Chapter 6 Doubly Stochastic Poisson Models

6.1 Introduction

In this chapter we turn from the study of models of spike waveforms, to models of the arrival times of the action potentials invoked in response to an experimental stimulus. The work described here was carried out jointly with J. Linden. The methods that will be discussed have been applied to data[1] collected from the lateral intraparietal area in two macaques during fixation and saccade tasks involving visual and auditory targets. A detailed discussion of this application is presented by Linden (1999).

6.1.1 Point processes

In chapter 5 we examined a variety of statistical models that described the spike waveforms recorded by extracellular electrodes. While the shape of the waveform provided us with information about the identity of the neuron in which the associated action potential occurred, it is not actually used by the nervous system to transmit information between neurons. Instead, from the point of view of the neuron, the action potential is an all-or-nothing pulse: any information that needs to be relayed between cells is carried in the occurrence and timing of the pulses alone.

Statistically, we may view a train of action potentials or spikes[2] from a single neuron as the outcome of a **stochastic point process**. The theory of such processes has been studied extensively in the statistics literature (Cox and Lewis 1966; Cox and Isham 1980; Snyder and Miller 1991). The outcome of a point process may be represented in one of two ways: either as a sequence of N event times $\{\tau_i : i = 1 \dots N\}$ or as a sequence of T counts $\{x_t : t = 1 \dots T\}$. The count x_t indicates the number of events that fall within the small interval $[t\delta, (t+1)\delta)$; thus $\sum_t x_t = N$ and $0 \leq t_i < T\delta$. We will always take the intervals to be of the same length, given by the **bin width**, δ. In this chapter we will be concerned solely with the counting representation. It will frequently be useful to collect the counts x_t into the vector, $\mathbf{x}$.

A prominent distribution, that plays a rôle in point-process theory quite similar to that of the Gaussian in continuous random variable theory, is the **Poisson process**. In particular, this is the maximum entropy distribution for a given density of events. Under the Poisson distribution for a counting process each of the counting random variables is independent. A single parameter, ρ_t, the

[1] The data were collected by J. Linden and Dr. A. Grunewald, in Dr. R. A. Andersen's laboratory.
[2] For the purposes of this chapter we need not distinguish between the two.

mean or **rate** of the process, characterizes the distribution of the variable x_t

$$\mathsf{P}_{\rho_t}(x_t) = \frac{e^{-\rho_t}\rho_t^{x_t}}{x_t!} \tag{6.1}$$

Thus the probability of the count vector $\mathbf{x}$, given a rate vector $\boldsymbol{\rho}$ is

$$\mathsf{P}_{\boldsymbol{\rho}}(\mathbf{x}) = \prod_{t=1}^{T} \frac{e^{-\rho_t}\rho_t^{x_t}}{x_t!} \tag{6.2}$$

If ρ_t is the same for each interval the Poisson process is called **homogeneous**. In this chapter we will be primarily concerned with **inhomogeneous** processes.

6.1.2 Spike response variability

Many neurophysiological experiments are conducted as follows. A stimulus is presented to an animal subject and the times of action potentials in one or more neurons in the subject's brain are recorded. The stimulus may well elicit some trained behaviour from the animal: action potentials are recorded for the entire duration of experimental interest around both the stimulus presentation and behavioural event, if any. The same stimulus (and, presumably, behaviour) is then repeated over many different experimental trials, often randomly interleaved with other, similar, stimuli. On each repetition, the times of the action potentials that arise in the same neurons are noted. The result is a database of stimulus-response pairs for each cell.

The neurons of interest in a given experiment usually alter their patterns of firing during the trial, in a manner linked to the presentation of the stimulus or to the execution of the behaviour (or both). Such neurons appear to be related to the processing of either the stimulus or the behavioural response. However, very rarely does a neuron respond to multiple trials in an exactly repeatable manner; this is particularly true of cells in the cerebral cortex of mammals, such as those to be modeled here. This variability in the response of a neuron is what leads us to treat the pattern of spikes as the output of a stochastic process.

Spike trains observed in response to the same stimulus have often been modeled as independently drawn from a single inhomogeneous Poisson process (Perkel *et al*. 1967). In detail such a model must be wrong. Both the refractory period and the presence of bursts violate the independence assumption of the Poisson counting process. However, in situations where the counting intervals are sufficiently large, it has been thought to be a reasonable approximation.

Poisson processes, including those with inhomogeneous rate, have the property that the distribution of counts retains the form (6.1) whatever the choice of the counting interval. In particular, we might select the interval $[0, T)$, to obtain the total spike count during a trial. Provided the original process is Poisson, this count will still be distributed according to (6.1). That distribution has the

property that its variance is equal to its mean.

In practice, the variance in spike count from across repeated, experimentally identical, trials is often larger than can be accounted for by the simple Poisson model (Tolhurst *et al.* 1981; Dean 1981; Tolhurst *et al.* 1983; Vogels *et al.* 1989; Softky and Koch 1993; Gershon *et al.* 1998; Shadlen and Newsome 1998). This same result is apparent in the data to be modeled here (Linden 1999), where the ratio between variance and mean (known as the **Fano factor**) appears to be closer to 1.5 than to 1. One possible source of this additional variance across trials might be slow changes in the overall excitability of neurons or of the cortical area. A number of recent reports have provided direct or indirect evidence for this idea (Brody 1998; Oram *et al.* 1998; also see Tomko and Crapper 1974; Rose *et al.* 1990; Tolhurst *et al.* 1981; Arieli *et al.* 1996). Such slow variation in neuronal excitability might result in an apparently stochastic scaling of the underlying inhomogeneous Poisson rate. This hypothesis will form the basis of the model to be discussed here.

