Statistical analysis of neural data:
The expectation-maximization (EM) algorithm*

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

August 6, 2009

Contents

*Thanks to R. Kass for helpful feedback preparing these notes.

So far we have focused on the problem of estimating encoding models for which all the necessary pieces are completely observed: for example, in the GLM setting, we have assumed that both the spike times and the stimuli X are observed noiselessly, and there were no other unobserved ("latent") components of the model that we needed to observe in order to compute the likelihood. In many important cases this assumption of complete observations is overly restrictive: spike times may be observed with some noise, for example, or we might have reason to believe that there are unobserved variables (e.g. the attentive state of the animal) that affect the probability of spiking. Indeed, much of the remainder of this book will deal with "state-space" (aka "hidden Markov") models, where the hidden Markovian state-space variables play a key role in the system dynamics and in our approach to inference in these models. Thus we would like to develop techniques to deal with these phenomena; as we will see, this extension grants us a great deal of additional flexibility.

The "expectation-maximization" (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1996) will be a primary tool in dealing with these situations. In this chapter we will derive this algorithm, discuss its properties, and illustrate its use in a variety of neural examples. To properly introduce the basic ideas behind the EM method, we will need to develop just a bit of background on an optimization technique known as "bound optimization." However, first it is useful to begin with an important concrete example, to give the basic flavor of the algorithm.

1 Example: Mixture models and spike sorting

The standard model of spike waveforms in extracellular recordings in illustrated in Fig. 1 (Lewicki, 1998; Pouzat et al., 2004; Quian Quiroga, 2007): the extracellular voltage signal is filtered and thresholded, and then each snippet of voltage triggered on a threshold crossing is classified, using some clustering algorithm, as a spike or non-spike. The simplest probabilistic model underlying this clustering process is a Gaussian mixture model (McLachlan and Peel, 2000), in which each voltage waveform snippet $\vec{V}$ may be represented as a sample from a mixture distribution

$$p(\vec{V}) = \sum_{z=0}^{J} \alpha_z p_z(\vec{V}),$$

with J denoting the number of distinct units present in the recording, z indexing the different units, and the individual distributions given, for example, by the multivariate Gaussian

$$p_z(\vec{V}) = \mathcal{N}_{\vec{\mu}_z, C_z}(\vec{V})$$

with means $\vec{\mu}_z$ and covariance matrices C_z[1]. The mixture probabilities α_z satisfy $\sum_z \alpha_z = 1$ and $\alpha_z \geq 0$ for all z, as usual. Thus the parameter vector θ of interest here includes our information about the underlying mixture components and weights,

$$\theta = \{(\vec{\mu}_z, C_z, \alpha_z)_{0 \leq z \leq J}\}.$$

How can we go about estimating the parameters of this model? Assume we observe a set of voltage snippets $\{\vec{V}_i\}$. Define z_i to be the identity of the mixture component from

[1]Typically we take $\vec{\mu}_0 = 0$; this is the "noise" cluster corresponding to the absence of a spike. In addition, it is often a reasonable approximation (particularly in low-SNR recordings) to take all the covariances to be equal, $C_z = C$. However, it turns out to be a little simpler to describe the case of general $\vec{\mu}_z$ and C_z here; the constrained-C_z case may then be derived as a fairly straightforward extension.

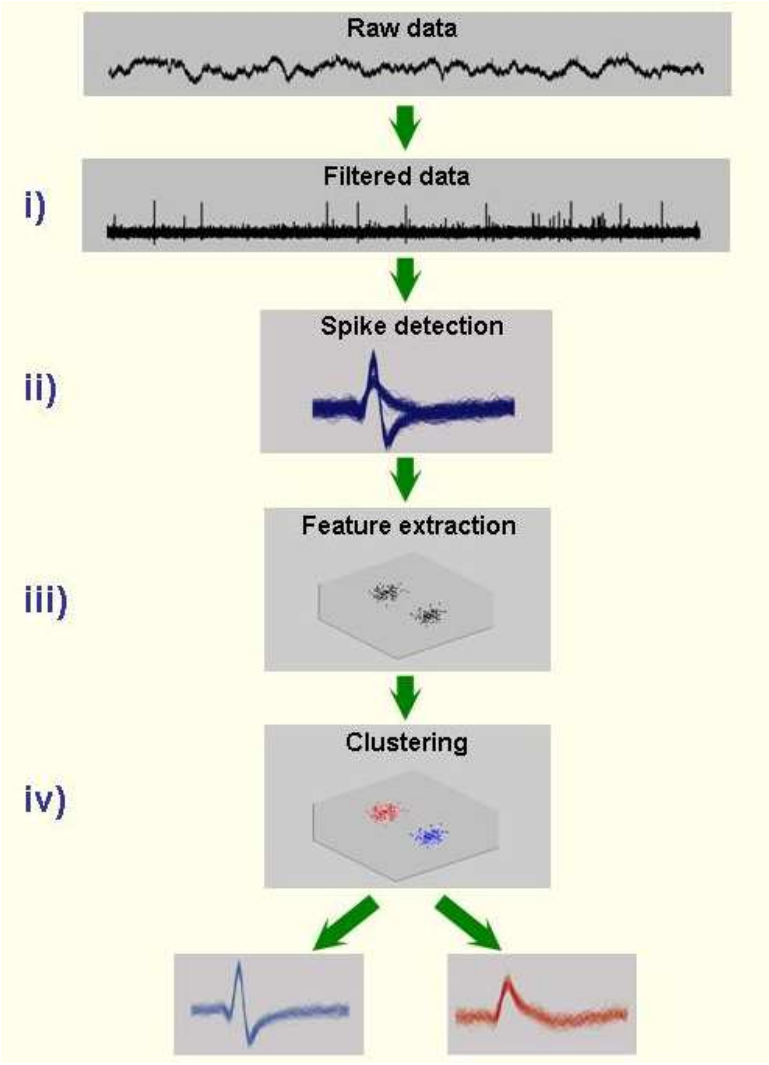

Figure 1: Schematic overview of the process of sorting spikes from extracellular recordings (adapted from (Quian Quiroga, 2007)). Step i) The continuous raw data is band-pass filtered, e.g. between 300 Hz and 3000 Hz, to discard slow fluctuations in the voltage signal and high-frequency noise, respectively. Step ii) Spikes are detected, usually using simple threshold-crossing methods. Step iii) Relevant features of the spike shapes are extracted, often via PCA or some other dimensionality reduction method. Step iv) These features are the input of a clustering algorithm that performs the classification of the spike waveforms.

which the i-th voltage sample was actually drawn; of course, we do not observe z_i directly. Now if we knew all of the identities $\{z_i\}$, then estimating θ by maximum likelihood would be easy: we just divide the data set $\{\vec{V}_i\}$ into $J+1$ groups, according to the labels $\{z_i\}$, and then estimate $\vec{\mu}_z$ and C_z by maximum likelihood individually within each group. Similarly, the maximum likelihood estimate of the mixture probability vector α_z is just given by the empirical frequency of each label z.

Conversely, given the parameter vector θ we could very easily estimate the labels $\{z_i\}$,

by computing the posterior probability $p(z_i = j|\theta, V_i)$ that z_i is equal to j given θ and the observed snippet $\vec{V_i}$, for $j = 0, 1, \ldots$ or J. Of course, we don't know θ; this is parameter we are trying to infer. But if we have a decent initial guess about what θ might be, then it is natural to iterate these two steps: (1) given our current estimate of θ, infer the posterior probability of the mixture labels $\{z_i\}$ given θ and the observed data $\{\vec{V_i}\}$, and (2) given these "soft" probabilistic assignments of the mixture labels $\{z_i\}$, use some likelihood-based estimate, possibly weighted by our confidence in the assignments $\{z_i\}$, to update our estimate of the parameter θ. It turns out that the EM algorithm for this mixture model has exactly this alternating form, as we will see in more detail in section 5 below. However, first we must lay some groundwork for the derivation of EM in the general case.

2 The method of bound optimization via auxiliary functions provides a useful alternative optimization technique

Perhaps the simplest derivation of the EM algorithm is based on the idea of optimization via auxiliary bound functions. This idea is useful in its own right and is easy to describe: in many settings it may be difficult to directly maximize an objective function $F(x)$ by standard gradient-based methods (conjugate gradient descent or Newton-Raphson). An alternate approach which is sometimes more effective involves the construction of an "auxiliary" function $Q(x, x')$ with the following three properties:

1. $Q(x, x')$ is a lower bound on $F(x)$:

$$F(x) \geq Q(x, x') \ \forall x'; \tag{1}$$

2. the values of $F(x)$ and $Q(x, x')$ match at $x = x'$:

$$F(x) = Q(x, x). \tag{2}$$

3. $Q(x, x')$ may be efficiently maximized as a function of x, for any fixed x';

If these conditions are met, it is not hard to see that the algorithm

$$x_{j+1} = \arg\max_x Q(x, x_j)$$

leads to a monotonically increasing method for optimizing the original function $F(x)$, since:

$$F(x_{j+1}) \geq Q(x_{j+1}, x_j) \geq Q(x_j, x_j) = F(x_j);$$

the first inequality is by property (1) of the auxiliary function $Q(.,.)$, the second is by the fact that $x_{j+1} = \arg\max_x Q(x, x_j)$, and the last equality is by property (2). See Fig. 2 for an illustration. If $Q(x, .)$ is significantly easier to optimize than $F(x)$, then this auxiliary function approach can lead to more efficient optimization schemes than the direct approaches discussed above (Darroch and Ratcliff, 1972; Dempster et al., 1977; Collins et al., 2000; Sha et al., 2003)[2]. Of course, it is very important to remember that this bound optimization method is only guaranteed to increase the objective function on each iteration; thus if $F(x)$

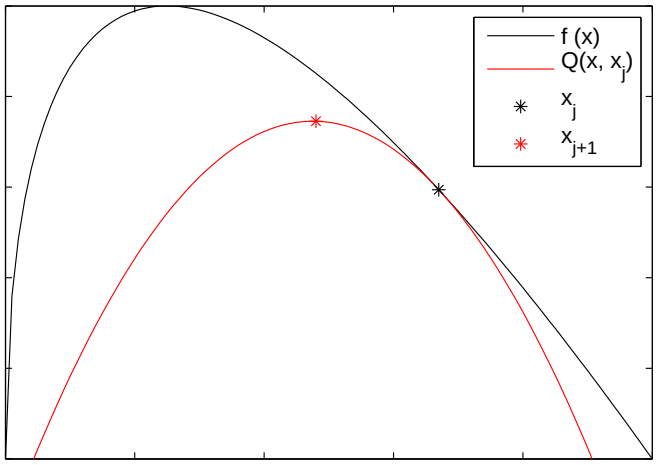

Figure 2: Illustration of the bound optimization idea: the j-th function evaluation $F(x_j)$ is always less than or equal to $F(x_{j+1})$. Also note that the first derivative of $F(x)$ and $Q(x, x_j)$ are equal at the point $x = x_j$ (as discussed further in section 11), while we have the bound $d^2Q(x, x_j)/dx^2|_{x=x_j} \leq d^2F(x)/dx^2|_{x=x_j}$ on the second derivatives (as discussed further in section 12).

is multimodal the bound optimization method may be just as prone to finding suboptimal local maxima as any other ascent algorithm.

