model and assessing its validity. We will show how the likelihood can be derived for several types of neural models, and discuss theoretical considerations underlying the formulation and estimation of such models. Finally, we will discuss several ideas for evaluating model performance, including time-rescaling of spike trains and optimal decoding using Bayesian inversion of the likelihood function.

3.1 The Neural Coding Problem

Neurons exhibit stochastic variability. Even for repeated presentations of a fixed stimulus, a neuron's spike response cannot be predicted with certainty. Rather, the relationship between stimuli and neural responses is probabilistic. Understanding the neural code can therefore be framed as the problem of determining $p(y|x)$, the probability of response y conditional on a stimulus x. For a complete solution, we need to be able compute $p(y|x)$ for any x, meaning a description of the full response distribution for any stimulus we might present to a neuron. Unfortunately, we cannot hope to get very far trying to measure this distribution directly, due to the high dimensionality of stimulus space (e.g., the space of all natural images) and the finite duration of neurophysiology experiments. Figure 3.1 shows an illustration of the general problem.

A classical approach to the neural coding problem has been to restrict at-

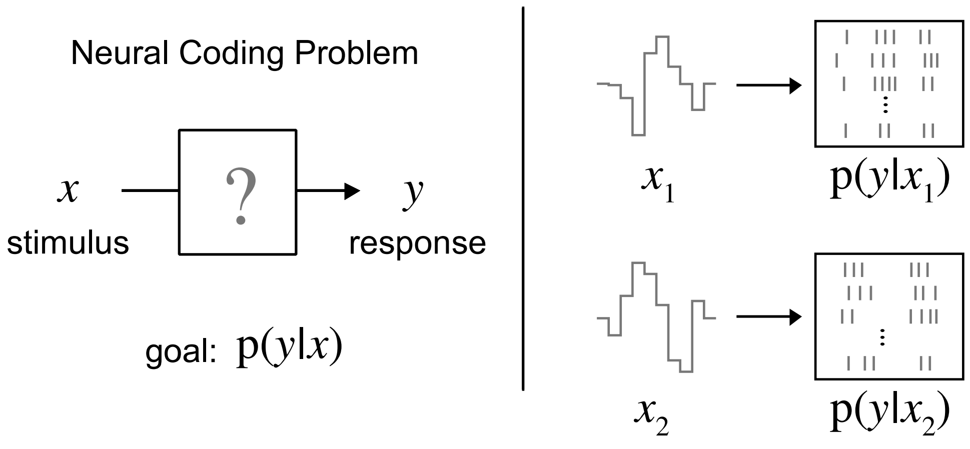

Figure 3.1 Illustration of the neural coding problem. The goal is to find a model mapping x to y that provides an accurate representation of the conditional distribution $p(y|x)$. Right: Simulated distribution of neural responses to two distinct stimuli, x_1 and x_2 illustrating (1) stochastic variability in responses to a single stimulus, and (2) that the response distribution changes as a function of x. A complete solution involves predicting $p(y|x)$ for any x.

tention to a small, parametric family of stimuli (e.g., flashed dots, moving bars, or drifting gratings). The motivation underlying this approach is the idea that neurons are sensitive only to a restricted set of *stimulus features*, and that we can predict the response to an arbitrary stimulus simply by knowing the response to these features. If $x_{\{\psi\}}$ denotes a parametric set of features to which a neuron modulates its response, then the classical approach posits that $p(y|x) \approx p(y|x_\psi)$, where x_ψ is the stimulus feature that most closely resembles x.

Although the "classical" approach to neural coding is not often explicitly framed in this way, it is not so different in principle from the "statistical modeling" approach that has gained popularity in recent years, and which we pursue here. In this framework, we assume a probabilistic model of the neural response, and attempt to fit the model parameters θ so that $p(y|x,\theta)$, the response probability under the model, provides a good approximation to $p(y|x)$. Although the statistical approach is often applied using stimuli drawn stochastically from a high-dimensional ensemble (e.g. Gaussian white noise) rather than a restricted parametric family (e.g. sine gratings), the goals are essentially similar: to find a simplified and computationally tractable description of $p(y|x)$. The statistical framework differs primarily in its emphasis on detailed quantitative prediction of spike responses, and in offering a unifying mathematical framework (likelihood) for fitting and validating models.

3.2 Model Fitting with Maximum Likelihood

Let us now turn to the problem of using likelihood for fitting a model of an individual neuron's response. Suppose we have a set of stimuli $\mathbf{x} = \{x_i\}$ and a set of spike responses $\mathbf{y} = \{y_i\}$ obtained during a neurophysiology experiment, and we would like to fit a model that captures the mapping from $\mathbf{x}$ to $\mathbf{y}$. Given a particular model, parametrized by the vector θ, we can apply a tool from classical statistics known as *maximum likelihood* to obtain an asymptotically optimal estimate of θ. For this, we need an algorithm for computing $p(\mathbf{y}|\mathbf{x}, \theta)$, which, considered as a function of θ, is called the *likelihood* of the data. The maximum likelihood (ML) estimate $\hat{\theta}$ is the set of parameters under which these data are most probable, or the maximizer of the likelihood function:

$$\hat{\theta} = \arg\max_{\theta} p(\mathbf{y}|\mathbf{x}, \theta). \tag{3.1}$$

Although this solution is easily stated, it is unfortunately the case that for many models of neural response (e.g. detailed biophysical models such as Hodgkin-Huxley) it is difficult or impossible to compute likelihood. Moreover, even when we can find simple algorithms for computing likelihood, maximizing it can be quite difficult; in most cases, θ lives in a high-dimensional space, containing tens to hundreds of parameters (e.g. describing a neuron's receptive field and spike-generation properties). Such nonlinear optimization problems are often intractable.

In the following sections, we will introduce several probabilistic neural spike models, derive the likelihood function for each model, and discuss the factors affecting ML estimation of its parameters. We will also compare ML with standard (e.g. moment-based) approaches to estimating model parameters.