6.2 The Generative Model

The generative model for a spike train $\mathbf{x}$, output by a given cell in response to given experimental conditions, is as follows. The cell-stimulus pair is taken to specify a non-negative **intensity profile**, $\boldsymbol{\lambda}$, that describes the time-course of the cell's response to the stimulus. This profile is scaled by a latent variable, s, which is drawn from a gamma distribution with unit mean, and which is meant to represent the excitability of the neuron on a given trial. The action potential times are then generated by an inhomogeneous Poisson process with rate vector $\boldsymbol{\rho} = s\boldsymbol{\lambda}$.

This model is known in the point process literature as an inhomogeneous Polya process (see Snyder and Miller 1991). It is a special case of the **doubly stochastic Poisson process**: "doubly stochastic" because the Poisson rate is itself a random variable (Cox 1955; Snyder and Miller 1991). Clearly, any such process is a latent variable model. Other examples of doubly stochastic Poisson processes have also been used to model neural spike data by other investigators; for example, some authors have taken the rate to be a piecewise constant function generated from a Markov chain (Radons *et al.* 1994; Abeles *et al.* 1993; Seidemann *et al.* 1996; Gat *et al.* 1997). The present choice is, in part, appealing for its simplicity and relative tractability. As can be seen from the applications discussed by Linden (1999), it can produce useful results.

The standard form of the gamma density (for the scale s) depends on two parameters α and β. It is given by

$$\mathsf{P}_{\alpha,\beta}(s) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}} s^{\alpha-1} e^{-s/\beta} \tag{6.3}$$

It may be easily verified that the mean of this distribution is $\alpha\beta$. Thus, our requirement that the distribution have unit mean constrains the parameters such that $\beta = 1/\alpha$, and we obtain instead

the single parameter density

$$\mathsf{P}_\alpha(s) = \frac{\alpha^\alpha}{\Gamma(\alpha)} s^{\alpha-1} e^{-s\alpha} \tag{6.4}$$

We will refer to the parameter α as the **stability**, since as it grows the variability in spike count drops.

Combining this with the expression for the inhomogeneous Poisson process probability (6.2), we obtain the joint density of a spike train $\mathbf{x}$ being observed along with a scale factor s.

$$\mathsf{P}_{\boldsymbol{\lambda},\alpha}(\mathbf{x}, s) = \left(\prod_{t=1}^{T} \frac{e^{-s\lambda_t}(s\lambda_t)^{x_t}}{x_t!}\right)\left(\frac{\alpha^\alpha}{\Gamma(\alpha)} s^{\alpha-1} e^{-s\alpha}\right) \tag{6.5}$$

The scale, s, is not directly observable, making this a latent variable model. While we may approach learning in this model by the EM algorithm that we have used before, in this case it proves to be useful to obtain a closed form for the marginal distribution function of $\mathbf{x}$, by integrating the joint density of (6.5) with respect to s. The resultant marginal is

$$\mathsf{P}_{\boldsymbol{\lambda},\alpha}(\mathbf{x}) = \left(\prod_{t=1}^{T} \frac{\lambda_t^{x_t}}{x_t!}\right)\left(\frac{\Gamma(X+\alpha)}{\Gamma(\alpha)}\right)\alpha^\alpha(\Lambda+\alpha)^{-(X+\alpha)} \tag{6.6}$$

Here, Λ and X are the sums of the elements in the corresponding vectors: $\Lambda = \sum_{t=1}^{T}\lambda_t$ and $X = \sum_{t=1}^{T} x_t$.

We assume that a set of spike trains, $\mathcal{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_N\}$, collected from the same cell under identical trial conditions, is obtained by drawing each one independently from this distribution. We use the subscript n to identify the spike train and write X_n for the corresponding total spike count. Thus, we obtain the log-likelihood of the parameters $\boldsymbol{\lambda}$ and α under the set of observations $\mathcal{X}$,

$$\ell_{\mathcal{X}}(\boldsymbol{\lambda}, \alpha) = \log Z + \sum_{n=1}^{N}\left(\sum_{t=1}^{T} x_{nt}\log\lambda_t + \log\left(\frac{\Gamma(X_n+\alpha)}{\Gamma(\alpha)}\right) + \alpha\log\alpha - (X_n+\alpha)\log(\Lambda+\alpha)\right) \tag{6.7}$$

where the normalizing constant Z absorbs terms independent of the parameters.

As it stands, this model has a large number of independent degrees of freedom in its parameters. In particular, for small counting intervals and reasonable experimental durations, the vector $\boldsymbol{\lambda}$ may have hundreds of elements. It is impractical to expect reasonable parameter estimates from the small amounts of data that can usually be collected. Therefore, we impose a prior density on the parameters. The prior introduces inter-dependencies between the elements of $\boldsymbol{\lambda}$, reducing the effective number of degrees of freedom.

The stability parameter, α is taken to be independent of the intensity function and is distributed according to the density $e^{-1/\alpha}$. As a result, small values of α are subject to a slight penalty. In practice, this prior is vague enough to have little effect on the parameter estimates and is included

only for completeness.

The prior distribution of the intensity function is a stationary Gaussian process with zero mean and covariance matrix C. The stationarity indicates that we have no prior belief about the course of the intensity function during the experiment. In mathematical terms, it requires that the matrix C be Töplitz (that is, diagonally striped).

