3 Neural Decoding

3.1   Encoding and Decoding

In chapters 1 and 2, we considered the problem of predicting neural responses to known stimuli. The nervous system faces the reverse problem, determining what is going on in the real world from neuronal spiking patterns. It is interesting to attempt such computations ourselves, using the responses of one or more neurons to identify a particular stimulus or to extract the value of a stimulus parameter. We will assess the accuracy with which this can be done primarily by using optimal decoding techniques, regardless of whether the computations involved seem biologically plausible. Some biophysically realistic implementations are discussed in chapter 7. Optimal decoding allows us to determine limits on the accuracy and reliability of neuronal encoding. In addition, it is useful for estimating the information content of neuronal spike trains, an issue addressed in chapter 4.

As we discuss in chapter 1, neural responses, even to a single repeated stimulus, are typically described by stochastic models due to their inherent variability. In addition, stimuli themselves are often described stochastically. For example, the stimuli used in an experiment might be drawn randomly from a specified probability distribution. Natural stimuli can also be modeled stochastically as a way of capturing the statistical properties of complex environments.

Given this twofold stochastic model, encoding and decoding are related through a basic identity of probability theory called Bayes theorem. Let $\mathbf{r}$ represent the response of a neuron or a population of neurons to a stimulus characterized by a parameter s. Throughout this chapter, $\mathbf{r} = (r_1, r_2, \ldots, r_N)$ for N neurons is a list of spike-count firing rates, although, for the present discussion, it could be any other set of parameters describing the neuronal response. Several different probabilities and conditional probabilities enter into our discussion. A conditional probability is just an ordinary probability of an event occurring, except that its occurrence is subject to an additional condition. The conditional probability of event A occurring subject to the condition B is denoted by $P[A|B]$. The probabilities we need are:

conditional probability

- $P[s]$, the probability of stimulus s being presented, often called the prior probability

- $P[\mathbf{r}]$, the probability of response $\mathbf{r}$ being recorded

joint probability
- $P[\mathbf{r}, s]$, the probability of stimulus s being presented and response $\mathbf{r}$ being recorded. This is called the joint probability

- $P[\mathbf{r}|s]$, the conditional probability of evoking response $\mathbf{r}$, given that stimulus s was presented

- $P[s|\mathbf{r}]$, the conditional probability that stimulus s was presented, given that response $\mathbf{r}$ was recorded.

Note that $P[\mathbf{r}|s]$ is the probability of observing the rates $\mathbf{r}$, given that the stimulus took the value s, while $P[\mathbf{r}]$ is the probability of the rates taking the values $\mathbf{r}$ independent of what stimulus was used. $P[\mathbf{r}]$ can be computed from $P[\mathbf{r}|s]$ by summing over all stimulus values weighted by their probabilities,

$$P[\mathbf{r}] = \sum_s P[\mathbf{r}|s]P[s] \;\; \text{and similarly} \;\; P[s] = \sum_{\mathbf{r}} P[s|\mathbf{r}]P[\mathbf{r}]\,. \qquad (3.1)$$

An additional relationship between the probabilities listed above can be derived by noticing that the joint probability $P[\mathbf{r}, s]$ can be expressed as either the conditional probability $P[\mathbf{r}|s]$ times the probability of the stimulus, or as $P[s|\mathbf{r}]$ times the probability of the response,

$$P[\mathbf{r}, s] = P[\mathbf{r}|s]P[s] = P[s|\mathbf{r}]P[\mathbf{r}]\,. \qquad (3.2)$$

Bayes theorem   This is the basis of Bayes theorem relating $P[s|\mathbf{r}]$ to $P[\mathbf{r}|s]$:

$$P[s|\mathbf{r}] = \frac{P[\mathbf{r}|s]P[s]}{P[\mathbf{r}]}\,, \qquad (3.3)$$

assuming that $P[\mathbf{r}] \neq 0$. Encoding is characterized by the set of probabilities $P[\mathbf{r}|s]$ for all stimuli and responses. Decoding a response, on the other hand, amounts to determining the probabilities $P[s|\mathbf{r}]$. According to Bayes theorem, $P[s|\mathbf{r}]$ can be obtained from $P[\mathbf{r}|s]$, but the stimulus probability $P[s]$ is also needed. As a result, decoding requires knowledge of the statistical properties of experimentally or naturally occurring stimuli.