One simple but useful application of this idea is as follows (Krishnapuram et al., 2005). If $F(x)$ is a smooth, strictly concave function such that the Hessian of $F(x)$, $H(x)$, satisfies the lower bound

$$H(x) \geq H_{lb} \quad \forall x$$

(where the matrix inequality $H \geq H_{lb}$ is interpreted, as usual, to mean that $H - H_{lb}$ is a positive semidefinite matrix), then the quadratic function

$$Q(x, x') = F(x') + \nabla F(x')^t(x - x') + \frac{1}{2}(x - x')^t H_{lb}(x - x')$$

satisfies our three auxiliary function conditions. Clearly, we have

$$\arg\max_x Q(x, x') = x' - H_{lb}^{-1}\nabla F(x'),$$

so the auxiliary updates in this case reduce to the simple Newton-like updates

$$x_{j+1} = x_j - H_{lb}^{-1}\nabla F(x_j).$$

The nice thing about this choice is that we only need to compute H_{lb}^{-1} once for all x', so each step may be computed in just $O(\dim(x)^2)$ time for the matrix multiplication, instead of the

[2]Also, note that we may relax the definition of x_{j+1} to be any x such that $Q(x, x_j) \geq F(x)$, and the proof still holds; thus, we may do partial maximizations and still ensure that $F(.)$ increases (or stays the same) on each iteration. This is helpful because partial optimizations on each iteration may be much quicker and just as effective as full optimizations.

usual $O(\dim(x)^3)$ required to invert a general matrix in the full Newton step. (Of course, if H_{lb}^{-1} has any useful structure — e.g., if H_{lb}^{-1} is Toeplitz or if H_{lb} or H_{lb}^{-1} is banded — then further speedups are possible.)

For example, consider modeling a neuron whose firing rate is given by $\lambda(t) = f(\vec{k} \cdot \vec{x}(t))$, with f chosen so that $0 \leq d^2 f(u)/du^2 \leq c_1$ and $-c_2 \leq d^2 \log f(u)/du^2 \leq 0$ for some finite constants c_1, c_2. The likelihood here is, as usual,

$$L(\vec{k}) \equiv \sum_i \log f(\vec{k} \cdot \vec{x}(t_i)) - \int_0^T f(\vec{k} \cdot \vec{x}(t))dt,$$

and the Hessian of this likelihood (which depends on $\vec{k}$) may be bounded from below by a fixed matrix H_{lb}:

$$\begin{aligned} \nabla\nabla_{\vec{k}} L(\vec{k}) &= \sum_i \frac{d^2 \log f(u)}{du^2}\Big|_{u=\vec{k}\cdot\vec{x}(t_i)} \vec{x}(t_i)\vec{x}(t_i)^t - \int_0^T \frac{d^2 f(u)}{du^2}\Big|_{u=\vec{k}\cdot\vec{x}(t)} \vec{x}(t)\vec{x}(t)^t dt \\ &\geq -c_2 \sum_i \vec{x}(t_i)\vec{x}(t_i)^t - c_1 \int_0^T \vec{x}(t)\vec{x}(t)^t dt \\ &\equiv H_{lb}, \end{aligned}$$

with H_{lb} independent of $\vec{k}$. Thus a convergent, simple update rule for maximizing the likelihood may be derived quite easily: $\vec{k}_{j+1} = \vec{k}_j - H_{lb}^{-1} \nabla L(\vec{k}_j)$, where, again, H_{lb}^{-1} need only be computed once, greatly accelerating the iteration when $\dim(\vec{k})$ is large. On the other hand, as we will discuss further in section 12, this bound optimization method may require more iterations to converge than direct Newton-Raphson optimization.

3 The EM algorithm for maximizing the likelihood given hidden data may be derived as a bound optimization algorithm

We are now in a position to introduce the famous EM (Expectation-Maximization) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1996). As discussed above, in many cases, we are interested in "latent variable" models, in which the likelihood is of the form

$$p(y|\theta) = \int p(y, z|\theta)dz,$$

with y the observed data, θ the parameter of interest, and z some "latent," unobserved, data. Note that a direct approach towards maximizing this likelihood requires that we marginalize out z, and this integration may be difficult in general. The EM algorithm was developed as a clever method for estimating θ without having to compute this integral.

One method for deriving the EM algorithm is via the auxiliary function approach described above[3]. We define our objective function as the log-likelihood

$$F(\theta) = \log p(y|\theta) = \log \int p(y, z|\theta)dz.$$

[3]It is also possible to justify the EM algorithm in terms of alternating maximization of a certain "free energy" function (Neal and Hinton, 1999; Saad and Opper, 2001; Beal and Ghahramani, 2003), but we will not take this approach here.

Now our strategy is to use Jensen's inequality to develop a bound on $F(\theta)$, and from there obtain a suitable auxiliary function. We have

$$\begin{aligned} F(\theta) &= F(\theta') + \log \frac{p(y|\theta)}{p(y|\theta')} \\ &= F(\theta') + \log \int p(z|y,\theta') \frac{p(z|y,\theta)p(y|\theta)}{p(z|y,\theta')p(y|\theta')} dz \\ &= F(\theta') + \log \int p(z|y,\theta') \frac{p(y,z|\theta)}{p(y,z|\theta')} dz \\ &\geq F(\theta') + \int p(z|y,\theta') \log \frac{p(y,z|\theta)}{p(y,z|\theta')} dz \\ &= F(\theta') - \frac{1}{p(y|\theta')} D\left(p(y,z|\theta'); p(y,z|\theta)\right) \\ &\equiv Q(\theta,\theta'), \end{aligned} \tag{3}$$

where the equalities follow from straightforward algebraic manipulations, $D(p;q)$ denotes the Kullback-Leibler divergence between the two distributions p and q,

$$D(p;q) = \int p(z) \log \frac{p(z)}{q(z)} dz,$$

and in the last line we have defined our auxiliary function $Q(\theta,\theta')$; the inequality is an application of Jensen's bound, and establishes property (1) of our auxiliary function. To establish property (2), simply note the well-known fact that $D(p;p) = 0$ (which, in turn, follows from the equality conditions of Jensen's inequality). Thus we have established $Q(\theta,\theta')$ as a suitable auxiliary function for our marginal log-likelihood objective function $\log p(y|\theta)$ here.

Now to compute $\arg\max_\theta Q(\theta,\theta')$ we write out $Q(\theta,\theta')$ and drop all the terms which are independent of θ to obtain

$$\begin{aligned} \arg\max_\theta Q(\theta,\theta') &= \arg\max_\theta \int p(z|y,\theta') \log p(y,z|\theta) \\ &= \arg\max_\theta E_{p(z|y,\theta')} \log p(y,z|\theta). \end{aligned}$$

Thus we may break up each iteration of the auxiliary function optimization into two steps: an "E-step" corresponding to the computation of the expectation $E_{p(z|y,\theta')} \log p(y,z|\theta)$, and an "M-step" corresponding to a maximization of this function as a function of θ. Each iteration of these two steps is guaranteed never to decrease the loglikelihood $\log p(y|\theta)$, and therefore we can use these iterations as a tool for (locally) optimizing $p(y|\theta)$.

In many cases it turns out to be much easier to optimize $E_{p(z|y,\theta')} \log p(y,z|\theta)$ as a function of θ, rather than the original objective function $\log p(y|\theta)$; indeed, we will see a number of examples below in which the former optimization can be performed analytically (and therefore effectively instantaneously), while directly optimizing $\log p(y|\theta)$ is relatively intractable. On the other hand, EM may require more iterations to reach the optimizer than do "direct" approaches like Newton-Raphson or conjugate-gradient; we will discuss the convergence rate of EM in more detail in section 12. It is also worth noting that we never need to explicitly calculate the function $\log p(y|\theta)$; again, this is useful because in many cases computing the integral $\int p(y,z|\theta)dz$ directly may be intractable. Finally, the quantities computed in the

E-step are often of independent interest; for example, we can easily read off the gradient $\nabla_\theta \log p(y|\theta)$ from the output of the E-step (see section 11). We will discuss a variety of further examples below.

4 EM may easily be adapted to optimize the log-posterior instead of the log-likelihood

So we have demonstrated that the EM algorithm may be used to (locally) maximize the likelihood. Of course, the MLE computed via EM is just as susceptible to overfitting effects as the MLE computed via any other algorithm — in fact, in situations where we have hidden variables and may therefore apply EM, overfiting is an even bigger concern, since in general complete observations are more informative than incomplete observations (and as we discussed previously, overfitting is in some sense a symptom of not having sufficient information to constrain our parameter estimates).

Therefore it is natural to ask if we can use EM to optimize a log-posterior, instead of a log-likelihood. Luckily, the required modification is quite straightforward: it turns out that the E-step remains unchanged, and the M-step is the same as before but with another term (corresponding to the log-prior) included in the objective function. In particular, we know that the function $Q(\theta, \theta')$ defined in eq. (3) is an auxiliary function (i.e., satisfies conditions (1) and (2)) for the loglikelihood $\log p(y|\theta)$; therefore, $Q(\theta, \theta') + \log p(\theta)$ is an auxiliary function for the (unnormalized) log-posterior $\log p(y|\theta) + \log p(\theta)$. To evaluate this auxiliary function $Q(\theta, \theta') + \log p(\theta)$ we need to compute the expectation $E_{p(z|y,\theta')} \log p(y, z|\theta)$ just as before; thus the E-step is unchanged. Maximizing this auxiliary function (the M-step) requires that we optimize $Q(\theta, \theta') + \log p(\theta)$ as a function of θ, instead of just maximizing $Q(\theta, \theta')$. We will encounter a number of examples below.