The resultant log posterior can be written:

$$\log \mathsf{P}\left(\boldsymbol{\lambda}, \alpha \mid \mathbf{x}_1, \ldots, \mathbf{x}_N\right) = \log Z - \frac{1}{2}\boldsymbol{\lambda}^{\mathsf{T}}\mathsf{C}^{-1}\boldsymbol{\lambda} - \frac{1}{\alpha} + \sum_{n=1}^{N}\left(\mathbf{x}_n^{\mathsf{T}}\log\boldsymbol{\lambda} - (X_n+\alpha)\log(\Lambda+\alpha) + \alpha\log\alpha + \log\left(\frac{\Gamma(X_n+\alpha)}{\Gamma(\alpha)}\right)\right) \tag{6.8}$$

where Z has now absorbed, in addition, the normalization term of the Gaussian.

The reduction in degrees of freedom is achieved by choice of a suitable prior. We select a matrix which is based on an auto-covariance function that is Gaussian[3] in shape: that is, the covariance between two elements of the intensity vector λ_s and λ_t under the prior is of the form

$$\mathsf{C}_{st} = \exp\left(-\frac{(s-t)^2}{2\Delta^2}\right) \tag{6.9}$$

The quantity Δ, which is chosen *a priori*, reflects the expected time-scale of changes in the intensity function, expressed in terms of the counting interval length δ. Thus, this choice of prior covariance expresses a belief in the smoothness of the underlying intensity function.

If Δ is fairly large, the matrix C will be ill-conditioned. As such, the inverse that appears in (6.8) creates a numerical instability. This can be resolved by diagonalizing the covariance matrix. Recall that the eigenvectors of any Töplitz matrix are the basis vectors of the discrete Fourier transform (DFT), and so C is diagonalized by the DFT matrix $\mathsf{F}^*_{st} = \frac{1}{\sqrt{T}}\exp(-2\pi i(s-1)(t-1)/T)$. Rather than use this complex form, it will be convenient to introduce a real transform matrix which separates the real and imaginary parts. Such a matrix is given by

$$\hat{\mathsf{F}}_{st} = \frac{1}{\sqrt{T}} \times \begin{cases} 1 & \text{if} \quad s = 1 \\ \cos(2\pi\frac{s}{2}\frac{(t-1)}{T}) & \text{if} \quad s > 1 \text{ and is even} \\ \sin(2\pi\frac{(s-1)}{2}\frac{(t-1)}{T}) & \text{if} \quad s > 1 \text{ and is odd} \end{cases} \tag{6.10}$$

We have assumed that T, the total number of counting intervals, is even.

Thus, the matrix $\hat{\mathsf{F}}\mathsf{C}\hat{\mathsf{F}}^{\mathsf{T}}$ is diagonal, representing the independence of the Fourier components of a stationary process. The ill-conditioning now reveals itself in the presence of one or more diagonal elements that are very close to zero. Thus, in the frequency domain, the ill-conditioning of C is

[3] It is important to distinguish between the Gaussian *distribution* of the prior and the Gaussian *shape* of the auto-covariance. One does not imply the other.

easy to interpret; it reflects the fact that in certain frequencies very little power is expected under the prior. In effect, the prior imposes a band-limitation on the intensity function. The particular choice of Gaussian auto-covariance function, for example, leads to a half-Gaussian shaped fall-off in expected power as frequency increases from 0, with the highest frequencies effectively excluded. It is important to realize, however, that the imposition of this prior is not equivalent to simply filtering the intensity function by the expected frequency profile.

We now restrict the transform matrix to a rectangular form F in which rows corresponding to the eigenvalues of C that fall below some low threshold have been eliminated. Thus the matrix FCF^T is also diagonal, but is of order less than T and is well-conditioned. We will also apply this restricted transform to the intensity function. In doing so, we force the power of the intensity function to zero at those frequencies at which the expected power is vanishingly small.

We proceed to rewrite the posterior (6.8) in terms of this transformed intensity function. In practice, it proves to be useful to represent the intensity function by the transformed logarithm $\boldsymbol{\phi} = \mathsf{F} \log \boldsymbol{\lambda}$ (where the logarithm is taken to apply element by element). The introduction of the logarithm enforces the requirement that the intensity be positive; this would otherwise be difficult to ensure when working in the frequency domain. The log-posterior now becomes

$$\begin{aligned} \log \mathsf{P}\left(\boldsymbol{\phi}, \alpha \mid \mathbf{x}_1, \ldots, \mathbf{x}_N\right) = \log Z - \frac{1}{2} e^{\boldsymbol{\phi}^\mathsf{T}\mathsf{F}} \mathsf{R} e^{\mathsf{F}^\mathsf{T}\boldsymbol{\phi}} - \frac{1}{\alpha} + \langle \mathbf{x} \rangle^\mathsf{T} \mathsf{F}^\mathsf{T} \boldsymbol{\phi} \\ - (\langle \mathbf{x} \rangle^\mathsf{T} \mathbf{1} + N\alpha) \log(e^{\boldsymbol{\phi}^\mathsf{T}\mathsf{F}} \mathbf{1} + \alpha) + N\alpha \log \alpha + \sum_{n=1}^{N} \log \left(\frac{\Gamma(X_n + \alpha)}{\Gamma(\alpha)} \right) \end{aligned} \tag{6.11}$$

where $\langle \mathbf{x} \rangle$ represents the sum of the different observations, $\mathbf{1}$ is a vector of T ones introduced to indicate summation of elements, and $\mathsf{R} = \mathsf{F}^\mathsf{T}(\mathsf{FCF}^\mathsf{T})^{-1}\mathsf{F}$. Exponentiation of a vector term is taken to apply element by element.