In the above discussion, we have assumed that both the stimulus and the response are characterized by discrete values so that ordinary probabilities, not probability densities, are used to describe their distributions. For example, firing rates obtained by counting spikes over the duration of a trial take discrete values and can be described by a probability. However, we sometimes treat the response firing rates or the stimulus values as continuous variables. In this case, the probabilities listed must be replaced by the corresponding probability densities, $p[\mathbf{r}]$, $p[\mathbf{r}|s]$, etc. Nevertheless, the relationships discussed above are equally valid.
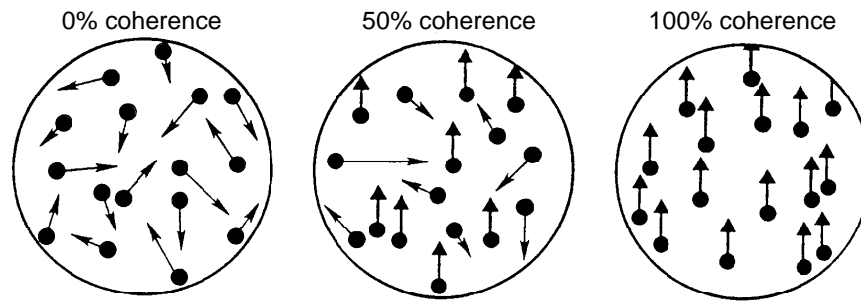
Figure 3.1 The moving random-dot stimulus for different levels of coherence. The visual image consists of randomly placed dots that jump every 45 ms according to the scheme described in the text. At 0% coherence the dots move randomly. At 50% coherence, half the dots move randomly and half move together (upward in this example). At 100% coherence all the dots move together. (Adapted from Britten et al., 1992.)

In the following sections, we present examples of decoding that involve both single neurons and neuronal populations. We first study a restricted case of single-cell decoding, discrimination between two different stimulus values. We then consider extracting the value of a parameter that characterizes a static stimulus from the responses of a population of neurons. As a final example, we return to single neurons and discuss spike-train decoding, in which an estimate of a time-varying stimulus is constructed from the spike train it evokes.

3.2   Discrimination

To introduce the notion of discriminability and the receiver operating characteristic that lie at the heart of discrimination analysis, we will discuss a fascinating study performed by Britten et al. (1992). In their experiments, a monkey was trained to discriminate between two directions of motion of a visual stimulus. The stimulus was a pattern of dots on a video monitor that jump from random initial locations to new locations every 45 ms. To introduce a sense of directed movement at a particular velocity, a percentage of the dots move together by a fixed amount in a fixed direction (figure 3.1). The coherently moving dots are selected randomly at each time step, and the remaining dots move to random new locations. The percentage of dots that move together in the fixed direction is called the coherence level. At 0% coherence, the image appears chaotic with no sense of any particular direction of motion. As the coherence increases, a sense of movement in a particular direction appears in the image until, at 100% coherence, the entire array of dots moves together on the monitor. By varying the degree of coherence, the task of detecting the movement direction can be made more or less difficult.