5 Example: Deriving the EM algorithm for the mixture model (spike sorting) case

Now we may return to the spike sorting example discussed in section 1 above, and show how to derive the EM algorithm in this particular case. First we must identify the ingredients of the general EM algorithm discussed above: the observed data y are the N observed voltage snippets $\vec{V}_i$, and the unobserved latent variable z corresponds to the identities of the mixture components from which each voltage sample was actually drawn. We have already identified the parameter θ as the set of mixture means, covariances, and probabilities $\{(\vec{\mu}_z, C_z, \alpha_z)_{0 \le z \le J}\}$. We may now write the complete log-likelihood as

$$\begin{aligned}
\log p(y, z|\theta) &= \log p(z|\theta) + \log p(y|z, \theta) \\
&= \sum_{i=1}^{N} \log p(z_i|\theta) + \sum_{i=1}^{N} \log p(y_i|z_i, \theta) \\
&= \sum_{i=1}^{N} \log \alpha_{z(i)} + \sum_{i=1}^{N} \log \mathcal{N}_{\mu_{z(i)}, C_{z(i)}}(\vec{V}_i) \\
&= \sum_{i=1}^{N} \left[\log \alpha_{z(i)} - \frac{1}{2} \left(\log |C_{z(i)}| + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right) \right] + const., \quad (4)
\end{aligned}$$

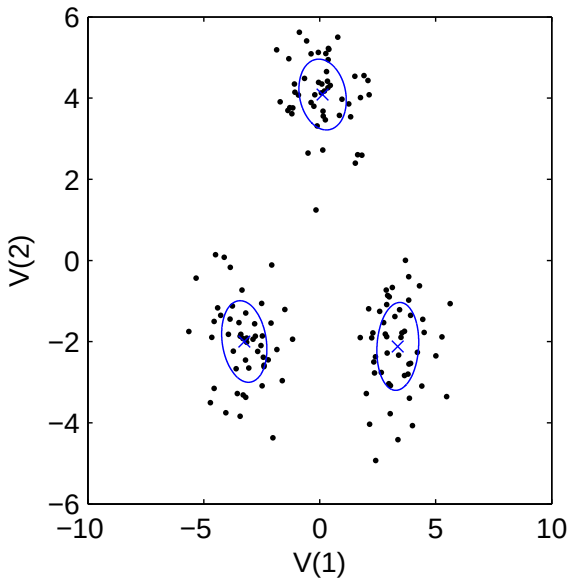

Figure 3: Illustration of the mixture model: we observe the data shown in scatterplot form here (two-dimensional simulated data were used, for simplicity), and the EM algorithm run with three mixture components recovers the means and covariances shown here (ellipses represent one standard deviation). Thanks to Sean Escola for the code.

where $z(i)$ is the identity of the mixture component from which the i-th sample voltage trace was drawn. Ignoring constants, the expected complete log-likelihood is

$$\begin{aligned} E_{p(z|y,\theta')} \log p(y,z|\theta) &= E_{p(z|y,\theta')} \left(\sum_{i=1}^{N} \log p(z_i|\theta) + \sum_{i=1}^{N} \log p(y_i|z_i,\theta) \right) \\ = \sum_{j=0}^{J} \sum_{i=1}^{N} p(z(i)=j|\theta') \left[\log \alpha_{z(i)} - \frac{1}{2} \left(\log |C_{z(i)}| + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right) \right] \\ = \sum_{j=0}^{J} \sum_{i=1}^{N} p(z(i)=j|\theta') \log \alpha_{z(i)} - \frac{1}{2} \sum_{j=0}^{J} \sum_{i=1}^{N} p(z(i)=j|\theta') \left[\left(\log |C_{z(i)}| + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right) \right], \quad (5) \end{aligned}$$

with

$$\begin{aligned} p(z(i)=j|\theta') &= \frac{1}{Z} \alpha'_j \mathcal{N}_{\mu'_{z(j)}, C'_{z(j)}}(\vec{V}_i) \\ &= \frac{\exp \left[\log \alpha'_j - \frac{1}{2} \left(\log |C'_j| + (\vec{V}_i - \vec{\mu}'_j)^t C_j'^{-1} (\vec{V}_i - \vec{\mu}'_j) \right) \right]}{\sum_{j'=0}^{J} \exp \left[\log \alpha'_{j'} - \frac{1}{2} \left(\log |C'_{j'}| + (\vec{V}_i - \vec{\mu}'_{j'})^t C_{j'}'^{-1} (\vec{V}_i - \vec{\mu}'_{j'}) \right) \right]} \end{aligned}$$

denoting the conditional probability that the mixture component $z(i) = j$, under the parameters θ'.

Thus the E-step in this case corresponds to a probabilistic assignment of cluster identity to each sample i, given the parameters θ' from the previous iteration. In the M-step we have

to optimize $E_{p(z|y,\theta')} \log p(y, z|\theta)$ as a function of θ: since the objective function (5) is a sum of simpler, separable functions, we may optimize each summand independently. When we optimize

$$\sum_{j=0}^{J} \sum_{i=1}^{N} p(z(i) = j|\theta') \log \alpha_{z(i)}$$

as a function of α, under the constraint that $\sum_z \alpha_z = 1$ and $\alpha_z \geq 0$, we obtain the intuitive solution

$$\alpha_j^{new} = \frac{1}{N} \sum_{i=1}^{N} p(z(i) = j|\theta');$$

i.e., our updated mixture probability is just the average fraction of samples assigned to index j under the parameter setting θ'. Similarly, when we optimize

$$-\frac{1}{2} \sum_{i=1}^{N} p(z(i) = j|\theta') \left(\log |C_{z(i)}| + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right) = -\frac{1}{2} \sum_{i=1}^{N} w_{ij} \left(\log |C_{z(i)}| + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right)$$

with respect to $\vec{\mu}_j$ and C_j (where we have made the abbreviation $w_{ij} = p(z(i) = j|\theta')$ to simplify the notation in the formulas below), we obtain

$$\vec{\mu}_j^{new} = \frac{\sum_{i=1}^{N} w_{ij} \vec{V}_i}{\sum_{i=1}^{N} w_{ij}}$$

and

$$C_j^{new} = \frac{\sum_{i=1}^{N} w_{ij} (\vec{V}_i - \vec{\mu}_j)(\vec{V}_i - \vec{\mu}_j)^T}{\sum_{i=1}^{N} w_{ij}}.$$

This is a generalization of the standard maximum likelihood estimate for the mean and covariance of a multivariate Gaussian, where we have replaced the usual normalized sum over observed samples i with a normalized weighted sum, with the weights given by the probabilistic mixture assignments $w_{ij} = p(z(i) = j|\theta')$. As we will see below, this is a recurring feature of the EM algorithm: in the M-step, the usual loglikelihood (a sum over N terms) is replaced by a *weighted* sum over N terms, where the weights are derived from the conditional probabilities $p(z|y, \theta')$ computed in the E-step.

Note that it is straightforward to generalize to other mixture models. For example, (Shoham et al., 2003) argue that spike voltage waveforms are better modeled as a mixture of multivariate-t distributions, which have much fatter tails than the multivariate Gaussian and are therefore more robust to noise. Fitting the parameters of the multivariate-t (the M-step) proceeds much as in the multivariate Gaussian case, although unfortunately the nice analytic solutions for the mean and covariance parameters do not extend to the multivariate-t case (Peel and McLachlan, 2000; Shoham et al., 2003).

The spike sorting literature continues to expand, and a number of important problems remain open. A couple specific directions are worth noting. First, many extracellular recordings display a strong degree of nonstationarity, due for example to slow changes in the electrode position relative to the observed neurons (which in turn lead to changes in the size and, to a lesser degree, the shape of the observed spike waveforms). (Bar-Hillel et al., 2006) present an adaptive algorithm for tracking the cluster means and covariances as they evolve over the

course of the experiment; their method is based on a state-space framework that we will describe in more depth over the next few chapters (see also (Calabrese and Paninski, 2009)). A second important problem involves model selection: how do we choose the number of clusters J? Typically, J is selected "by eye" or by cross-validation approaches; more recently, (Wood and Black, 2008) presented Dirichlet process techniques that sidestep this issue somewhat, by effectively placing a prior over all possible values of J and then computing the posterior over the model parameters (including J) via MCMC methods.

A third key problem involves synchronous or near-synchronous firing, when spikes from different neurons occur within 1 ms of one another, "colliding" and invalidating our simple mixture model in which spikes occur without interacting with one another. This is a key problem in the setting of large-scale multi-electrode recordings, where a given neuron might contribute its voltage signal to many neighboring electrodes (Litke et al., 2004; Segev et al., 2004; Petrusca et al., 2007), leading to many potential collision sites. One effective method is to extend the mixture model to incorporate linear superpositions of spike waveforms (Lewicki, 1998; Sahani, 1999; Segev et al., 2004); electrical signals from different neurons combine linearly on the electrode to a very good approximation. We write the voltage signal as

$$V_t = \sum_{j,a} V_a^j(t - t_a^j) + \epsilon_t,$$

where $V_a^j(.)$ is the shape of the a-th voltage waveform snippet from neuron j, and t_a^j is the time of this event; ϵ_t denotes stochastic noise. In this setting the E step consists largely of determining the spike times t_a^j, while as usual the M-step involves an update of the properties of the distributions from which the waveforms $V_a^j(.)$ are drawn. It is much more difficult to evaluate the E-step exactly here without resorting to computationally expensive MCMC-based approaches, and therefore cheaper greedy computational methods are often employed instead: the simplest effective method is to add spikes one by one, choosing the neuron identity j and spike time t_a^j which will increase the log-posterior the most on each iteration, and stopping when any additional spikes will decrease the log-posterior. (The prior here encodes the number of spikes we expect each neuron to contribute; the likelihood, as usual, is derived from the assumed properties of the noise ϵ_t.) If the noise ϵ_t is taken to be Gaussian, these log-posterior computations reduce to linear filtering and template-matching operations under a quadratic loss function, and may be performed quite efficiently using fast Fourier techniques. See (Lewicki, 1998) for further discussion.