6.3 Optimization

We have presented a latent variable model for spike generation. In principle, we might employ the EM algorithm to find the maximum-likelihood — or, given the prior, maximum *a posteriori* — parameter estimates, as we have done with the other latent variable models discussed in this dissertation. Inspection of the joint probability (6.5), however, suggests that this may not be as easy as in our earlier examples. The latent variable, s, will enter into the joint log-likelihood in the logarithm. Thus, calculation of the expected value of this likelihood requires not only the first one or two moments of the latent variable posterior, as in our previous examples, but also the expectation of $\log s$.

To avoid this, we optimize the marginalized posterior (6.11) directly by numerical gradient-based methods. Conceptually, this may be thought of as a simple gradient ascent algorithm, although, in

practice, better results are obtained by use of a quasi-second order method (see, for example, Press *et al.* 1993). Such optimizations can be efficiently executed using numerical methods software such as the MATLAB package.

6.4 Goodness of Fit

While the basic structure of the statistical model described in this chapter has been chosen to embody our beliefs about the origin of neuronal variability, the exact densities used (that is, the gamma and Poisson) have by and large been selected arbitrarily. Both are high entropy distributions, which is appropriate in situations where little constraining knowledge is available, but it must be admitted that, to a significant extent, the choice has been driven by mathematical expediency. In some details, we must expect the model to be incorrect. As was already pointed out, both the refractory period and the tendency of some cells to fire in bursts, violate the independence of counts assumption inherent in the Poisson process. Similarly, we have no guarantee that the scaling will be gamma distributed, nor even that the variability due to excitability can be expressed entirely as multiplicative scaling (on this last point see Linden 1999).

In this section we will investigate through Monte-Carlo means the degree to which the model is appropriate to describe a given set of spike trains recorded in mammalian cortex. These data were collected by J. Linden and A. Grunewald from area LIP of 2 macaque monkeys. For data collection procedures and further information the reader is referred to Linden (1999).

In general, such goodness of fit testing is a difficult problem. We have encountered the issue of model selection repeatedly in this dissertation, where the best of a group of competing models needs to be selected. In this case, though, there is no clear alternative. Based solely on the single model and the available data, we would like to decide whether or not the model is acceptable; that is, whether it is plausible that the data are indeed distributed in the manner specified. The general framework for making such decisions falls within the Neyman-Pearson significance testing literature that is fundamental to traditional developments of statistical theory (see, for example, Hoel *et al.* 1971). Many specific tests have been developed for particular simple distributions (some examples may be found in Zar 1998). For one dimensional data a general technique, known as the Kolmogorov-Smirnov test, is available to assess the validity of an arbitrary distribution (see, for example, Press *et al.* 1993). This can be extended into a small number of dimensions (Fasano and Franceschini 1987), but for more complicated models, describing higher dimensional data, as in the current instance, such straightforward techniques are not available.

Instead, we approach the problem by a novel Monte-Carlo technique, asking whether the obtained likelihood of the best fit model for the observed data matches corresponding values obtained for simulated data known to be generated from the distribution. The steps of the procedure are as

follows.

- Given a set of observed spike trains $\mathcal{X}^o = \{\mathbf{x}_1^o \ldots \mathbf{x}_N^o\}$, find the MAP parameter estimates $\boldsymbol{\lambda}^o$ and α^o.
- Calculate the likelihood on the observed data

$$\ell^o = \ell_{\mathcal{X}^o}\left(\boldsymbol{\lambda}^o, \alpha^o\right) \tag{6.12}$$

- Repeat for $s = 1 \ldots S$:
  - Generate a set of simulated spike trains from the optimized model

$$\mathcal{X}^s = \{\mathbf{x}_1^s \ldots \mathbf{x}_N^s\} \sim \text{ iid } \mathsf{P}_{\boldsymbol{\lambda}^o, \alpha^o}\left(\mathbf{x}\right) \tag{6.13}$$

  - Re-fit the model to the simulated data $\mathcal{X}^s$ to obtain new MAP estimates $\boldsymbol{\lambda}^s, \alpha^s$.
  - Obtain the optimal likelihood on the simulated data

$$\ell^s = \ell_{\mathcal{X}^s}\left(\boldsymbol{\lambda}^s, \alpha^s\right) \tag{6.14}$$

- Find the rank of the observed likelihood within the set of simulated likelihoods

$$r^o = |\{s : \ell^s < \ell^o\}| \tag{6.15}$$

If this procedure is repeated a number of times — each time starting with a different set of observed spike trains, perhaps derived from a different cell — and if the model represents the correct family of distributions, we would expect the resultant ranks to be uniformly distributed between 0 and S.

Two points about the process might require elucidation. First, the simulated data are generated using the MAP parameter values so that the likelihoods measured in the simulations are drawn from the same region of the parameter space as the true likelihoods. Likelihoods under simulated data taken in an an entirely different parameter regime might be quite different. Second, the likelihoods under the simulated data need to be evaluated at the re-fit parameter values so as to avoid a bias due to over-fitting. If this were not done, we would expect the observed likelihoods ℓ^o to be larger than the simulated values, as the parameters would be perfectly tailored to the observed data alone.