The experiments combined neural recording with behavioral measurements. In the behavioral part, the monkey had to report the direction
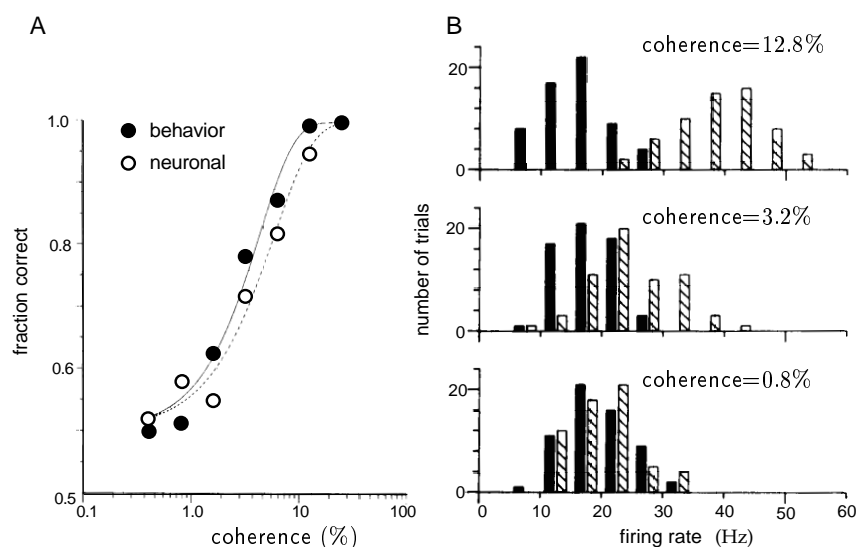
Figure 3.2 Behavioral and electrophysiological results from a random-dot motion-discrimination task. (A) The filled circles show the fraction of correct discriminations made by a monkey as a function of the degree of coherence of the motion. The open circles show the discrimination accuracy that an ideal observer could achieve on the analogous two-alternative forced-choice discrimination task, given the neural responses. (B) Firing-rate histograms for three different levels of coherence. Hatched rectangles show the results for motion in the plus direction, and solid rectangles, for motion in the minus direction. The histograms have been thinned for clarity so that not all the bins are shown. (Adapted from Britten et al., 1992.)

of motion in the random dot images. During the same task, recordings were made from neurons in area MT. Only two possible directions of coherent movement of the dots were used while a particular neuron was being recorded; either the direction that produced the maximum response in that neuron, or the opposite direction. The monkey's task was to discriminate between these two directions. The filled circles and solid curve in figure 3.2A show the proportion of correct responses in a typical experiment. Below 1% coherence, the responses were near chance (fraction correct = 0.5), but the monkey approached perfect performance (fraction correct = 1) above 10% coherence.

Figure 3.2B shows histograms of average firing rates in response to different levels of movement coherence. The firing rates plotted are the number of spikes recorded during the 2 s period that the stimulus was presented, divided by 2 s. The neuron shown tended to fire more spikes when the motion was in its preferred direction, which we will call the plus (or $+$) direction (hatched histogram), than in the other, minus (or $-$) direction (solid histogram). At high coherence levels, the firing-rate distributions corresponding to the two directions are fairly well separated, while at low coherence levels, they merge. Although spike count rates take only discrete values, it is more convenient to treat r as a continuous variable for our discussion. Treated as probability densities, these two distributions are

approximately Gaussian with the same variance, σ_r^2, but different means, $\langle r \rangle_+$ for the plus direction and $\langle r \rangle_-$ for the minus direction. A convenient measure of the separation between the distributions is the discriminability

discriminability d'

$$d' = \frac{\langle r \rangle_+ - \langle r \rangle_-}{\sigma_r} \, , \qquad (3.4)$$

which is the distance between the means in units of their common standard deviation. The larger d', the more separated the distributions.

In the example we are considering, decoding involves using the neural response to determine in which of the two possible directions the stimulus moved. A simple decoding procedure is to determine the firing rate r during a trial and compare it to a threshold number z. If $r \geq z$, we report "plus"; otherwise we report "minus". Figure 3.2B suggests that if we choose z to lie somewhere between the two distributions, this procedure will give the correct answer at high coherence, but will have difficulty distinguishing the two directions at low coherence. This difficulty is clearly related to the degree to which the two distributions in figure 3.2B overlap, and thus to the discriminability.