3.2.1 The LNP Model

One of the best-known models of neural response is the linear-nonlinear-Poisson (LNP) model, which is alternately referred to as the linear-nonlinear "cascade" model. The model, which is schematized in the left panel of figure 3.2, consists of a linear filter (k), followed by a point nonlinearity (f), followed by Poisson spike generation. Although many interpretations are possible, a simple description of the model's components holds that:

- k represents the neuron's space-time receptive field, which describes how the stimulus is converted to intracellular voltage;

- f describes the conversion of voltage to an instantaneous spike rate, accounting for such nonlinearities as rectification and saturation;

- instantaneous rate is converted to a spike train via an inhomogeneous Poisson process.

The parameters of this model can be written as $\theta = \{k, \phi_f\}$, where ϕ_f are the parameters governing f. Although the LNP model is not biophysically realistic (especially the assumption of Poisson spiking), it provides a compact and reasonably accurate description of average responses, e.g., peri-stimulus time histogram (PSTH), in many early sensory areas.

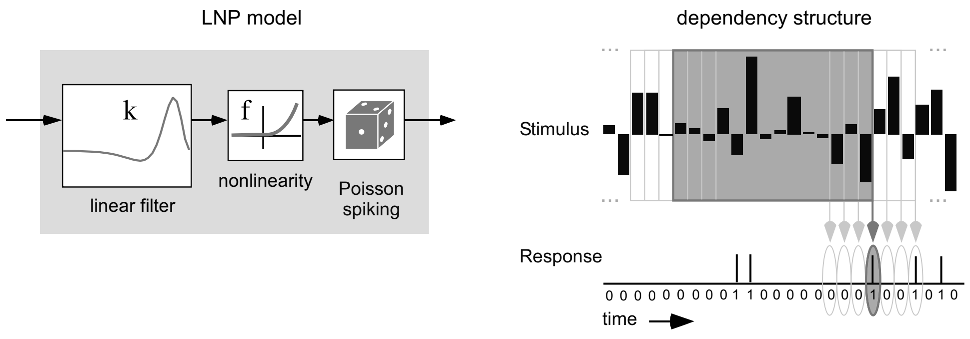

Figure 3.2 Schematic and dependency structure of the linear-nonlinear-Poisson model. Left: LNP model consists of a linear filter k, followed by a point nonlinearity f, followed by Poisson spike generation. Right: Depiction of a discretized white noise Gaussian stimulus (above) and spike response (below). Arrows indicate the causal dependency entailed by the model between portions of the stimulus and portions of the response. The highlighted gray box and gray oval show this dependence for a single time bin of the response, while gray boxes and arrows indicate the (time-shifted) dependency for neighboring bins of the response. As indicated by the diagram, all time bins of the response are conditionally independent given the stimulus (equation (3.2)).

Another reason for the popularity of the LNP model is the existence of a simple and computationally efficient fitting algorithm, which consists in using spike-triggered average (STA) as an estimate for k and a simple histogram procedure to estimate ϕ_f (see [6, 8]). It is a well-known result that the STA (or "reverse correlation") gives an unbiased estimate of the direction of k (i.e. the STA converges to αk, for some unknown α) if the raw stimulus distribution $p(x)$ is spherically symmetric, and f shifts the mean of the spike-triggered ensemble away from zero (i.e. the expected STA is not the zero vector) [7, 15]. However, the STA does *not* generally provide an optimal estimate of k, except in a special case we will examine in more detail below [16].

First, we derive the likelihood function of the LNP model. The right panel of figure 3.2 shows the dependency structure (also known as a graphical model) between stimulus and response, where arrows indicate conditional dependence. For this model, the bins of the response are conditionally independent of one another given the stimulus, an essential feature of Poisson processes, which means that the probability of the entire spike train factorizes as

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_i p(y_i|x_i, \theta), \tag{3.2}$$

where y_i is the spike count in the ith time bin, and x_i is the stimulus vector causally associated with this bin. Equation (3.2) asserts that the likelihood of the entire spike train is the product of the single-bin likelihoods. Under this model, single-bin likelihood is given by the Poisson distribution with rate parameter $\Delta f(k{\cdot}x_i)$, where $k{\cdot}x_i$, is the dot product of k with x_i and Δ is the width of the time bin. The probability of having y_i spikes in the ith bin is therefore

$$p(y_i|x_i,\theta) = \frac{1}{y_i!}\,[\Delta f(k\cdot x_i)]^{y_i}\, e^{-\Delta f(k\cdot x_i)}, \tag{3.3}$$

and the likelihood of the entire spike train can be rewritten as:

$$p(\mathbf{y}|\mathbf{x},\theta) = \Delta^n \prod_i \frac{f(k\cdot x_i)^{y_i}}{y_i!} e^{-\Delta f(k\cdot x_i)}, \tag{3.4}$$

where n is the total number of spikes.

We can find the ML estimate $\hat{\theta} = \{\hat{k}, \hat{\phi}_f\}$ by maximizing the log of the likelihood function (which is monotonically related to likelihood), and given by

$$\log p(\mathbf{y}|\mathbf{x},\theta) = \sum_i y_i \log f(k\cdot x_i) - \Delta \sum_i f(k\cdot x_i) + c, \tag{3.5}$$

where c is a constant that does not depend on k or f. Because there is an extra degree of freedom between the amplitude of k and input scaling of f, we can constrain k to be a unit vector, and consider only the angular error in estimating k. By differentiating the log-likelihood with respect to k and setting it to zero, we find that the ML estimate satisfies:

$$\lambda\hat{k} = \sum_i y_i \frac{f'(\hat{k}\cdot x_i)}{f(\hat{k}\cdot x_i)} x_i - \Delta \sum_i f'(\hat{k}\cdot x_i) x_i, \tag{3.6}$$

where λ is a Lagrange multiplier introduced to constrain k to be a unit vector. As noted in [16], the second term on the right hand converges to a vector proportional to k if the stimulus distribution $p(x)$ is spherically symmetric. (It is the expectation over $p(x)$ of a function radially symmetric around k.) If we replace this term by its expectation, we are left with just the first term, which is a weighted STA, since y_i is the spike count and x_i is the stimulus preceding the ith bin. This term is proportional to the (ordinary) STA if f'/f is constant, which occurs only when $f(z) = e^{az+b}$.