In principle, we may now test for uniformity of the ranks by a Kolmogorov-Smirnov or other, more specialized, hypothesis test. In practice it is obvious from inspection that, in this case, the ranks are not uniformly distributed. Figure 6.1 shows the ranks obtained using different groups of

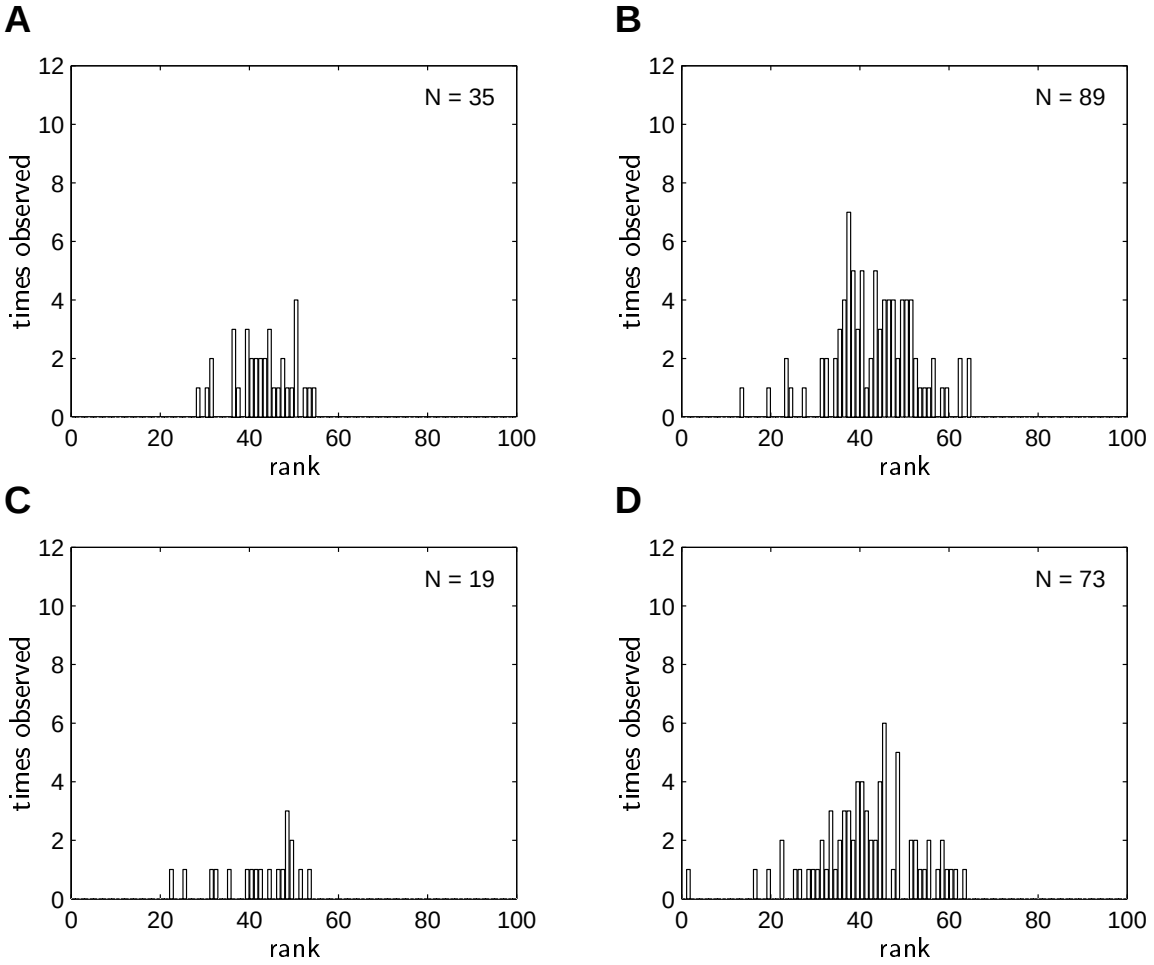

Figure 6.1: Distributions of likelihood ranks

cells under different stimulus conditions. Each panel represents a set of spike trains collected under identical experimental conditions. Only spike trains from cells that appeared to be responsive under the specific conditions were used (the number of these is given by the quoted value of N in each panel), and a single set was taken from each such cell. In each case, the number of simulations, S, was 100.

It is clear from the distributions in figure 6.1 that the ranks are far from uniformly distributed. This suggests that the model we have developed in this chapter is not, in fact, an accurate description of the recorded data. However, had the model been entirely off base, we might have expected the simulated data to almost always have yielded higher best-fit likelihoods than the real observations. For example, if the smoothing invoked by the prior were too severe then the derived intensity function would be greatly inaccurate for the real data, leading to much lower probabilities. Clearly, this is not the case either; almost half the time ℓ^s is smaller than ℓ^o. Thus, we conclude that while the model

is not correct, it is reasonably capable of describing the data. In particular, it would be difficult to tell, simply by looking at the optimal likelihood, whether a given set of spike trains were genuine neural data or simply simulations.

A further point of interest in figure 6.1 is that the distributions of ranks obtained for the four different experimental conditions — and frequently, from different cells — are extremely similar. We might take this as evidence that the statistics of the spike trains from these different cells and under these different experimental conditions are actually the same. Thus, while our current model is inadequate, we might hope that by some refinement we can, in fact, find an appropriate model.