The probability that the procedure outlined in the previous paragraph will generate the correct answer (called a hit) when the stimulus is moving in the plus direction is the conditional probability that $r \geq z$ given a plus stimulus, $P[r \geq z|+]$. The probability that it will give the answer "plus" when the stimulus is actually moving in the minus direction (called a false alarm) is similarly $P[r \geq z|-]$. These two probabilities completely determine the performance of the decoding procedure because the probabilities for the other two cases (reporting "minus" when the correct answer is "plus", and reporting "minus" when the correct answer is "minus") are $1 - P[r \geq z|+]$ and $1 - P[r \geq z|-]$, respectively. In signal detection theory, the quantity used to perform the discrimination, r in our case, is called the test, and the two probabilities corresponding to reporting a "plus" answer have specific names:

test size and power or false alarm and hit rate

$$
\begin{aligned}
\alpha(z) &= P[r \geq z|-] \quad \text{is the size or false alarm rate of the test} \\
\beta(z) &= P[r \geq z|+] \quad \text{is the power or hit rate of the test.}
\end{aligned} \qquad (3.5)
$$

The following table shows how the probabilities of the test giving correct and incorrect answers in the different cases depend on α and β.

stimulus	probability	
	correct	incorrect
$+$	β	$1 - \beta$
$-$	$1 - \alpha$	α

The performance of the decoding procedure we have been discussing depends critically on the value of the threshold z to which the rate r is compared. Obviously, we would like to use a threshold for which the size

is near 0 and the power near 1. In general, it is impossible to choose the threshold so that both the size and the power of the test are optimized; a compromise must be made. A logical optimization criterion is to maximize the probability of getting a correct answer, which is equal to $(\beta(z) + 1 - \alpha(z))/2$ if the plus and minus stimuli occur with equal probability. Although this is a possible approach for the experiment we are studying, the analysis we present introduces a powerful technique that makes better use of the full range of recorded data and can be generalized to tasks where the optimal strategy is unknown. This approach makes use of ROC curves, which indicate how the size and power of a test trade off as the threshold is varied.

ROC Curves

receiver operating characteristic, ROC

The receiver operating characteristic (ROC) curve provides a way of evaluating how test performance depends on the choice of the threshold z. Each point on an ROC curve corresponds to a different value of z. The x coordinate of the point is α, the size of the test for this value of z, and the y coordinate is β, its power. As the threshold is varied continuously, these points trace out the ROC plot. If $z = 0$, the firing rate will always be greater than or equal to z, so the decoding procedure will always give the answer "plus". Thus, for $z = 0$, $\alpha = \beta = 1$, producing a point at the upper-right corner of the ROC plot. At the other extreme, if z is very large, r will always be less than z, the test will always report "minus", and $\alpha = \beta = 0$. This produces a point at the bottom-left corner of the plot. Between these extremes, a curve is traced out as a function of z.

Figure 3.3 shows ROC curves computed by Britten et al. for several different values of the stimulus coherence. At high coherence levels, when the task is easy, the ROC curve rises rapidly from $\alpha(z) = 0$, $\beta(z) = 0$ as the threshold is lowered from a high value, and the probability $\beta(z)$ of a correct "plus" answer quickly approaches 1 without a concomitant increase in $\alpha(z)$. As the threshold is lowered further, the probability of giving the answer "plus" when the correct answer is "minus" also rises, and $\alpha(z)$ increases. When the task is difficult, the curve rises more slowly as z is lowered; and if the task is impossible, in that the test merely gives random answers, the curve will lie along the diagonal $\alpha = \beta$, because the probabilities of answers being correct and incorrect are equal. This is exactly the trend of the ROC curves at different coherence levels shown in figure 3.3.