6 Example: Spike sorting given stimulus observations

It is natural to ask if we can increase the accuracy of the mixture model spike sorting method described above by including relevant "side" information (Ventura, 2008). For example, if we know that the neurons whose spikes we are attempting to classify are driven by some sensory stimulus, we can try to incorporate a model of the stimulus tuning in the mixture model. For simplicity, imagine that each neuron may be modeled as a Poisson process with rate $\lambda_z(t) = f(\vec{k}_z^T \vec{x}(t))$; we will also assume that all the observed waveforms i in fact correspond to true spikes — and furthermore that all the spikes have been detected — and therefore we may neglect the the noise unit $z = 0$. (This assumption is for notational simplicity only, and may be relaxed easily.) Instead of fixing the mixture weights α_z, here the "mixture weights" are effectively functions of time (depending on the value of the stimulus $\vec{x}$ at time t), where

we may define

$$\alpha_z(t) = \frac{f(\vec{k}_z^T \vec{x}(t))}{\sum_{j=1}^J f(\vec{k}_j^T \vec{x}(t))}.$$

Thus our parameter θ here is given by $\theta = \{(\vec{\mu}_z, C_z, \vec{k}_z)\}$, and we modify our complete loglikelihood

$$\log p(y, z|X, \theta)$$
$$= \sum_{i=1}^N \left[\log f(\vec{k}_z^T \vec{x}(t_i)) - \frac{1}{2} \left(\log\det(C_{z(i)}) + (\vec{V}_i - \vec{\mu}_{z(i)})^t C_{z(i)}^{-1} (\vec{V}_i - \vec{\mu}_{z(i)}) \right) \right] - \sum_{j=1}^J \int_0^T f(\vec{k}_j^T \vec{x}(t)) dt + const.,$$

where t_i denotes the i-th spike time.

The E-step proceeds as before, except now we have to incorporate the model's predicted firing rates into the definition of the "mixture assignments"

$$p(z(i) = j|X, \theta') = \frac{\exp\left[\log f\left(\vec{x}(t_i)^T \vec{k}_j' \right) - \frac{1}{2} \left(\log\det(C_j') + (\vec{V}_i - \vec{\mu}_j')^T C_j'^{-1} (\vec{V}_i - \vec{\mu}_j') \right) \right]}{\sum_{j'=1}^J \exp\left[\log f\left(\vec{x}(t_i)^T \vec{k}_{j'}' \right) - \frac{1}{2} \left(\log\det(C_{j'}') + (\vec{V}_i - \vec{\mu}_{j'}')^T C_{j'}'^{-1} (\vec{V}_i - \vec{\mu}_{j'}') \right) \right]}.$$

Now the M-step for updating the means and covariances $\vec{\mu}_z$ and C_z proceeds exactly as before, using the $p(z(i) = j|X, \theta')$ computed above. We replace the M-step for the mixture weights α_z with an analogous M-step for $\vec{k}_z$: for each $1 \le j \le J$ we maximize

$$\sum_{i=1}^N p(z(i) = j|\theta') \log f(\vec{k}_j^T \vec{x}(t_i)) - \int_0^T f(\vec{k}_j^T \vec{x}(t)) dt.$$

Note that each of these optimizations may be computed independently (unlike in the M-step for $\vec{\alpha}$, where the constraint $\sum \alpha_z = 1$ served to couple the optimizations together); moreover, note that each optimization is concave, effectively a weighted generalization of our usual point process likelihood optimization problem.

Of course, we may generalize this model significantly. One important extension is to incorporate the fact that sequential spike waveforms V_i are not in fact independent (for example, partial inactivation of sodium channels following a spike may reduce the peak voltage of a second spike observed shortly after the first). See, e.g., (Pouzat et al., 2004) for details of a more sophisticated (but much more computationally expensive) MCMC-based approach that incorporates some of these effects.

7 Example: Generalized linear point-process models with spike-timing jitter

Let's turn our attention to single-neuron data, where we are confident that we have isolated and classified each spike correctly. Again, we assume the true (unobserved) spike trains $\{t_i\}$ are generated via an LNP model with conditional intensity function $\lambda(t) = f(X_t \cdot \theta)$. Now, however, let us imagine that each spike time is subject to some random jitter (e.g., due to timing variability in the detection of the spike waveform due to noisy threshold crossing in low-SNR extracellular recordings, or variability in the speed of axonal propagation of the

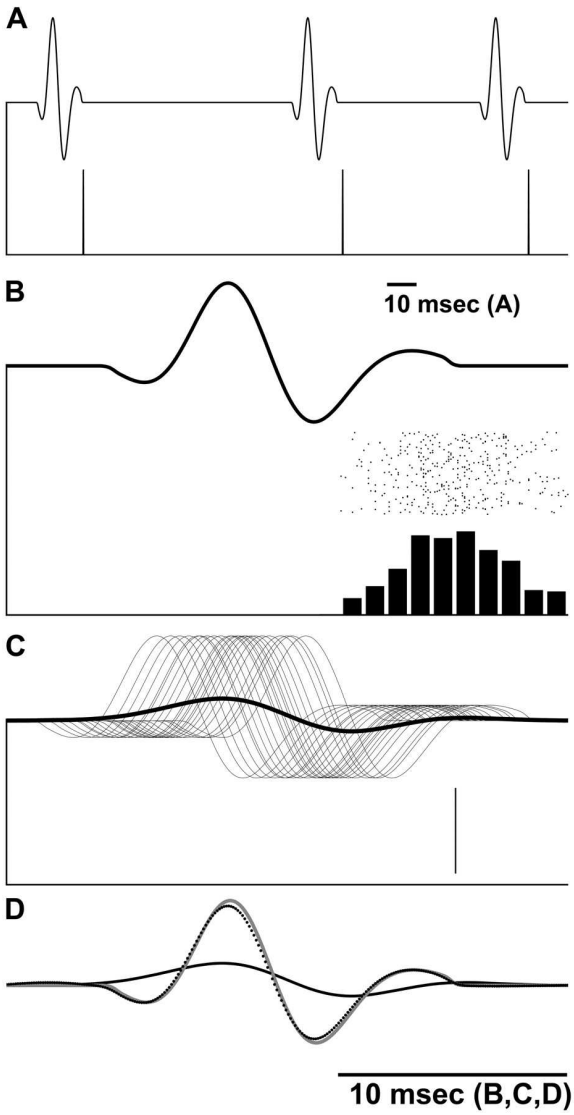

Figure 4: Illustration of the LNP-jitter model (Fig. 1 of (Aldworth et al., 2005)). **A**: responses of a simulated neuron to three identical stimulus presentations; each stimulus elicits a response which is slightly jittered in time; responses are tabulated in PSTH form in panel **b**. **C**: Computing the spike-triggered average stimulus (solid trace) results in a much smoother, smaller waveform than the true stimulus (gray traces), since we have effectively convolved the true stimulus with the Gaussian spiking distribution shown in panel b. **D**: Illustration of the "dejittered STA" applied to this simulated data: see (Aldworth et al., 2005) for details.

action potential due to stochastic ion channel fluctuations (Faisal and Laughlin, 2007)): more concretely, imagine that the observed spike train $\{u_i\}$ is produced by jittering each true spike time t_i independently by Gaussian noise with mean zero and variance σ^2,

$$u_i = t_i + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma^2).$$

Related models have been considered by (Aldworth et al., 2005; Chang et al., 2005; Dimitrov and Gedeon, 2006; Gollisch, 2006); see also (Dimitrov et al., 2009) for a generalization to spatial jitters applied to visual receptive fields, and (Amarasingham et al., 2005) for a different approach to analyzing jitter in spike trains. We want to fit the parameters (θ, σ^2) by EM.

The complete log-likelihood for this model may be derived along the same lines as discussed in the last two sections: we obtain

$$\log p(\{t_i\}, \{u_i\}|\theta) \approx \sum_i \log f(X_{t_i} \cdot \theta) - \int_0^T f(X_s \cdot \theta) ds - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_i (t_i - u_i)^2,$$

where N denotes the total number of spikes. Here we have made the approximation that the observed spikes u_i are far enough apart (measured in units of σ) that we may neglect the probability that the observed spike u_i is actually due to the true spike t_j, with $i \neq j$.

The expected complete log-likelihood under $(\hat{\theta}_j, \hat{\sigma}_j^2)$, the estimated parameters after j EM iterations, may be computed as

$$\begin{aligned} Q &= \int p(\{t_i\}|\{u_i\}, \hat{\theta}_j, \hat{\sigma}_j^2) \log p(\{t_i\}, \{u_i\}|\theta, \sigma^2) d\{t_i\} \\ &\approx \sum_i \int_0^T r_i(t) \log f(X_t \cdot \theta) dt - \int_0^T f(X_s \cdot \theta) ds - \frac{N}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_i \int_0^T r_i(t)(t - u_i)^2 dt, \end{aligned}$$

where

$$p(t_i|u_i, X, \hat{\theta}_j, \hat{\sigma}_j^2) \approx \frac{e^{(t-u_i)^2/2\hat{\sigma}_j^2} f(X_t \cdot \hat{\theta}_j)}{\int_0^T e^{(t-u_i)^2/2\hat{\sigma}_j^2} f(X_t \cdot \hat{\theta}_j) dt} \equiv r_i(t)$$

denotes the conditional density that the true i-th spike time is given by t, given the observed time u_i, stimulus X, and current parameters $\hat{\theta}_j, \hat{\sigma}_j^2$. Computing the functions $r_i(t)$ constitutes the E step; the M step may be performed easily, by noting that, as in the last example, since $r_i(t) \geq 0$,

$$\sum_i \int_0^T r_i(t) \log f(X_t \cdot \theta) dt - \int_0^T f(X_s \cdot \theta) ds$$

is a concave function of θ under the usual convexity and log-concavity conditions on $f(.)$; i.e., once again, in the M-step we optimize a weighted average version of the usual point-process likelihood. Thus we set

$$\hat{\theta}_{j+1} = \arg\max_\theta \sum_i \int_0^T r_i(t) \log f(X_t \cdot \theta) dt - \int_0^T f(X_s \cdot \theta) ds$$

and

$$\hat{\sigma}_{j+1}^2 = \frac{1}{N} \sum_i \int_0^T r_i(t)(t - u_i)^2 dt.$$

Once the parameter estimates $(\hat{\theta}_j, \hat{\sigma}_j^2)$ have converged, we may use the corresponding $r_i(t) = p(t_i|u_i, X, \hat{\theta}_j, \hat{\sigma}_j^2)$ to perform denoising, that is, to estimate and subtract out the jitter ϵ_i (Aldworth et al., 2005). We may, for example, estimate the true spike time t_i as

$$\hat{t}_i = E(t_i|u_i, X, \hat{\theta}_j, \hat{\sigma}_j^2) \approx \int r_i(t) t dt.$$

See (Aldworth et al., 2005) for applications and further discussion.