6.5 Clustering Spike Trains

It is often a matter of scientific interest to ask whether the cells within a given area of the brain fall into clusters based on the time-courses of their responses to a given stimulus. If such clusters are apparent, they may indicate the presence of distinct sub-populations of neurons that play different rôles in the neural computation.

A common difficulty encountered when attempting to apply traditional clustering techniques such as the k-means algorithm or its variants, to spike trains, is the problem of finding a suitable metric. Such algorithms require a notion of distance between two spike trains, but how is such a distance to be defined? One approach has been to smooth the spike trains, by binning or by convolving with a Gaussian kernel, and then to sample each such smoothed spike train to obtain a vector representation (see, for example, Richmond and Optican 1987; Optican and Richmond 1987; McClurkin *et al*. 1991). These vectors are then treated as though they were embedded in the standard Euclidean inner-product space. There is, however, no *a priori* reason to expect such a distance to be an appropriate metric for spike train clustering. This point is discussed at some length by Victor and Purpura (1997), who propose an alternative metric, though also on an *ad hoc* basis.

Fortunately, we can avoid this problem. In chapter 2 we saw that, in many cases, the generative modeling approach to clustering is to be preferred. In particular, this is true if we are interested in identifying the process from which the observed data arose, rather than simply grouping the data themselves. The appropriate generative model in such situations is the mixture model given by the weighted sum of M component distributions:

$$\mathsf{P}_{\boldsymbol{\theta}}(\mathbf{x}) = \sum_{m=1}^{M} \pi_m \mathsf{P}_{\boldsymbol{\theta}_m}(\mathbf{x}) \tag{6.16}$$

The parameters of the mixture decompose into independent and disjoint sets $\boldsymbol{\theta} = (\boldsymbol{\theta}_1 \ldots \boldsymbol{\theta}_M, \pi_1 \ldots \pi_M)$, where the parameters $\boldsymbol{\theta}_m$ describe the mth component or cluster. Learning algorithms for such mix-

tures were discussed at length in chapters 2 and 3.

Such an approach effectively sidesteps the issue of identifying a suitable metric within the space of spike trains. The clusters are no longer described within the observation space; instead, they are described by the parameters $\boldsymbol{\theta}_m$ which live in a different space altogether. We no longer need to compute the separation between two spike trains: we need only find the "distance" between a spike train and the cluster parameters. A natural candidate for such a distance is obvious: the probability of the spike train under the cluster model. Thus, the probabilistic treatment espoused throughout this dissertation allows us to rigourously arrive at a unique clustering solution from only a few explicitly stated assumptions about the distributions of spike trains.

To this point, we have regarded each spike train $\mathbf{x}_n$ as a separate observation; now, we will instead treat all of the spike trains collected from the same cell under the same experimental conditions as a single outcome of the generative model. For the ith cell-experiment pair we can collect the N_i individual count vectors into a matrix X_i, in which each count vector appears as a column. Careful inspection of the probability (6.7) reveals that, in fact, we are only interested in the marginal sums of this matrix. Thus, we compute and store the following sufficient statistics: the sum of the count vectors $\mathsf{X}_i\mathbf{1}$, the vector of total spike counts $\mathsf{X}_i^\mathsf{T}\mathbf{1}$, and the total of all the elements $\mathbf{1}^\mathsf{T}\mathsf{X}_i\mathbf{1}$. In these expressions the vector $\mathbf{1}$ should be taken to contain either T or N_i ones as appropriate.

We can then write the form of the mth component probability distribution, written in terms of the Fourier domain intensity $\boldsymbol{\phi}_m$ and the stability α_m,

$$\mathsf{P}_m\left(\mathsf{X}_i\right) \propto e^{\boldsymbol{\phi}_m^\mathsf{T}\mathsf{F}\mathsf{X}_i\mathbf{1}}\alpha_m^{N_i\alpha_m}\left(e^{\boldsymbol{\phi}_m^\mathsf{T}\mathsf{F}}\mathbf{1}+\alpha_m\right)^{-\left(\mathbf{1}^\mathsf{T}\mathsf{X}_i\mathbf{1}+N_i\alpha_m\right)}\exp\left[\mathbf{1}^\mathsf{T}\log\left(\frac{\Gamma(\mathsf{X}_i^\mathsf{T}\mathbf{1}+\alpha_m\mathbf{1})}{\Gamma(\alpha_m)}\right)\right] \tag{6.17}$$

In the final factor, the gamma function and the logarithm should be taken to apply element by element. We have left out a factor given by the product of the factorials of each of the elements in X_i. This factor is identical across all of the component distributions and thus has no impact on any of the optimization algorithms and need never be computed.

We then fit a mixture model for the entire ensemble of recordings taken across multiple cells $\mathcal{X} = \{\mathsf{X}_i\}$, given by $\mathsf{P}_{\boldsymbol{\theta}}\left(\mathcal{X}\right) = \prod_i \sum_m \mathsf{P}_m\left(\mathsf{X}_i\right)$. In doing so, we assume that a "cluster" of spike trains are such that they may have arisen from exactly the same intensity function, although with possibly different scalings. The "extent" of the cluster is defined by the model, as well as by the learned value of the stability parameter.