Therefore, the STA corresponds to the ML estimate for k whenever f is exponential; conversely, if f differs significantly from exponential, equation (3.6) specifies a different weighting of the spike-triggered stimuli, and the traditional STA is suboptimal. Figure 3.3 illustrates this point with a comparison between the STA and the ML estimate for k on spike trains simulated using three different nonlinearities. In the simulations, we found the ML estimate by directly maximizing log-likelihood (equation (3.5)) for both k and ϕ_f, beginning with the STA as an initial estimate for k. As expected, the ML estimate outperforms the STA except when f is exponential (rightmost column).

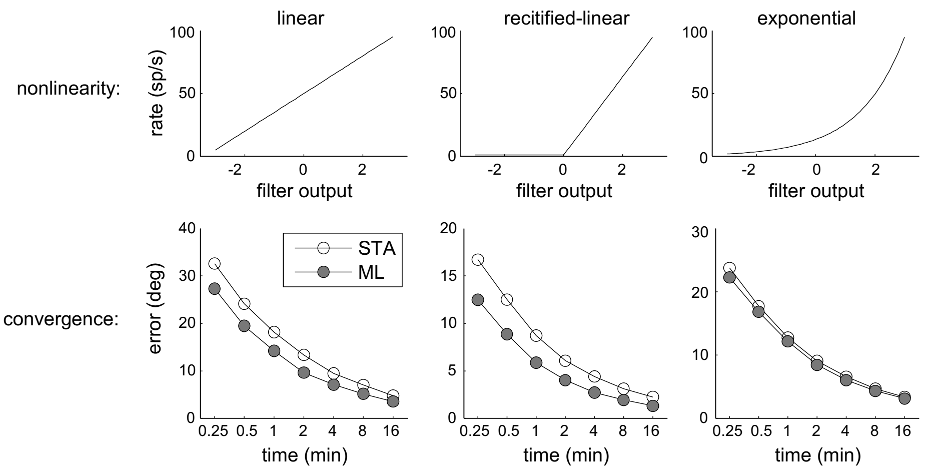

Figure 3.3 Comparison of spike-triggered average (STA) and maximum likelihood (ML) estimates of the linear filter k in an LNP model. Top row: three different types of nonlinearity f: a linear function (left), a half-wave rectified linear function (middle), and an exponential. For each model, the true k was a 20-tap temporal filter with biphasic shape similar to that found in retinal ganglion cells. The stimulus was temporal Gaussian white noise with a frame rate of 100 Hz, and k was normalized so that filter output had unit standard deviation. Bottom row: Plots show the convergence behavior for each model, as a function of the amount of data collected. Error is computed as the angle between the estimate and the true k, averaged over 100 repeats at each stimulus length.

Figure 3.4 shows a similar analysis comparing ML to an estimator derived from spike-triggered covariance (STC) analysis, which uses the principal eigenvector of the STC matrix to estimate k. Recent work has devoted much attention to fitting LNP models with STC analysis, which is relevant particularly in cases where the f is approximately symmetric [10, 22, 26, 1, 25, 3, 23]. The left column of figure 3.4 shows a simulation where f is a quadratic, shifted slightly from the origin so that both the STA and the first eigenvector of the STC provide consistent (asymptotically convergent) estimates of k. Both, however, are significantly outperformed by the ML estimator. Although it is beyond the scope of this chapter, a derivation similar to the one above shows that there is an f for which the ML estimator and the STC estimate are identical. The relevant f is a *quadratic* in the argument of an exponential, which can also be represented as a ratio of two Gaussians (see [20] for a complete discussion). The right column of figure 3.4 shows results obtained with such a nonlinearity. If we used a similar nonlinearity in which the first term of the quadratic is negative, e.g. $f(x) = \exp(-x^2)$, then f produces a reduction in variance along k, and the STC eigenvector with the *smallest* eigenvalue is comparable to the ML estimate [20].