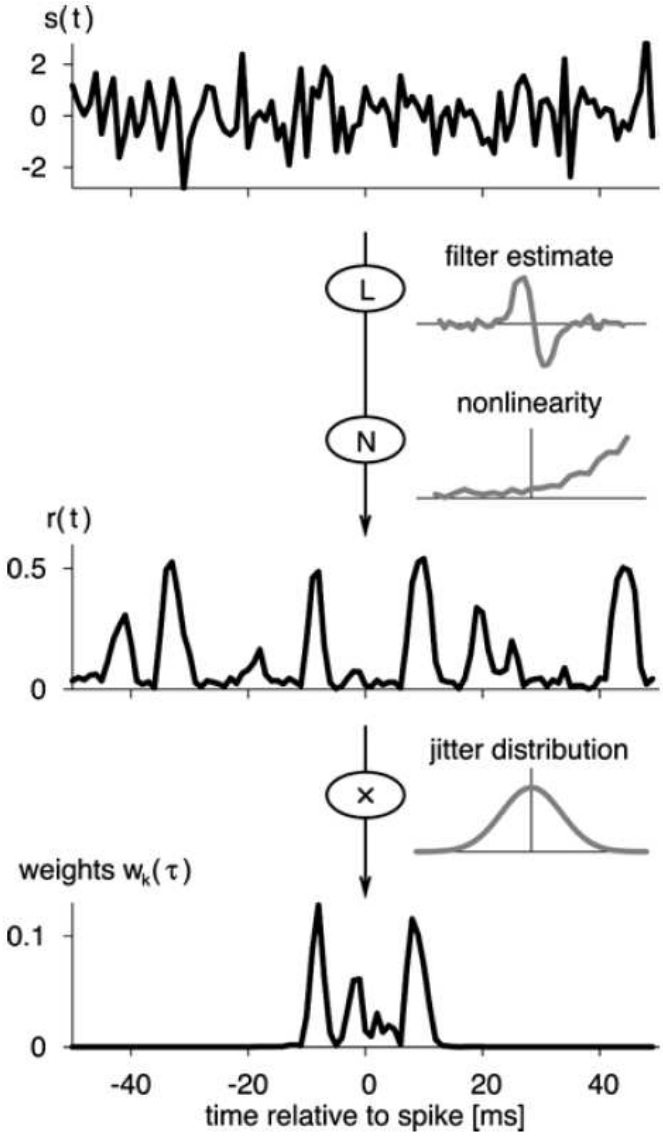

Figure 5: Illustration of the E-step in the LNP-jitter model (Fig. 2 of (Gollisch, 2006)). Given the stimulus $s(t)$, we form the rate $r(t)$ predicted under the LNP model and then multiply this by a Gaussian centered at the observed spike time t_k and normalize to obtain the posterior $p(u_k|t_k, \theta, \{s(t)\})$ (note that Gollisch's notation is different from ours).

8 Example: Fitting hierarchical generalized linear models for spike trains

Let's assume we have recorded from a number of neurons in a given brain area. We might expect many of these neurons to have similar tuning characteristics. Indeed, we would like to exploit any such similarities (for example, the more we know about a brain area *a priori*, the easier it should be to estimate the receptive fields of any new neurons we enounter in this area) and to quantify the heterogeneity of tuning properties in a given brain area, cortical

layer, etc. This quantification of "functional anatomy" — how the functional properties of neurons vary as a function of anatomy — is, of course, one of the major goals of systems neuroscience; see, e.g., (Ringach et al., 2002; Ringach, 2002) for examples of these analyses in primate primary visual cortex.

One simple way of exploiting these similarities in tuning properties is as follows. Imagine for simplicity that each neuron in the brain region of interest may be modeled with our standard GLM, $\lambda_i(t) = f(X_t^T \theta_i)$. Now further assume that the model parameters θ_i are themselves sampled in an i.i.d. manner from some distribution $p(\theta)$, whose properties we would like to quantify: e.g., how variable is θ? What do "typical" parameter values θ look like (i.e., what is $E(\theta)$)? In order to summarize these quantities more efficiently, we might parameterize $p(\theta) = p(\theta|\Gamma)$, where Γ denotes the "hyperparameters" which specify the shape of $p(\theta)$: for example, if we take $p(\theta)$ to be Gaussian, then Γ would summarize the mean and covariance of θ (Behseta et al., 2005).

This kind of multi-stage model — where the observed data depend on some unknown parameters whose distribution in turn depends on some unknown hyperparameters — is known as a "hierarchical" model in the statistics literature. Given the hyperparameters Γ, fitting the individual parameters θ_i is fairly straightforward: we simply compute the posterior

$$\log p(\theta_i|D, \Gamma) = \log p(D_i|\theta_i) + \log p(\theta_i|\Gamma) + const.,$$

where D_i denotes the data recorded from the i-th neuron in the dataset (we assume that these $\{D_i\}$ are conditionally independent given $\{\theta_i\}$). For example, we could compute $\hat{\theta}_{i,MAP} = \arg\max_{\theta_i} \log p(\theta_i|D, \Gamma)$, for each θ_i. The nice thing about this model is that the more we learn about Γ (e.g., by observing more neurons), the better our estimates of θ_i will be; thus we can in effect "share" information between (conditionally) independent experiments D_i.

To fit Γ, on the other hand, we may derive an EM algorithm, treating θ_i as latent variables. We want to maximize the marginal loglikelihood

$$p(D|\Gamma) = \int p(D, \vec{\theta}|\Gamma) d\vec{\theta} = \int \prod_i p(D_i|\theta_i) p(\theta_i|\Gamma) d\vec{\theta}.$$

Computing this integral directly is often infeasible if the number of neurons i is large (although if our usual Gaussian approximation

$$p(D_i, \theta_i|\Gamma) = p(D_i|\theta_i) p(\theta_i|\Gamma) \approx w_i G_{\mu_i, C_i}(\theta_i)$$

is accurate, then we can in fact compute this integral analytically given (w_i, μ_i, C_i) (Sahani and Linden, 2003; Ahrens et al., 2008)). To derive the EM algorithm here, we write down the expected complete log-likelihood as usual:

$$\begin{aligned} E_{p(\vec{\theta}|D,\Gamma^{(j)})} \log p(\vec{\theta}, D|\Gamma) &= E_{p(\vec{\theta}|D,\Gamma^{(j)})} \sum_i \big(\log p(D_i|\theta_i) + \log p(\theta_i|\Gamma) \big) \\ &= \sum_i E_{p(\theta_i|D_i,\Gamma^{(j)})} \big(\log p(D_i|\theta_i) + \log p(\theta_i|\Gamma) \big). \end{aligned}$$

At this point, we need to introduce a concrete model for $p(\theta|\Gamma)$; for simplicity, we take θ to be Gaussian with mean μ and covariance C (i.e., $\Gamma = (\mu, C)$), although as usual we emphasize that other model choices are feasible here. Thus, plugging in the Gaussian density

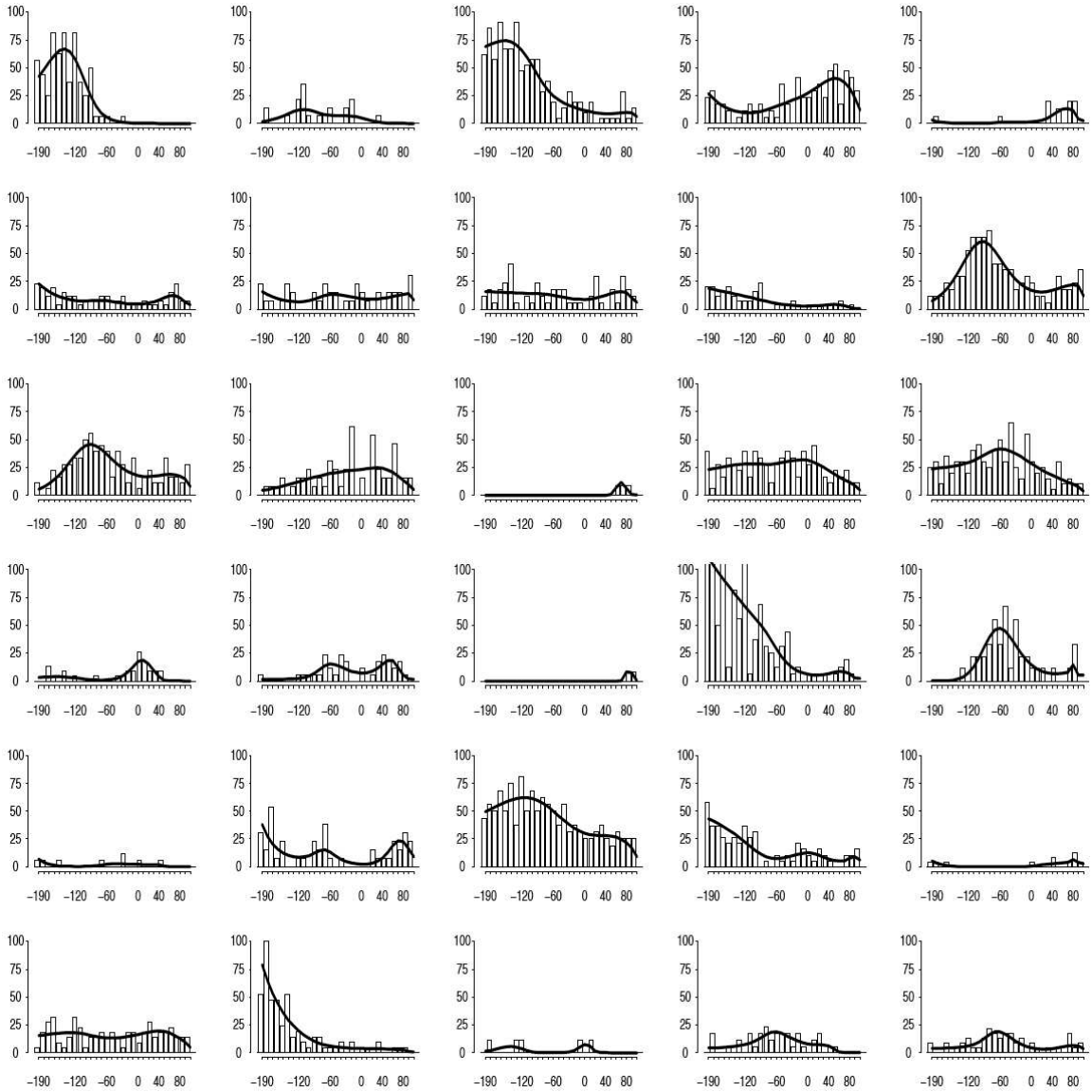

Figure 6: Normalized histograms displaying observed firing rates for 30 neurons recorded from the primary motor cortex of a primate performing a sequential pointing task (Behseta et al., 2005). Solid traces indicate estimated firing-rate curves obtained from a hierarchical Bayesian model fit. Horizontal axes run from 200 ms before the monkey touched the target to 100 ms after; vertical axes from 0 to 100 spikes per second. In this case, the spike count per bin was modeled as inhomogeneous Poisson; the prior on the time-varying rate function was specified in terms of a mixture of Gaussian processes formed by adding together a collection of spline functions with random (Gaussian) coefficients, with an additional hierarchical prior on the spline knots. The firing rates here were estimated by a hierarchical MCMC method which integrated over the unobserved Gaussian spline latent functions; see (Behseta et al., 2005) for further details.