For the single component model, the introduction of the prior was important to achieve regularized estimation. In the mixture, this regularization is, if anything, more important as the complexity of the model has increased. We choose the prior on the parameter set $\{\boldsymbol{\phi}_m\} \cup \{\alpha_m\}$ to factor over the different components; that is, the intensity function and stability for one component are *a priori* independent of those of any other component distribution. For any one component we choose the

priors on $\boldsymbol{\phi}_m$ and α_m to be exactly as before. The covariance matrix C is taken to be common to all of the clusters. The mixing parameters π_m are subject to a uniform prior: this does not affect the results of the estimation and will be not be written explicitly.

The basic EM algorithm suitable for learning in such models was described in section 2.4. We recall that the E-step involves computation of responsibilities according to (2.9)

$$r_{m,i} = \frac{\pi_m \mathsf{P}_m\left(x_i\right)}{\sum_l \pi_l \mathsf{P}_l\left(x_i\right)} \tag{6.18}$$

where, the component distributions are given by (6.17). The M-step update of the mixing probabilities is common to all mixture models (2.12)

$$\pi_m \leftarrow \frac{\sum_i r_{m,i}}{|\mathcal{X}|} \tag{6.19}$$

The update of the component parameters in the maximum likelihood context of chapter 2 was given by (2.15)

$$\theta_m \leftarrow \operatorname*{argmax}_{\theta_m} \sum_i r_{m,i} \log \mathsf{P}_{\theta_m}\left(\mathsf{X}_i\right) \tag{6.20}$$

where θ_m stands for the parameters of the mth component. In the present example, however, we have a non-trivial prior distribution on the component parameters. Given our assumption that the prior factorizes over the different models, we can correct (6.20) by the addition of the log-prior for the mth model to the right hand side. The updated parameters of the mth component are thus obtained by optimizing the expression

$$\begin{aligned} Q(\boldsymbol{\phi}_m, \alpha_m) = \log Z - \frac{1}{2} e^{\boldsymbol{\phi}_m^{\mathsf{T}} \mathsf{F}} \mathsf{R} e^{\mathsf{F}^{\mathsf{T}} \boldsymbol{\phi}_m} - \frac{1}{\alpha_m} \\ + \sum_i r_{m,i} \quad \Big[\boldsymbol{\phi}_m^{\mathsf{T}} \mathsf{F} \mathsf{X}_i \mathbf{1} - (\mathbf{1}^{\mathsf{T}} \mathsf{X}_i \mathbf{1} + N\alpha_m) \log(e^{\boldsymbol{\phi}_m^{\mathsf{T}} \mathsf{F}} \mathbf{1} + \alpha_m) \\ + N\alpha_m \log \alpha_m + \mathbf{1}^{\mathsf{T}} \log \left(\frac{\Gamma(\mathsf{X}_i^{\mathsf{T}} \mathbf{1} + \alpha_m)}{\Gamma(\alpha_m)} \right) \Big] \end{aligned} \tag{6.21}$$

As before, this optimization must be performed numerically, and thus, the computational cost of the M-step is considerably greater than that of the E-step. It is useful to recall the Generalized EM (GEM) algorithm, mentioned briefly in section 1.8, in which the M-step is only partially completed; that is, the free energy is increased by the update of the parameters, but not necessarily maximized. This generalization shares the guaranteed convergence with the standard EM algorithm, but is more efficient. In the present case, this partial completion is equivalent to executing only a limited number of steps of the numerical optimization at each M-step.

The GEM algorithm described above was run on a subset of the data described previously, that was collected from different cells under the same experimental conditions. The results are shown