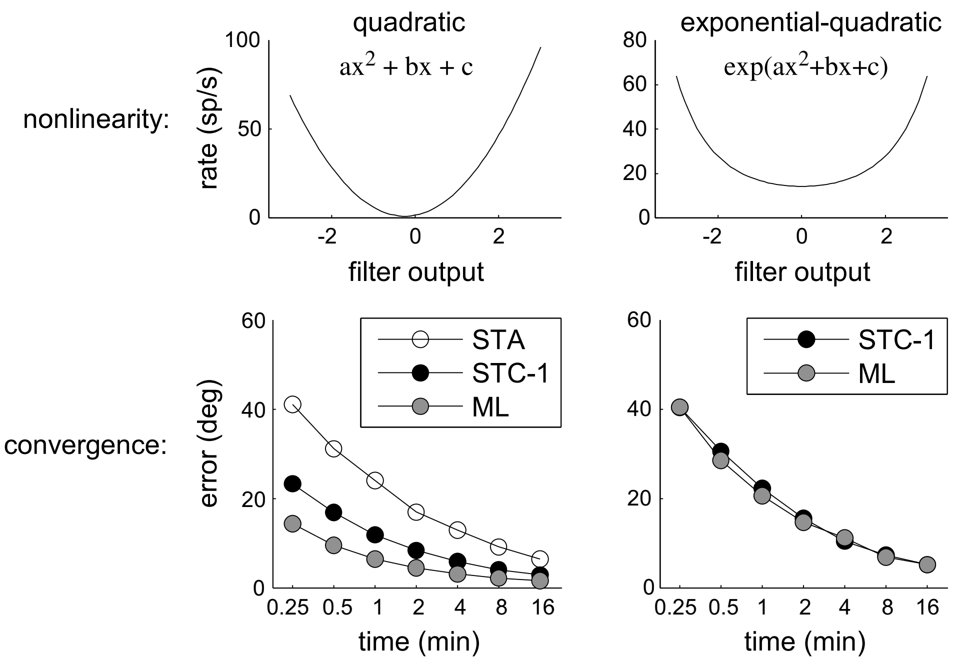

Figure 3.4 Comparison of STA, STC, and ML estimates k in an LNP model. Top row: Two types of nonlinearity functions used to generate responses; a quadratic function (left), a quadratic raised to an exponential (right). Stimulus and true k as in figure 3.3. Bottom row: Convergence behavior of the STA, first (maximum-variance) eigenvector of the STC, and ML estimate. The STA is omitted from the right plot, as it fails to converge under a symmetric nonlinearity.

Before closing this section, it is useful to review several other general characteristics of ML estimation in LNP models. Firstly, note that the LNP model can be generalized to include multiple linear filters and a multidimensional nonlinearity, all of which can be fit using ML. In this case, the likelihood function is the same as in equation (3.4), only the instantaneous spike rate is now given by:

$$\text{rate}(x_i) = f(k_1 \cdot x_i, k_2 \cdot x_i, \ldots, k_m \cdot x_i), \tag{3.7}$$

where $\{k_1, k_2, \ldots, k_m\}$ is a collection of filters and f is an m-dimensional point nonlinearity. Secondly, ML estimation of the LNP model enjoys the same statistical advantages as several information-theoretic estimators that have been derived for finding "maximally informative dimensions" or features of the stimulus space [15, 24]. Specifically, the ML estimator is unbiased even when the raw stimulus distribution lacks spherical symmetry (e.g. "naturalistic stimuli") and it is sensitive to higher-order statistics of the spike-triggered ensemble, making it somewhat more powerful and more general than STA or STC analysis. Unfortunately, ML also shares the disadvantages of these information-theoretic estimators: it is computationally intensive, difficult to use for recovering multiple (e.g. > 2) filters (in part due to the difficulty of choosing an appropriate parametrization for f), and cannot be guaranteed to converge to the true maximum using gradient ascent, due to the existence of multiple local maxima in the likelihood function.

We address this last shortcoming in the next two sections, which discuss models constructed to have likelihood functions that are free from sub-optimal local maxima. These models also introduce dependence of the response on spike-train history, eliminating a second major shortcoming of the LNP model, the assumption of Poisson spike generation.

3.2.2 Generalized Linear Model

The generalized linear model (or GLM), schematized in figure 3.5, generalizes the LNP model to incorporate feedback from the spiking process, allowing the model to account for history-dependent properties of neural spike trains such as the refractory period, adaptation, and bursting [16, 27]. As shown in the dependency diagram (right panel of figure 3.5), the responses in distinct time bins are no longer conditionally independent given the stimulus; rather, each bin of the response depends on some time window of the recent spiking activity. Luckily, this does not prevent us from factorizing the likelihood, which can now be written

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_i p(y_i|x_i, y_{[i-k\,:\,i-1]}, \theta), \tag{3.8}$$

where $y_{[i-k\,:\,i-1]}$ is the vector of recent spiking activity from time bin $i - k$ to $i - 1$. This factorization holds because, by Bayes' rule, we have

$$p(y_i, y_{[i-k\,:\,i-1]}|\mathbf{x}, \theta) = p(y_i|y_{[i-k\,:\,i-1]}, \mathbf{x}, \theta)p(y_{[i-k\,:\,i-1]}|\mathbf{x}, \theta), \tag{3.9}$$

and we can apply this formula recursively to obtain equation (3.8). (Note, however, that no such factorization is possible if we allow loopy, e.g. bidirectional, causal dependence between time bins of the response.)

Except for the addition of a linear filter, h, operating on the neuron's spike-train history, the GLM is identical to the LNP model. We could therefore call it the "recurrent LNP" model, although its output is no longer a Poisson process, due to the history-dependence induced by h. The GLM likelihood function is similar to that of the LNP model. If we let

$$r_i = f(k \cdot x_i \,+\, h \cdot y_{[i-k\,:\,i-1]}) \tag{3.10}$$

denote the instantaneous spike rate (or "conditional intensity" of the process), then the likelihood and log-likelihood, following equation (3.4) and (3.5), are

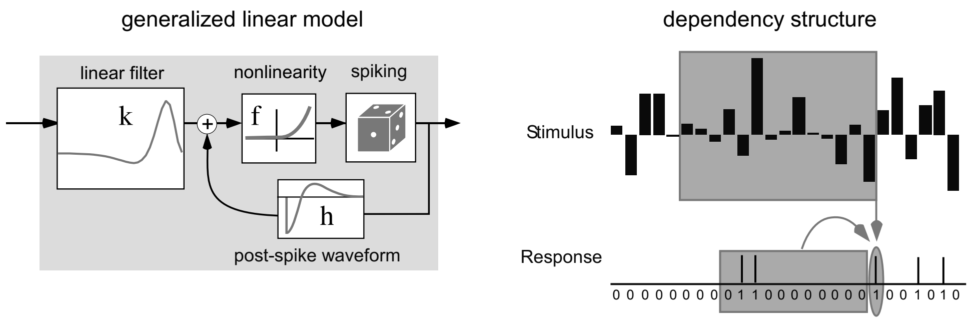

Figure 3.5 Diagram and dependency structure of a generalized linear model. Left: Model schematic, showing the introduction of history-dependence in the model via a feedback waveform from the spiking process. In order to ensure convexity of the negative log-likelihood, we now assume that the nonlinearity f is exponential. Right: Graphical model of the conditional dependencies in the GLM. The instantaneous spike rate depends on both the recent stimulus and recent history of spiking.

given by:

$$p(\mathbf{y}|\mathbf{x},\theta) = \Delta^n \prod_i \frac{r_i^{y_i}}{y_i!} e^{-\Delta r_i} \tag{3.11}$$

$$\log p(\mathbf{y}|\mathbf{x},\theta) = \sum_i y_i \log r_i - \Delta \sum_i r_i + c. \tag{3.12}$$

Unfortunately, we cannot use moment-based estimators (STA and STC) to estimate k and h for this model, because the consistency of those estimators relies on spherical symmetry of the input (or Gaussianity, for STC), which the spike-history input term $y_{[i-k\,:\,i-1]}$ fails to satisfy [15].