for $\log p(\theta_i|\Gamma)$ (and dropping the $\log p(D_i|\theta_i)$ term, which does not depend on Γ), we see that to optimize the expected log-likelihood with respect to μ and C we need to optimize

$$\sum_i E_{p(\theta_i|D_i,\Gamma^{(j)})}\left(-\frac{1}{2}\left(\log\det(C) + (\theta_i-\mu)^T C^{-1}(\theta_i-\mu)\right)\right);$$

we obtain

$$\mu^{(j+1)} = \frac{1}{N}\sum_{i=1}^{N} E(\vec{\theta}_i|D_i,\mu^{(j)},C^{(j)})$$

and

$$C^{(j+1)} = \frac{1}{N}\sum_{i=1}^{N}\left(Cov(\vec{\theta}_i|D_i,\mu^{(j)},C^{(j)}) + (\mu_i-\mu)(\mu_i-\mu)^T\right),$$

where μ_i abbreviates $E(\vec{\theta}_i|D_i,\mu^{(j)},C^{(j)})$ and N denotes the number of neurons in the data set. As usual, these are fairly intuitive gneralizations of what we would expect to see in the fully-observed setting: the one possibly unfamiliar piece is the update for C, in which we have to combine both the variability in the observed μ_i vectors and also the residual uncertainties $Cov(\vec{\theta}_i|D_i,\mu^{(j)},C^{(j)})$.

The only remaining ingredient is the E-step, in which we obtain the means μ_i and covariances $Cov(\vec{\theta}_i|D_i,\mu^{(j)},C^{(j)})$. We have

$$\begin{aligned} p(\theta_i|D_i,\mu^{(j)},C^{(j)}) &= \frac{1}{Z}p(D_i|\theta)p(\theta_i|\mu^{(j)},C^{(j)}) \\ &= \frac{1}{Z}\left(\prod_t e^{-f(\theta_i^T X_t)dt} f(\theta_i^T X_t)^{n_i(t)}\right) e^{-\frac{1}{2}(\theta_i-\mu^{(j)})^T (C^{(j)})^{-1}(\theta_i-\mu^{(j)})}; \end{aligned}$$

the first and second moments of these posterior distributions may be approximated via any of our standard tools (Laplace approximation, expectation propagation, Monte Carlo, etc.). See (Behseta et al., 2005) (and Fig. 6) for details of a Monte Carlo implementation of a closely related model (though note that (Behseta et al., 2005) employed a "fully Bayesian" approach — i.e., the parameters were integrated out, instead of maximized, as in the EM method described here).

9 Example: Latent-variable models of overdispersion and common-input correlations in spike counts

We have spent a great deal of time discussing generalized linear models for spike trains. For concreteness, let's consider our usual model, in which the spike count n_i is Poisson distributed, with rate $f(X_i\theta)dt$. The only source of variability in this model, given the covariate X and the parameter θ, is in the Poisson response. Of course, in reality there are many factors influencing neuronal variability, and it seems overly crude to lump all of these effects together in this single Poisson output distribution (Brockwell et al., 2007). In addition, the Poisson assumption ties together the mean and variance of the response (since the mean and variance of n_i are equal given $X_i\theta$), which greatly reduces the flexibility of this model (Shoham et al., 2005). Finally, it has been argued (Nykamp, 2005; Nykamp, 2007; Kulkarni and Paninski, 2007; Vidne et al., 2009; Yu et al., 2009b; Santhanam et al., 2009) that the GLM framework

inappropriately models correlations in neural populations: recall that in the multineuronal GLM setting, any observed correlations are modeled with spike-coupling terms that reflect direct connections between the subset of neurons that happened to be observed during a given experiment. Instead, "common input" (from the majority of neurons that we do not observe during any typical experiment) will typically play a much more important role in determining correlation structure.

Thus, it seems reasonable to include an additional source of variability in our basic GLM. As a first step, let's consider a model of the form

$$n_i \sim Poiss[\exp(X_i\theta_i + \epsilon_i)dt], \tag{6}$$

where we have introduced the latent variable ϵ and chosen $f(.) = \exp(.)$, for concreteness. (This is a special case of what is known as a "random effects" model in the statistics literature.) For simplicity, let the latent variable be Gaussian, $\epsilon \sim \mathcal{N}_{\mu,C}$. After a rescaling, this model can be expressed in the simpler form

$$n_i \sim Poiss[\exp(X_i\theta_i + \epsilon_i)];$$

we have simply shifted the mean of ϵ by $\log dt$ here.

To derive an EM algorithm for μ and C, we proceed exactly as in the previous section; the latent-Gaussian and observed-Poisson structure of these models are mathematically equivalent. (The EM approach turns out to be relatively inefficient for the estimation of θ_i; we will address this issue in more depth below.)

Let's turn now to the question of initialization of our estimates $\hat{\mu}$ and $\hat{C}$ in this model. One simple approach is to use the method of moments: we observe the empirical values of moments such as $\widehat{E(n_i)}, \widehat{Cov(\vec{n})}$, and $\widehat{E(X_i n_i)}$, and set the paramters (θ, μ, C) so that the expectations of these functions of the data given (θ, μ, C) match the observed values: i.e., choose $\widehat{(\theta, \mu, C)}$ as the solution to the equations $E(n_i|\theta, \mu, C) = \widehat{E(n_i)}$, $Cov(n_i|\theta, \mu, C) = \widehat{Cov(n_i)}$, and so on.

We start by initializing θ. It turns out that a good deal of our standard theory for estimating θ still applies here. Let's examine the marginal firing rate

$$\lambda(z) = E_\epsilon f(z + \epsilon) = \int p(\epsilon) f(z + \epsilon) d\epsilon,$$

where z abbreviates $X\theta$. Now it is easy to show, using standard properties of convex and log-concave functions (Paninski, 2005), that if $f(z)$ is a convex, log-concave, increasing function of z, then so is $E_\epsilon f(z + \epsilon)$. This in turn implies that if the distribution of the covariate X_i is elliptically symmetric, then we may consistently estimate θ_i via either the standard covariance-adjusted spike-triggered average (Paninski, 2003) (as discussed in an earlier chapter) or by the maximum likelihood estimator for the GLM parameter, using the (incorrect) nonlinearity $f(.)$ to compute the likelihood (Paninski, 2004), even when the correct values of μ and C are unknown (again, recall the discussion in the GLM chapter). In either case, consistency holds up to a scalar constant; i.e., given enough data, either of our standard STA or GLM methods produces an estimate $\hat{\theta}_i$ which converges to a scalar multiple of θ_i.

So, given this initial estimate for the vector parameters $\{\theta_i\}$, we need to estimate μ, C, and a set of scalar gain factors $\{a_i\}$ corresponding to each vector parameter $\{\theta_i\}$. We start

with $\{a_i\}$. First, we write down the moments

$$\begin{aligned} E(n_i) &= E_{z,\epsilon} \exp(a_i z_i + \epsilon_i) \\ &= E_{z_i} \exp(a_i z_i) E_{\epsilon_i} \exp(\epsilon_i) \\ &= M_i(a_i) r_i. \end{aligned}$$

and

$$\begin{aligned} E(z_i n_i) &= E_{z,\epsilon} \left[\exp(a_i z_i + \epsilon_i) z_i\right] \\ &= E_{z_i} \left[\exp(a_i z_i) z_i\right] E_{\epsilon_i} \exp(\epsilon_i) \\ &= \frac{d}{da_i} M_i(a_i) r_i, \end{aligned}$$

where we have used the independence of z_i and ϵ_i and made the abbreviations $M_i(a_i)$ for the moment-generating function

$$M_i(a_i) = E[\exp(a_i z_i)]$$

and r_i for the Gaussian expectation

$$r_i = E[\exp(\epsilon_i)] = \exp\left(\mu_i + \frac{1}{2} C_{ii}\right).$$

Note that $M_i(a_i)$ may be computed directly from the known distribution of z; for example, in the case that z is Gaussian, $z_i \sim \mathcal{N}_{\mu_z, \sigma_z^2}$, we may compute directly $M_i(a) = \exp(a\mu_z + a^2\sigma_z^2/2)$. Now we see that we may simply divide these two equations to find that

$$\frac{E(z_i n_i)}{E(n_i)} = \frac{dM_i(a_i)/da_i}{M_i(a_i)} = \frac{d}{da_i} \log M_i(a_i).$$

This equation has a unique solution in a_i, due to the strict convexity of the cumulant-generating function $\log M_i(a_i)$ (Schervish, 1995); given a_i, we see that we have a unique solution for r_i as well.

To obtain C (and therefore μ, given r_i), we use the standard formula for the variance of a mixture:

$$Cov(\vec{n}) = E_{z,\epsilon} Cov(\vec{n}|z,\epsilon) + Cov_{z,\epsilon} E(\vec{n}|z,\epsilon) \tag{7}$$

We evaluate each term in turn:

$$E_{z,\epsilon} Cov(\vec{n}|z,\epsilon) = E_{z,\epsilon} \mathrm{diag}[\exp(a_i z_i + \epsilon_i)] = \mathrm{diag}[E(n_i)], \tag{8}$$

and

$$\begin{aligned} \left[Cov_{z,\epsilon} E(\vec{n}|z,\epsilon)\right]_{ij} &= E_{z,\epsilon}\left[E(n_i|z_i,\epsilon_i) E(n_j|z_j,\epsilon_j)\right] - E_{z,\epsilon}\left[E(n_i|z_i,\epsilon_i)\right] E_{z,\epsilon}\left[E(n_j|z_j,\epsilon_j)\right] \\ &= E_{z,\epsilon}\left[\exp(a_i z_i + \epsilon_i) \exp(a_j z_j + \epsilon_j)\right] - E_{z,\epsilon}\left[\exp(a_i z_i + \epsilon_i)\right] E_{z,\epsilon}\left[\exp(a_j z_j + \epsilon_j)\right] \\ &= E_z\left[\exp(a_i z_i + a_j z_j)\right] E_\epsilon\left[\exp(\epsilon_i + \epsilon_j)\right] - E(n_i) E(n_j) \\ &= M_{ij}(a_i, a_j) \exp(\mu_i + \mu_j + \frac{1}{2}(C_{ii} + C_{jj} + 2C_{ij})) - M_i(a_i) r_i M_j(a_j) r_j \\ &= M_{ij}(a_i, a_j) r_i r_j \exp(C_{ij}) - M_i(a_i) M_j(a_j) r_i r_j \\ &= r_i r_j \left[M_{ij}(a_i, a_j) \exp(C_{ij}) - M_i(a_i) M_j(a_j)\right], \end{aligned} \tag{9}$$

where we have introduced an additional abbreviation for the pairwise moment-generating function

$$M_{ij}(a_i, a_j) = E_z[\exp(a_i z_i + a_j z_j)].$$

Now, given a_i and r_i, equation (9) gives us a unique solution for C_{ij}, and therefore μ_i as well. Note that in practice there is no guarantee that the resulting estimate $\hat{C}$ is positive semidefinite. (Macke et al., 2009) discuss some approaches to project C onto the set of positive semidefinite matrices; one crude but straightforward approach is simply to set any negative eigenvalues of $\hat{C}$ to zero, or alternately to add $-\lambda_{min} I$ to $\hat{C}$, where λ_{min} is the bottom eigenvalue of $\hat{C}$.