Examination of figure 3.3 suggests a relationship between the area under the ROC curve and the level of performance on the task. When the ROC curve in figure 3.3 lies along the diagonal, the area underneath it is 1/2, which is the probability of a correct answer in this case (given any threshold). When the task is easy and the ROC curve hugs the left axis and upper limit in figure 3.3, the area under it approaches 1, which is again the probability of a correct answer (given an appropriate threshold). However, the precise relationship between task performance and the area under the
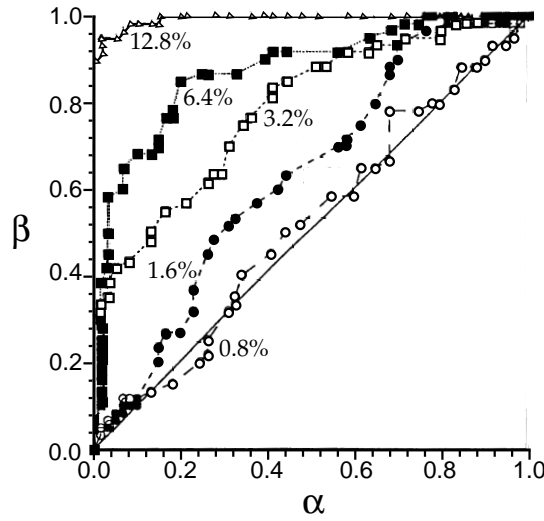
Figure 3.3 ROC curves for a variety of motion coherence levels. Each curve is the locus of points $(\alpha(z), \beta(z))$ for all z values. The values of α and β were computed from histograms such as those in figure 3.2B. The diagonal line is the ROC curve for random guessing. (Adapted from Britten et al., 1992.)

ROC curve is complicated by the fact that different threshold values can be used. This ambiguity can be removed by considering a slightly different task, called two-alternative forced choice. Here, the stimulus is presented twice, once with motion in the plus direction and once in the minus direction. The task is to decide which presentation corresponded to the plus direction, given the firing rates on both trials, r_1 and r_2. A natural extension of the test procedure we have been discussing is to answer trial 1 if $r_1 \geq r_2$ and otherwise answer trial 2. This removes the threshold variable from consideration.

two-alternative forced choice

In the two-alternative force-choice task, the value of r on one trial serves as the threshold for the other trial. For example, if the order of stimulus presentation is plus, then minus, the comparison procedure we have outlined will report the correct answer if $r_1 \geq z$ where $z = r_2$, and this has probability $P[r_1 \geq z|+] = \beta(z)$ with $z = r_2$. To determine the probability of getting the correct answer in a two-alternative forced-choice task, we need to integrate this probability over all possible values of r_2 weighted by their probability of occurrence. For small Δz, the probability that r_2 takes a value in the range between z and $z + \Delta z$ when the second trial has a minus stimulus is $p[z|-]\Delta z$, where $p[z|-]$ is the conditional firing-rate probability density for a firing rate $r = z$. Integrating over all values of z gives the probability of getting the correct answer,

$$P[\text{correct}] = \int_0^\infty dz\, p[z|-]\beta(z) \,. \tag{3.6}$$

Because the two-alternative forced-choice test is symmetric, this is also the

probability of being correct if the order of the stimuli is reversed.

The probability that $r \geq z$ for a minus stimulus, which is just $\alpha(z)$, can be written as an integral of the conditional firing-rate probability density $p[r|-]$,

$$\alpha(z) = \int_z^\infty dr\, p[r|-] \,. \tag{3.7}$$

Taking the derivative of this equation with respect to z, we find that

$$\frac{d\alpha}{dz} = -p[z|-] \,. \tag{3.8}$$

This allows us to make the replacement $dz\, p[z|-] \to -d\alpha$ in the integral of equation 3.6 and to change the integration variable from z to α. Noting that $\alpha = 1$ when $z = 0$ and $\alpha = 0$ when $z = \infty$, we find

$$P[\text{correct}] = \int_0^1 d\alpha\, \beta \,. \tag{3.9}$$

The ROC curve is just β plotted as a function of α, so this integral is the area under the ROC curve. Thus, the area under the ROC curve is the probability of responding correctly in the two-alternative forced-choice test.