As mentioned above, a significant shortcoming of the ML approach to neural characterization is that it may be quite difficult in practice to find the maximum of the likelihood function. Gradient ascent fails if the likelihood function is rife with local maxima, and more robust optimization techniques (like simulated annealing) are computationally exorbitant and require delicate oversight to ensure convergence.

One solution to this problem is to constrain the model so that we guarantee that the likelihood function is free from (non-global) local maxima. If we can show that the likelihood function is *log-concave*, meaning that the negative log-likelihood function is convex, then we can be assured that the only maxima are global maxima. Moreover, the problem of computing the ML estimate $\hat{\theta}$ is reduced to a convex optimization problem, for which there are tractable algorithms even in very high-dimensional spaces.

As shown by [16], the GLM has a concave log-likelihood function if the nonlinearity f is itself convex and log-concave. These conditions are satisfied if the second-derivative of f is non-negative and the second-derivative of $\log f$

is non-positive. Although this may seem like a restrictive set of conditions—it rules out symmetric nonlinearities, for example—a number of suitable functions seem like reasonable choices for describing the conversion of intracellular voltage to instantaneous spike rate, for example:

- $f(z) = \max(z + b, 0)$
- $f(z) = e^{z+b}$
- $f(z) = \log(1 + e^{z+b})$,

where b is a single parameter that we also estimate with ML.

Thus, for appropriate choice of f, ML estimation of a GLM becomes computationally tractable. Moreover, the GLM framework is quite general, and can easily be expanded to include additional linear filters that capture dependence on spiking activity in nearby neurons, behavior of the organism, or additional external covariates of spiking activity. ML estimation of a GLM has been successfully applied to the analysis of neural spike trains in a variety of sensory, motor, and memory-related brain areas [9, 27, 14, 19].

3.2.3 Generalized Integrate-and-Fire Model

We now turn our attention to a dynamical-systems model of the neural response, for which the likelihood of a spike train is not so easily formulated in terms of a conditional intensity function (i.e. the instantaneous probability of spiking, conditional on stimulus and spike-train history). Recent work has shown that the leaky integrate-and-fire (IF) model, a canonical but simplified description of intracellular spiking dynamics, can reproduce the spiking statistics of real neurons [21, 13] and can mimic important dynamical behaviors of more complicated models like Hodgkin-Huxley [11, 12]. It is therefore natural to ask whether likelihood-based methods can be applied to models of this type.

Figure 3.6 shows a schematic diagram of the generalized IF model [16, 18], which is a close relative of the well-known spike response model [12]. The model generalizes the classical IF model so that injected current is a linear function of the stimulus and spike-train history, plus a Gaussian noise current that introduces a probability distribution over voltage trajectories. The model dynamics (written here in discrete time, for consistency) are given by

$$\frac{v_{i+1} - v_i}{\Delta} = -\frac{1}{\tau}(v_i - v_L) + (k \cdot x_i) + (h \cdot y_{[i-k\,:\,i-1]}) + \sigma \mathcal{N}_i \Delta^{-\frac{1}{2}}, \tag{3.13}$$

where v_i is the voltage at the ith time bin, which obeys the boundary condition that whenever $v_i \geq 1$, a spike occurs and v_i is reset instantaneously to zero. Δ is the width of the time bin of the simulation, and $\mathcal{N}_i$ is a standard Gaussian random variable, drawn independently on each i. The model parameters k and h are the same as in the GLM: linear filters operating on the stimulus and spike-train history (respectively), and the remaining parameters are: τ, the time constant of the membrane leak; v_L, the leak current reversal potential; and σ, the amplitude of the noise.