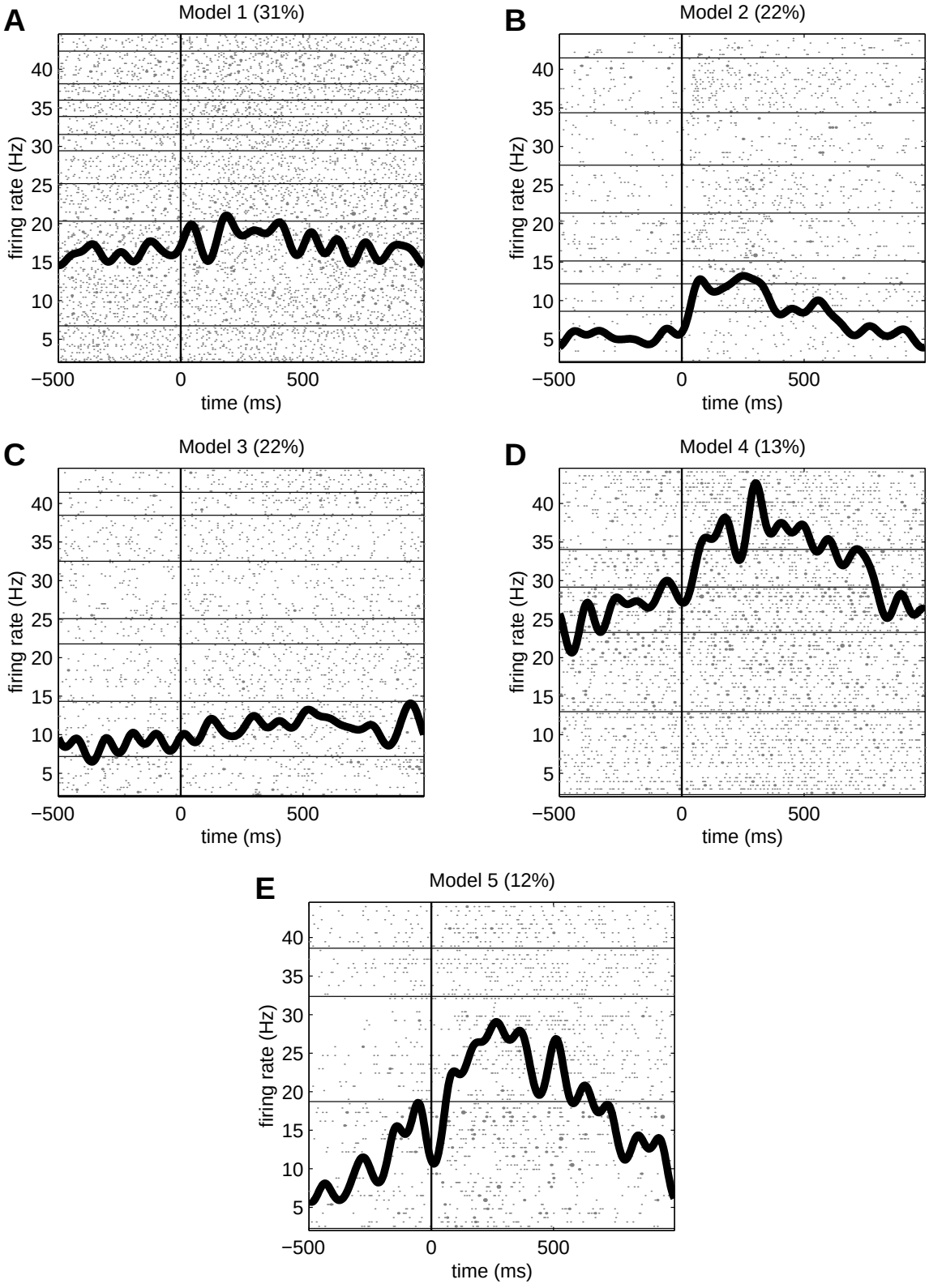

Figure 6.2: Clusters of spike trains

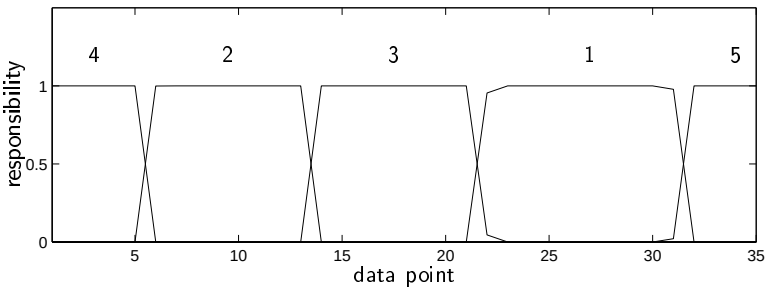

Figure 6.3: Responsibilities of the different models.

in figure 6.2. The size of the model was determined by the BIC penalized likelihood procedure (see section 1.3), which yielded a mixture of five components. The intensity function learned for each of these components is shown by the heavy black line in each panel of the figure. The mixing probabilities are indicated by the percentage figures above each panel. Cells have been assigned to the most likely cluster (that is, the one with the largest responsibility for the data from the cell), and the corresponding spike trains then shown in the background of the appropriate panel. The representation is similar to the conventional spike raster diagram: each row of dots represents a single trial; the presence of a dot time indicates that at least one spike was counted in a 5ms window around that time; the size of the dot indicates the number of spikes. The horizontal black lines separate spike trains from different cells.

Do the spike trains classified in figure 6.2 really fall into five distinct clusters? The fact that BIC model selection rejected the option of more components in the mixture suggests that this may well be the case. As a further reassurance we can examine the posterior assignment probabilities, or responsibilities (6.18), under the maximum likelihood solution. These values indicate the surety with which each data point is assigned to each cluster. If the components tended to share the responsibility for each spike train it would suggest that the clusters were not well separated. The responsibilities of each of the five component models are shown in figure 6.3. Each line shows the assignment probabilities of one model, indicated by the number above the line, for all of data; the data have been reordered to group spike trains assigned to the same cluster together. In all cases, only one model has high responsibility, very close to 1. This suggests that the clusters shown in figure 6.2 really are well separated.

6.6 Summary

In this chapter we have introduced a latent variable model to describe spike trains generated by a neuron under constant experimental conditions. The model is designed to capture certain recent observations about the statistics of neural responses: in particular, the fact that the variability in

cortical spike trains is often greater than that predicted by the Poisson process assumption, and that in many cases this greater variability might result from changes in the overall excitability of the neuron or cortical area. Although the EM algorithm involves a difficult E-step, it proves to be possible to fit the model by direct numerical optimization.

Using a Monte-Carlo goodness of fit procedure, we saw that the model does not describe the statistics of spiking exactly. However, the maximal likelihood values for the best-fit model under real neural data are quite similar to the values under simulated data generated from the model itself. Thus, we conclude that model is a reasonable, but not exact description.

The statistical model provides a rigorous foundation on which to base two analyses of neural data. First, maximum *a posteriori* optimization of the model with a suitable prior imposed on the parameters, leads to a smoothed estimate of the underlying spike-rate intensity. This technique provides a solid statistical basis for the smoothing, as well as correctly accounting for biases that might be introduced by any variable excitability. Second, by use of a mixture of such models, we are able to identify clusters of cells whose spike trains in response to the same stimuli are similar. *Ad hoc* methods for clustering spike trains suffer from the serious difficulty of the absence of a natural metric. In contrast, the probabilistic procedure avoids the issue of a distance measure entirely, and leads to a natural clustering algorithm.