Thus, to summarize, we can compute the first and second moments in model (6) explicitly as a function of the model parameters $(\{a_i\}, \mu, C)$, and by inverting this function we obtain a good initialization for these parameters. Similar computations (involving correlated Gaussian random vectors mapped through a nonlinear function) have been exploited in a number of recent papers (Dorn and Ringach, 2003; de la Rocha et al., 2007; Krumin and Shoham, 2008; Macke et al., 2009); see also (Niebur, 2007) for a different approach to sampling spike counts with a predefined correlation structure.

It is also worth noting that equations (7) and (8) together imply that

$$V(n_i) = [Cov(\vec{n})]_{ii} \geq E(n_i),$$

i.e., the spike count variance is always at least as large as the mean in this model, consistent with the overdispersion observed in spike counts in visual cortex (Tolhurst and Dean, 1983; Softky and Koch, 1993; Shadlen and Newsome, 1998).

Another interesting application is described in (Santhanam et al., 2009; Yu et al., 2009a). The basic idea is that, if the covariance matrix of the latent variable $\vec{\epsilon}$ is of low rank, then equation (6) can be interpreted as a generalized factor analysis model (Tipping and Bishop, 1999): our observations may be thought of as noise-contaminated samples from a distribution which is supported on a low-dimensional subspace spanned by $X_i\theta_i$ plus the eigenvectors of the hidden covariance matrix C. (Santhanam et al., 2009; Yu et al., 2009a) discuss how related models can be used to reduce the effective dimensionality of the observed count vector $\vec{n}$, using approximate EM methods to fit the parameters of the hidden random vector $\vec{\epsilon}$.

So far, we have ignored the question of how best to model temporal correlations in the spike counts n_i; we will address this important issue, and discuss efficient maximum-likelihood methods for estimating θ in the presence of temporally-correlated noise, in the chapter on state-space techniques.

10 Example: Iterative proportional fitting

Kass will fill in something here...

11 The E-step may be used to compute the gradients of the marginal likelihood

It turns out that the problem of computing the gradients of the likelihood in the latent-variable setting is intimately related to the E-step, as can be seen by simply expanding these

gradients via Bayes' rule (Salakhutdinov et al., 2003):

$$\begin{aligned}
\nabla_\theta \log p(y|\theta)\Big|_{\theta=\theta'} &= \nabla_\theta \left(\log \int p(y,z|\theta) dz \right) \Big|_{\theta=\theta'} \\
&= \frac{1}{p(y|\theta')} \nabla_\theta \left(\int p(y,z|\theta) dz \right) \Big|_{\theta=\theta'} \\
&= \int \frac{p(y,z|\theta')}{p(y|\theta')} \nabla_\theta \log p(y,z|\theta) \Big|_{\theta=\theta'} dz \\
&= \int p(z|y,\theta') \nabla_\theta \log p(y,z|\theta) \Big|_{\theta=\theta'} dz \\
&= \nabla_\theta Q(\theta,\theta') \Big|_{\theta=\theta'},
\end{aligned}$$

assuming that the necessary interchanges of derivative and integral may be justified. Since $\int p(z|y,\theta') \nabla_\theta \log p(y,z|\theta) dz$ is typically easy to compute once the E-step is complete, we may therefore compute the gradient fairly easily. We may observe this effect graphically in Fig. 2: since the auxiliary function Q meets the objective function L at θ, and since the auxiliary function is a lower bound on the marginal likelihood, if both functions are continuously differentiable then it is geometrically clear that the gradients of these functions must match at θ. (But note that the second derivatives typically do not match, as discussed below.) This relationship between the EM and gradient-based updates may be exploited to develop more efficient hybrid optimization algorithms (Salakhutdinov et al., 2003). We will see a variety of examples in the next few chapters.

12 The convergence rate of the EM algorithm depends on the "ratio of missing information"

As we mentioned above, EM generally takes more steps to converge than does direct Newton-Raphson maximization of the marginal loglikelihood (Meng and Rubin, 1991; Salakhutdinov et al., 2003). (Of course, the point of the EM trick is that in many cases it may be much easier to compute an EM step than a direct Newton step on the marginal loglikelihood.) The EM convergence rate turns out to depend critically on a natural measure of the information contained in the missing data, as we explain now.

To begin, let's take a closer look at Newton's method here; we iterate

$$\theta^{(i+1)} = \theta^{(i)} - H_{\theta^{(i)}}^{-1} \nabla_{\theta^{(i)}},$$

where H_θ and ∇_θ denote the Hessian and gradient, resepctively, of the marginal loglikelihood evaluated at θ. (Of course, it may be difficult to compute H_θ in practice, but in this section we will ignore these highly problem-dependent computational issues, to instead focus on the convergence rate as a function of iteration count, not wall clock time.) Let's subtract the optimal value θ_0 from both sides to get a sense of how the error $\theta^{(i)} - \theta_0$ behaves as a function of the iteration i:

$$\theta^{(i+1)} - \theta_0 = \theta^{(i)} - \theta_0 - H_{\theta^{(i)}}^{-1} \left(\nabla_{\theta^{(i)}} - \nabla_{\theta_0} \right).$$

(We have also used the fact that $\nabla_{\theta_0} = 0$.) Now we massage this a bit:

$$\begin{aligned}
\theta^{(i+1)} - \theta_0 &= \theta^{(i)} - \theta_0 - H_{\theta^{(i)}}^{-1} \left(\nabla_{\theta^{(i)}} - \nabla_{\theta_0}\right) \\
&= \theta^{(i)} - \theta_0 - H_{\theta^{(i)}}^{-1} \left(H_{\theta_0}(\theta^{(i)} - \theta_0) + O(||\theta^{(i)} - \theta_0||_2^2)\right) \\
&= \theta^{(i)} - \theta_0 - \left(H_{\theta_0} + O(||\theta^{(i)} - \theta_0||_2)\right)^{-1} \left(H_{\theta_0}(\theta^{(i)} - \theta_0) + O(||\theta^{(i)} - \theta_0||_2^2)\right) \\
&= \theta^{(i)} - \theta_0 - \left(\theta^{(i)} - \theta_0\right) + O(||\theta^{(i)} - \theta_0||_2^2) \\
&= O(||\theta^{(i)} - \theta_0||_2^2).
\end{aligned}$$

We have used a Taylor expansion of ∇_θ in the second line, and of H_θ in the third line (i.e., we have assumed that the marginal loglikelihood is sufficiently smooth to justify this expansion), and assumed that H_{θ_0} is negative definite. We conclude that Newton's method converges *quadratically* under these assumptions: once we are sufficiently close to the true optimum θ_0 that our Taylor approximations are accurate, each Newton iteration effectively squares the precision of our estimate (i.e., if $||\theta^{(i)} - \theta_0||_2 = 10^{-x}$, where x is the number of significant figures, then $||\theta^{(i+1)} - \theta_0||_2 \approx 10^{-2x}$, doubling our significant digits).

Now each EM step, at least locally, can be written in very similar form. If we are sufficiently close to the optimizer θ_0, then each M-step can be approximated with just a single Newton-step to optimize the auxiliary function $Q(\theta, \theta')$:

$$\theta^{(i+1)} = \theta^{(i)} - H_i^{-1} \nabla_{\theta^{(i)}}.$$

Here $\nabla_{\theta^{(i)}}$ is exactly as in the Newton step (recall our discussion of the gradients in the preceding section), but the Hessian term is different: $H_i = \nabla\nabla_\theta Q(\theta, \theta^{(i)})|_{\theta=\theta^{(i)}}$. Now, as discussed above, since Q is a lower bound on the marginal loglikelihood, we know that $H_i \leq H_{\theta^{(i)}}$ (in the sense that $H_{\theta^{(i)}} - H_i$ is positive semidefinite); thus, qualitatively, each M-step entails a smaller step-size than does a Newton step on the full marginal loglikelihood. Quantitatively, if we repeat the steps of our derivation of the Newton convergence rate above, we arrive at

$$\begin{aligned}
\theta^{(i+1)} - \theta_0 &= \theta^{(i)} - \theta_0 - H_i^{-1} \left(H_{\theta_0}(\theta^{(i)} - \theta_0) + O(||\theta^{(i)} - \theta_0||_2^2)\right) \\
&= \left(I - H_i^{-1} H_{\theta_0}\right)(\theta^{(i)} - \theta_0) + O(||\theta^{(i)} - \theta_0||_2^2)
\end{aligned}$$

in the EM case. Thus, if $H_{\theta^{(i)}} - H_i$ is positive definite, then

$$\theta^{(i+1)} - \theta_0 \approx \left(I - H_i^{-1} H_{\theta_0}\right)(\theta^{(i)} - \theta_0)$$

to the highest order, entailing a qualitatively slower, *linear* rate of convergence: we only add a fixed number of digits of precision with each EM iteration, instead of doubling our significant digits with each full Newton iteration.