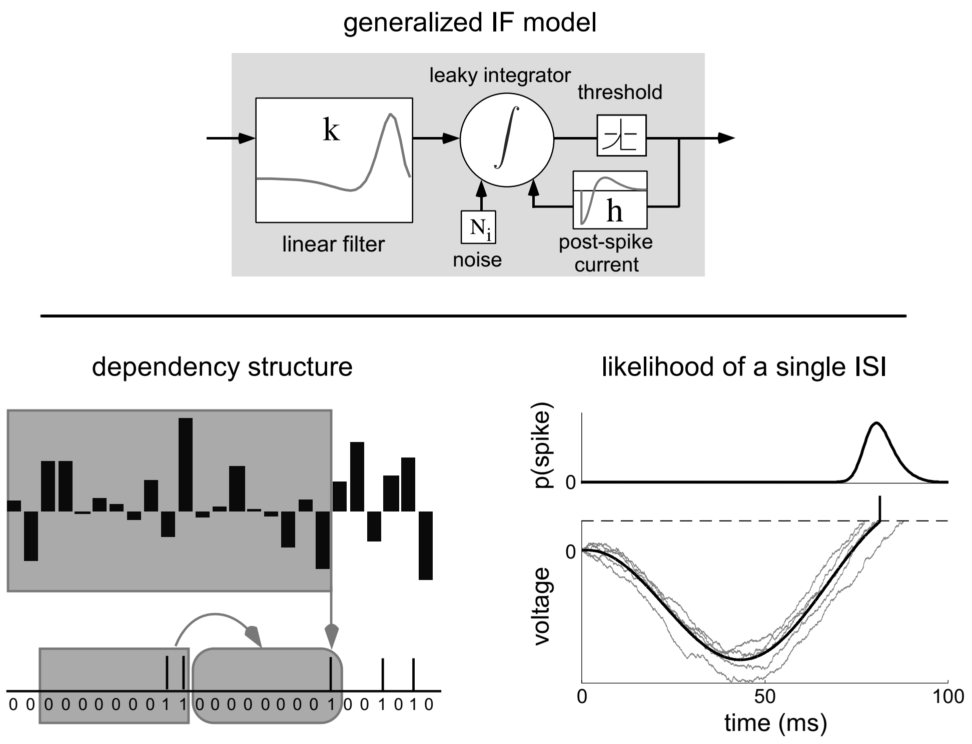

Figure 3.6 Generalized integrate-and-fire model. Top: Schematic diagram of model components, including a stimulus filter k and a post-spike current h that is injected into a leaky integrator following every spike, and independent Gaussian noise to account for response variability. Bottom left: Graphical model of dependency structure, showing that the likelihood of each interspike interval (ISI) is conditionally dependent on a portion of the stimulus and spike-train history prior to the interspike interval (ISI). Bottom right: Schematic illustrating how likelihood could be computed with Monte Carlo sampling. Black trace shows the voltage (and spike time) from simulating the model without noise, while gray traces show sample voltage paths (to the first spike time) with noise. The likelihood of the ISI is shown above, as a function of the spike time (black trace). Likelihood of an ISI is equal to the fraction of voltage paths crossing threshold at the true spike time.

The lower left panel of figure 3.6 depicts the dependency structure of the model as it pertains to computing the likelihood of a spike train. In this case, we can regard the probability of an entire interspike interval (ISI) as depending on a relevant portion of the stimulus and spike-train history. The lower right panel shows an illustration of how we might compute this likelihood for a single ISI under the generalized GIF model using Monte Carlo sampling. Computing the likelihood in this case is also known as the "first-passage time" problem. Given a setting of the model parameters, we can sample voltage trajectories from the model, drawing independent noise samples for each trajectory, and following each trajectory until it hits threshold. The gray traces show show five such sample paths, while the black trace shows the voltage path obtained in the absence of noise. The probability of a spike occurring at the ith bin is simply the fraction of voltage paths crossing threshold at this bin. The black trace (above) shows the probability distribution obtained by collecting the first passage times of a large number of paths. Evaluated at the actual spike, this density gives the likelihood of the relevant ISI. Because of voltage reset following a spike, all ISIs are conditionally independent, and we can again write the likelihood function as a product of conditionally independent terms:

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t_j} p(y_{[t_{j-1}+1\,:\,t_j]}|\mathbf{x}, y_{[0\,:\,t_j]}, \theta), \tag{3.14}$$

where $\{t_j\}$ is the set of spike times emitted by the neuron, $y_{[t_{j-1}+1\,:\,t_j]}$ is the response in the set of time bins in the jth ISI, and $y_{[0\,:\,t_j]}$ is the response during time bins previous to that interval.

The Monte Carlo approach to computing likelihood of a spike train can in principle be performed for any probabilistic dynamical-systems style model. In practice, however, such an approach would be unbearably slow and would likely prove intractable, particularly since the likelihood function must be computed many times in order find the ML estimate for θ. However, for the generalized IF model there exists a much more computationally efficient method for computing the likelihood function using the Fokker-Planck equation. Although beyond the scope of this chapter, the method works by "density propagation" of a numerical representation of the probability density over subthreshold voltage, which can be quickly computed using sparse matrix methods. More importantly, a recent result shows that the log-likelihood function for the generalized IF model, like that of the GLM, is concave. This means that the likelihood function contains a unique global maximum, and that gradient ascent can be used to find the ML estimate of the model parameters (see [17] for a more thorough discussion). Recent work has applied the generalized IF model to the responses of macaque retinal ganglion cells using ML, showing that it can be used to capture stimulus dependence, spike-history dependence, and noise statistics of neural responses recorded *in vitro* [18].

3.3 Model Validation

Once we have a used maximum likelihood to fit a particular model to a set of neural data, there remains the important task of validating the quality of the model fit. In this section, we discuss three simple methods for assessing the goodness-of-fit of a probabilistic model using the same statistical framework that motivated our approach to fitting.

3.3.1 Likelihood-Based Cross-Validation

Recall that the basic goal of our approach is to find a probabilistic model such that we can approximate the true probabilistic relationship between stimulus and response, $p(y|x)$, by the model-dependent $p(y|x,\theta)$. Once we have fit θ using a set of training data, how can we tell if the model provides a good description of $p(y|x)$? To begin with, let us suppose that we have two competing models, p_A and p_B, parametrized by θ_A and θ_B, respectively, and we wish to decide which model provides a better description of the data. Unfortunately, we cannot simply compare the likelihood of the data under the two models, $p_A(\mathbf{y}|\mathbf{x},\theta_A)$ vs. $p_B(\mathbf{y}|\mathbf{x},\theta_B)$, due to the problem of *overfitting*. Even though one model assigns the fitted data a higher likelihood than the other, it may not generalize as well to new data.

As a toy example of the phenomenon of overfitting, consider a data set consisting of 5 points drawn from a Gaussian distribution. Let model A be a single Gaussian, fit with the mean and standard deviation of the sample points (i.e. the ML estimate for this model). For model B, suppose that the data come from a mixture of five *very* narrow Gaussians, and fit this model by centering one of these narrow Gaussians at each of the 5 sample points. Clearly, the second model assigns higher likelihood to the data (because it concentrates all probability mass near the sample points), but it fails to generalize–it will assign very low probability to new data points drawn from the true distribution which do not happen to lie very near the five original samples.

This suggests a general solution to the problem of comparing models, which goes by the name *cross-validation*. Under this procedure, we generate a new set of "test" stimuli, $\mathbf{x}^*$ and present them to the neuron to obtain a new set of spike responses $\mathbf{y}^*$. (Alternatively, we could set aside a small portion of the data at the beginning.) By comparing the likelihood of these new data sets under the two models, $p_A(\mathbf{y}^*|\mathbf{x}^*,\theta_A)$ vs. $p_B(\mathbf{y}^*|\mathbf{x}^*,\theta_B)$, we get a fair test of the models' generalization performance. Note that, under this comparison, we do not actually care about the number of parameters in the two models: increasing the number of parameters in a model does *not* improve its ability to generalize. (In the toy example above, model B has more parameters but generalizes much more poorly. We can view techniques like regularization as methods for reducing the effective number of parameters in a model so that overfitting does not occur.) Although we may prefer a model with fewer parameters for aesthetic or computational reasons, from a statistical standpoint we care only about which model provides a better account of the novel data.

3.3.2 Time-Rescaling

Another powerful idea for testing validity of a probabilistic model is to use the model to convert spike times into a series of i.i.d. random variables. This conversion will only be successful if we have accurately modeled the probability distribution of each spike time. This idea, which goes under the name *time-rescaling* [5], is a specific application of the general result that we can convert any random variable into a uniform random variable using its cumulative density function (CDF).

First, let us derive the CDF of a spike time under the LNP and GLM models. If r_i is the conditional intensity function of the ith time bin (i.e. $f(k \cdot x_i)$ under the LNP model), then the probability that the "next" spike t_{j+1} occurs on or before bin k, given that the previous spike occurred at t_j, is simply 1 minus the probability that no spikes occur during the time bins $t_j + 1$ to k. This gives

$$p(t_{j+1} \leq k | t_j) = 1 - \left(\prod_{i \in [t_j+1, k]} e^{-\Delta r_i} \right) \tag{3.15}$$

which we can rewrite:

$$p(t_{j+1} \leq k | t_j) = 1 - \exp\left(-\Delta \sum_{t_j+1}^{k} r_i \right) \tag{3.16}$$

Note that the argument of the exponential is simply the negative integral of the intensity function since the time of the previous spike.

For the generalized IF model, computing the likelihood function involves computing the probability density function (PDF) over each interspike interval (as depicted in figure 3.6), which we can simply integrate to obtain the CDF [17].

Given the CDF for a random variable, a general result from probability theory holds that it provides a remapping of that variable to the one randomly distributed unit interval $[0, 1]$. Even though the CDF for each spike time is different, if we remap the entire spike train using $t_j \longrightarrow CDF_j(t_j)$, where CDF_j is the cumulative density of the jth spike time, then, if the model is correct, we should obtain a series of independent, uniform random variables. This suggests we test the validity of the model by testing the remapped spike times for independence; any correlation (or some other form of dependence) between successive pairs of remapped spike times (for example), indicates a failure of the model. We can also examine the marginal distribution of the remapped times (using a K-S test, for example) to detect deviations from uniformity. The structure of any deviations may be useful for understanding the model's failure modes: an excess of small-valued samples, for example, indicates that the model predicts too few short interspike intervals. If we wish to compare multiple models, we can use time-rescaling to examine which model produces the most nearly independent and most nearly uniform remapped spike times.

3.3.3 Model-Based Decoding

A third tool for assessing the validity of a probabilistic model is to perform stimulus decoding using the model-based likelihood function. Given the fitted model parameters, we can derive the *posterior* probability distribution over the stimulus given a spike train by inverting the likelihood function with Bayes' rule:

$$p(\mathbf{x}|\mathbf{y}, \theta) = \frac{p(\mathbf{y}|\mathbf{x}, \theta)p(\mathbf{x})}{p(\mathbf{y}|\theta)}, \tag{3.17}$$

where $p(x)$ is the prior probability of the stimulus (which we assume to be independent of θ), and the denominator is the probability of response $\mathbf{y}$ given θ. We can obtain the most likely stimulus to have generated the response $\mathbf{y}$ by maximizing the posterior for $\mathbf{x}$, which gives the maximum a posteriori (MAP) estimate of the stimulus, which we can denote

$$\hat{\mathbf{x}}_{MAP} = \arg\max_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}, \theta)p(\mathbf{x}) \tag{3.18}$$

since the denominator term $p(\mathbf{y}|\theta)$ does not vary with $\mathbf{x}$.

For the GLM and generalized IF models, the concavity of the log-likelihood function with respect to the model parameters also extends to the posterior with respect to the stimulus, since the stimulus interacts linearly with model parameters k. Concavity of the log-posterior holds so long as the prior $p(\mathbf{x})$ is itself log-concave (e.g. Gaussian, or any distribution of the form $\alpha e^{-(x/\sigma)^{\gamma}}$, with $\gamma \geq 1$). This means that, for both of these two models, we can perform MAP decoding of the stimulus using simple gradient ascent of the posterior.

If we wish to perform decoding with a specified loss function, for example, mean-squared error, optimal decoding can be achieved with Bayesian estimation, which is given by the estimator with minimum expected loss. In the case of mean-squared error, this estimator is given by

$$\hat{\mathbf{x}}_{Bayes} = E[\mathbf{x}|\mathbf{y}, \theta], \tag{3.19}$$

which is the conditional expectation of $\mathbf{x}$, or the mean of the posterior distribution over stimuli. Computing this estimate, however, requires sampling from the posterior distribution, which is difficult to perform without advanced statistical sampling techniques, and is a topic of ongoing research.