Now, qualitatively speaking, the smaller the matrix $(I - H_i^{-1} H_{\theta_0})$, the faster EM converges. Since H_i converges to $H_\infty \equiv \left(\nabla\nabla_\theta Q(\theta, \theta_0)|_{\theta=\theta_0}\right)^{-1} H_{\theta_0}$, we may summarize the local convergence rate with a single matrix, $(I - H_\infty^{-1} H_{\theta_0})$. If we rearrange this slightly,

$$I - H_\infty^{-1} H_{\theta_0} = H_\infty^{-1}(H_\infty - H_{\theta_0}),$$

we can see that this matrix has a natural interpretation as a ratio of *missing information*: $-(H_\infty - H_{\theta_0})$ is the Fisher information we have lost by not directly observing the missing

data z. (Remember, H_{θ_0} is the Hessian of the full marginal loglikelihood at θ_0, and H_∞ is the Hessian of our lower bound on the marginal loglikelihood at the same point.) In the limit of small missing information, $(H_\infty - H_{\theta_0}) \to 0$, we recover the quadratic convergence rate of the full Newton algorithm, whereas in the limit of large missing information, EM converges very slowly and Newton-type methods (including conjugate-gradient methods, with the gradient computed as discussed in the preceding section) become much more attractive (Salakhutdinov et al., 2003).

We have emphasized that the curvature of the auxiliary function $Q(.)$ is sharper than the curvature of the marginal loglikelihood function, which leads to a smaller effective step-size in the EM algorithm than in the full Newton algorithm. Another important consequence is that the limiting inverse curvature $-H_\infty^{-1}$ of $Q(\theta, \theta_0)$ generally underestimates our uncertainty about $\hat{\theta}$; remember, the standard Laplace approximation says that $Cov(\hat{\theta}) \approx -H_{\theta_0}^{-1}$, and $-H_{\theta_0}^{-1} \geq -H_\infty^{-1}$. The simplest way to think about this is that $Cov(\hat{\theta})$ includes two sources of uncertainty: 1) our posterior uncertainty about θ given the complete data (y, z), and 2) our posterior uncertainty about z. The auxiliary function $Q(.)$ is based on a conditional expectation over z, and therefore effectively captures this first component of our uncertainty but ignores the important second component, which corresponds exactly to the "missing" information (Meng and Rubin, 1991).

References

Ahrens, M., Paninski, L., and Sahani, M. (2008). Inferring input nonlinearities in neural encoding models. *Network: Computation in Neural Systems*, 19:35–67.

Aldworth, Z., Miller, J., Gedeon, T., Cummins, G., and Dimitrov, A. (2005). Dejittered spike-conditioned stimulus waveforms yield improved estimates of neuronal feature selectivity and spike-timing precision of sensory interneurons. *Journal of Neuroscience*, 25:5323–5332.

Amarasingham, A., Harrison, M., and Geman, S. (2005). Statistical techniques for analyzing non-repeating spike trains. *SFN Abstracts*.

Bar-Hillel, A., Spiro, A., and Stark, E. (2006). Spike sorting: Bayesian clustering of non-stationary data. *Journal of Neuroscience Methods*, 157:303–316.

Beal, M. and Ghahramani, Z. (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. In *Bayesian Statistics 7*. Oxford.

Behseta, S., Kass, R., and Wallstrom, G. (2005). Hierarchical models for assessing variability among functions. *Biometrika*, 92:419–434.

Brockwell, A. E., Kass, R. E., and Schwartz, A. (2007). Statistical signal processing and the motor cortex. *Proceedings of the IEEE*, 95:1–18.

Calabrese, A. and Paninski, L. (2009). Kalman-based methods for tracking nonstationary cluster means in spike-sorting applications. *In preparation*.

Chang, T., Chung, P., Chiu, T., and Poon, P. (2005). A new method for adjusting neural response jitter in the STRF obtained by spike-trigger averaging. *Biosystems*, 79:213–222.

Collins, M., Schapire, R. E., and Singer, Y. (2000). Logistic regression, Adaboost and Bregman distances. In *Computational Learing Theory*, pages 158–169.

Darroch, J. and Ratcliff, D. (1972). Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, 43:1470–1480.

de la Rocha, J., Doiron, B., Shea-Brown, E., Josic, K., and Reyes, A. (2007). Correlation between neural spike trains increases with firing rate. *Nature*, 448:802–806.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal Royal Stat. Soc., Series B*, 39:1–38.

Dimitrov, A. and Gedeon, T. (2006). Effects of stimulus transformations on characteristics of sensory neuron function. *Journal of Computational Neuroscience*, 20:265–283.

Dimitrov, A., Sheiko, M., Baker, J., and Yen, S.-C. (2009). Spatial and temporal jitter distort estimated functional properties of visual sensory neurons. *Journal of Computational Neuroscience*.

Dorn, J. D. and Ringach, D. L. (2003). Estimating membrane voltage correlations from extracellular spike trains. *Journal of Neurophysiology*, 89(4):2271–2278.

Faisal, A. A. and Laughlin, S. B. (2007). Stochastic simulations on the reliability of action potential propagation in thin axons. *PLoS Comput Biol*, 3(5):e79.

Gollisch, T. (2006). Estimating receptive fields in the presence of spike-time jitter. *Network: Computation in Neural Systems*, 17:103–129.

Krishnapuram, B., Figueiredo, M., Carin, L., and Hartemink, A. (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27:957–968.

Krumin, M. and Shoham, S. (2008). Characterization of input-output relations in single neurons using spatiotemporal photo-stimulation. In *14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics*, pages 378–379.

Kulkarni, J. and Paninski, L. (2007). Common-input models for multiple neural spike-train data. *Network: Computation in Neural Systems*, 18:375–407.

Lewicki, M. (1998). A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9:R53–R78.

Litke, A., Bezayiff, N., Chichilnisky, E., Cunningham, W., Dabrowski, W., Grillo, A., Grivich, M., Grybos, P., Hottowy, P., Kachiguine, S., Kalmar, R., Mathieson, K., Petrusca, D., Rahman, M., and Sher, A. (2004). What does the eye tell the brain? development of a system for the large scale recording of retinal output activity. *IEEE Trans Nucl Sci*, pages 1434–1440.

Macke, J., Berens, P., Ecker, A., Tolias, A., and Bethge, M. (2009). Generating spike trains with specified correlation coefficients. *Neural Computation*, 21:In press.

McLachlan, G. and Krishnan, T. (1996). *The EM Algorithm and Extensions*. Wiley-Interscience.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley-Interscience.

Meng, X.-L. and Rubin, D. B. (1991). Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm. *Journal of the American Statistical Association*, 86(416):899–909.

Neal, R. and Hinton, G. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M., editor, *Learning in Graphical Models*, pages 355–368. MIT Press.

Niebur, E. (2007). Generation of synthetic spike trains with defined pairwise correlations. *Neural Computation*, 19(7):1720–1738.

Nykamp, D. (2005). Revealing pairwise coupling in linear-nonlinear networks. *SIAM Journal on Applied Mathematics*, 65:2005–2032.

Nykamp, D. (2007). A mathematical framework for inferring connectivity in probabilistic neuronal networks. *Mathematical Biosciences*, 205:204–251.

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). Log-concavity results on Gaussian process methods for supervised and unsupervised learning. *Advances in Neural Information Processing Systems*, 17.

Peel, D. and McLachlan, G. J. (2000). Robust mixture modelling using the t distribution. *Statistics and Computing*, 10(4):339–348.

Petrusca, D., Grivich, M. I., Sher, A., Field, G. D., Gauthier, J. L., Greschner, M., Shlens, J., Chichilnisky, E. J., and Litke, A. M. (2007). Identification and characterization of a Y-like primate retinal ganglion cell type. *J. Neurosci.*, 27(41):11019–11027.

Pouzat, C., Delescluse, M., Viot, P., and Diebolt, J. (2004). Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: A Markov chain Monte Carlo approach. *Journal of Neurophysiology*, 91:2910–2928.

Quian Quiroga, R. (2007). Spike sorting. *Scholarpedia*, http://www.scholarpedia.org/article/Spike_sorting.

Ringach, D. (2002). Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *Journal of Neurophysiology*, 88:455–463.

Ringach, D., Shapley, R., and Hawken, M. (2002). Orientation selectivity in macaque v1: diversity and laminar dependence. *Journal of Neuroscience*, 22:5639–5651.

Saad, D. and Opper, M., editors (2001). *Advanced Mean Field Methods: Theory and Practice*. MIT Press.

Sahani, M. (1999). *Latent variable models for neural data analysis*. PhD thesis, California Institute of Technology.

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

Salakhutdinov, R., Roweis, S. T., and Ghahramani, Z. (2003). Optimization with EM and expectation-conjugate-gradient. *International Conference on Machine Learning*, 20:672–679.

Santhanam, G., Yu, B. M., Gilja, V., Ryu, S. I., Afshar, A., Sahani, M., and Shenoy, K. V. (2009). Factor-analysis methods for higher-performance neural prostheses. *Journal of Neurophysiology*. In press.

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

Segev, R., Goodhouse, J., Puchalla, J., and Berry, M. (2004). Recording spikes from a large fraction of the ganglion cells in a retinal patch. *Nature Neuroscience*, 7:1154–1161.

Sha, F., Saul, L., and Lee, D. (2003). Multiplicative updates for nonnegative quadratic programming in support vector machines. In *Advances in Neural Information Processing Systems 15*, pages 1041–1048. MIT Press.

Shadlen, M. and Newsome, W. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of Neuroscience*, 18:3870–3896.

Shoham, S., Fellows, M., and Normann, R. (2003). Robust, automatic spike sorting using mixtures of multivariate t-distributions. *Journal of Neuroscience Methods*, 127:111–122.

Shoham, S., Paninski, L., Fellows, M., Hatsopoulos, N., Donoghue, J., and Normann, R. (2005). Optimal decoding for a primary motor cortical brain-computer interface. *IEEE Transactions on Biomedical Engineering*, 52:1312–1322.

Softky, W. and Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.*, 13(1):334–350.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal Of The Royal Statistical Society Series B*, 61(3):611–622.

Tolhurst, D. Movshon, J. and Dean, A. (1983). The statistical reliability of single neurons in cat and monkey visual cortex. *Vision Research*, 23:775–785.

Ventura, V. (2008). Automatic spike sorting using tuning information. *Submitted*.

Vidne, M., Kulkarni, J., Ahmadian, Y., Pillow, J., Shlens, J., Chichilnisky, E., Simoncelli, E., and Paninski, L. (2009). Inferring functional connectivity in an ensemble of retinal ganglion cells sharing a common input. *COSYNE*.

Wood, F. and Black, M. (2008). A nonparametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173:1–12.

Yu, B., Cunningham, J., Santhanam, G., Ryu, S., Shenoy, K., and Sahani, M. (2009a). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *NIPS*.

Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V., and Sahani, M. (2009b). Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102:614–635.