Considered more generally, decoding provides an important test of model validity, and it allows us to ask different questions about the nature of the neural code. Even though it may not be a task carried out explicitly in the brain, decoding allows us to measure how well a particular model preserves the stimulus-related information in the neural response. This is a subtle point, but one worth considering: we can imagine a model that performs worse under cross-validation or time-rescaling analyses, but performs better at decoding, and therefore gives a better account of the stimulus-related information that is conveyed to the brain. For example, consider a model that fails to account for the refractory period (e.g. an LNP model), but which gives a slightly better description of the stimulus-related probability of spiking. This model assigns non-zero probability to spike trains that violate the refractory period, thereby "wasting" probability mass on spike trains whose probability is actually zero, and performing poorly under cross-validation. The model also performs poorly under time-rescaling, due to the fact that it over-predicts spike rate during the refractory period. However, when decoding a *real* spike train, we do not encounter violations of the refractory period, and the "wasted" probability mass affects only the normalizing term $p(\mathbf{y}|\theta)$. Here, the model's improved accuracy in predicting the stimulus-related spiking activity leads to a posterior that is more reliably centered around the true stimulus. Thus, even though the model fails to reproduce certain statistical features of the response, it provides a valuable tool for assessing what information the spike train carries about the stimulus, and gives a perhaps more valuable description of the neural code. Decoding may therefore serve as an important tool for validating likelihood-based models, and a variety of exact or approximate likelihood-based techniques for neural decoding have been explored [28, 4, 2, 18].

3.4 Summary

We have shown how to compute likelihood and perform ML fitting of several types of probabilistic neural models. In simulations, we have shown that ML outperforms traditional moment-based estimators (STA and STC) when the nonlinear function of filter output does not have a particular exponential form. We have also discussed models whose log-likelihood functions are provably concave, making ML estimation possible even in high-dimensional parameter spaces and with non-Gaussian stimuli. These models can also be extended to incorporate dependence on spike-train history and external covariates of the neural response, such as spiking activity in nearby neurons. We have examined several statistical approaches to validating the performance of a neural model, which allow us to decide which models to use and to assess how well they describe the neural code.

In addition to the insight they provide into the neural code, the models we have described may be useful for simulating realistic input to downstream brain regions, and in practical applications such as neural prosthetics. The theoretical and statistical tools that we have described here, as well as the vast computational resources that make them possible, are still a quite recent development in the history of theoretical neuroscience. Understandably, their achievements are still quite modest: we are some ways from a "complete" model that predicts responses to *any* stimulus (e.g., incorporating the effects of spatial and multi-scale temporal adaptation, network interactions, and feedback). There remains much work to be done both in building more powerful and accurate models of neural responses, and in extending these models (perhaps in cascades) to the responses of neurons in brain areas more deeply removed from the periphery.

References

[1] Aguera y Arcas B, Fairhall AL (2003) What causes a neuron to spike? *Neural Computation*, 15(8):1789–1807.

[2] Barbieri R, Frank L, Nguyen D, Quirk M, Solo V, Wilson M, Brown E (2004) Dynamic Analyses of Information Encoding in Neural Ensembles. *Neural Computation*, 16:277–307.

[3] Bialek W, de Ruyter van Steveninck R (2005) Features and dimensions: motion estimation in fly vision. *Quantitative Biology, Neurons and Cognition*, arXiv:q-bio.NC/0505003.

[4] Brown E, Frank L, Tang D, Quirk M, 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.

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

[6] Bryant H, Segundo J (1976) Spike initiation by transmembrane current: a white-noise analysis. *Journal of Physiology*, 260:279–314.

[7] Bussgang J (1952) Crosscorrelation functions of amplitude-distorted gaussian signals. *RLE Technical Reports*, 216.

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

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

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

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

[12] Jolivet R, Lewis T, Gerstner W (2003) The spike response model: a framework to predict neuronal spike trains. *Springer Lecture Notes in Computer Science*, 2714:846–853.

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

[14] Okatan M, Wilson M, Brown E (2005) Analyzing Functional Connectivity Using a Network Likelihood Model of Ensemble Neural Spiking Activity. *Neural Computation*, 17:1927–1961.

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

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

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

[18] Pillow JW, Paninski L, Uzzell VJ, Simoncelli EP, Chichilnisky EJ (2005) Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25:11003–11013.

[19] Pillow JW, Shlens J, Paninski L, Chichilnisky EJ, Simoncelli EP (2005) Modeling the correlated spike responses of a cluster of primate retinal ganglion cells. *SFN Abstracts*, 591.3.

[20] Pillow JW, Simoncelli EP (2006) Dimensionality reduction in neural models: an information-theoretic generalization of spike-triggered average and covariance analysis. *Journal of Vision*, 6(4):414–428.

[21] Reich DS, Victor JD, Knight BW (1998) The power ratio and the interval map: spiking models and extracellular recordings. *Journal of Neuroscience*, 18:10090–10104.

[22] Schwartz O, Chichilnisky EJ, Simoncelli EP (2002) Characterizing neural gain control using spike-triggered covariance. In T G Dietterich, S Becker, and Z Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA: MIT Press.

[23] Schwartz O, Pillow JW, Rust NC, Simoncelli EP (2006) Spike-triggered neural characterization. *Journal of Vision*, 6(4):484–507.

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

[25] Simoncelli E, Paninski L, Pillow J, Schwartz O (2004) Characterization of neural responses with stochastic stimuli. In M. Gazzaniga, editor, *The Cognitive Neurosciences*, 3rd edition, Cambridge, MA: MIT Press.

[26] Touryan J, Lau B, Dan Y (2002) Isolation of relevant visual features from random stimuli for cortical complex cells. *Journal of Neuroscience*, 22:10811–10818.

[27] Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN (2004) A point process framework for relating neural spiking activity to spiking history, neural ensemble and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089.

[28] Warland D, Reinagel P, Meister M (1997) Decoding visual information from a population retinal ganglion cells. *Journal of Neurophysiology*, 78:2336–2350.

Index