Suppose that $p[r|+]$ and $p[r|-]$ are both Gaussian functions with means $\langle r \rangle_+$ and $\langle r \rangle_-$, and a common variance σ_r^2. The reader is invited to show that, in this case,

$$P[\text{correct}] = \frac{1}{2}\text{erfc}\left(\frac{\langle r \rangle_- - \langle r \rangle_+}{2\sigma_r}\right) = \frac{1}{2}\text{erfc}\left(-\frac{d'}{2}\right), \tag{3.10}$$

complementary error function   where d' is the discriminability defined in equation 3.4 and $\text{erfc}(x)$ is the complementary error function (which is an integral of a Gaussian distribution) defined as

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty dy\, \exp(-y^2) \,. \tag{3.11}$$

In the case where the distributions are equal-variance Gaussians, the relationship between the discriminability and the area under the ROC curve is invertible because the complementary error function is monotonic. It is common to quote d' values even for non-Gaussian distributions by inverting the relationship between $P[\text{correct}]$ and d' in equation 3.10.

ROC Analysis of Motion Discrimination

To interpret their experiment as a two-alternative forced-choice task, Britten et al. imagined that, in addition to being given the firing rate of the recorded neuron during stimulus presentation, the observer is given the firing rate of a hypothetical "anti-neuron" having response characteristics

exactly opposite from the recorded neuron. In reality, the responses of this anti-neuron to a plus stimulus were just those of the recorded neuron to a minus stimulus, and vice versa. The idea of using the responses of a single neuron to opposite stimuli as if they were the simultaneous responses of two different neurons also reappears in our discussion of spike-train decoding. An observer predicting motion directions on the basis of just these two neurons at a level equal to the area under the ROC curve is termed an ideal observer.

Figure 3.2A shows a typical result for the performance of an ideal observer using one recorded neuron and its anti-neuron partner. The open circles in figure 3.2A were obtained by calculating the areas under the ROC curves for this neuron. Amazingly, the ability of the ideal observer to perform the discrimination task using a single neuron/anti-neuron pair is equal to the ability of the monkey to do the task. Although the choices of the ideal observer and the monkey do not necessarily match on a trial-to-trial basis, their performances are comparable when averaged over trials. This seems remarkable because the monkey presumably has access to a large population of neurons, while the ideal observer uses only two. One speculation is that correlations in the response variability between neurons limit the performance of the monkey.

The Likelihood Ratio Test

The discrimination test we have considered compares the firing rate to a threshold value. Could an observer do better than this already remarkable performance by comparing some other function of the firing rate to a threshold? What is the best test function to use for this purpose? The Neyman-Pearson lemma (proved in appendix A) shows that it is impossible to do better than to choose as the test function the ratio of probability densities (or, where appropriate, probabilities),

Neyman-Pearson lemma

$$l(r) = \frac{p[r|+]}{p[r|-]},$$ (3.12)

which is known as the likelihood ratio. The test function r used above is not equal to the likelihood ratio. However, if the likelihood ratio is a monotonically increasing function of r, as it is for the data of Britten et al., the firing-rate threshold test is equivalent to using the likelihood ratio and is indeed optimal. Similarly, any monotonic function of the likelihood ratio will provide as good a test as the likelihood itself, and the logarithm is frequently used.

likelihood ratio

There is a direct relationship between the likelihood ratio and the ROC curve. As in equations 3.7 and 3.8, we can write

$$\beta(z) = \int_z^\infty dr\, p[r|+] \quad \text{so} \quad \frac{d\beta}{dz} = -p[z|+].$$ (3.13)

Combining this result with 3.8, we find that

$$\frac{d\beta}{d\alpha} = \frac{d\beta}{dz}\frac{dz}{d\alpha} = \frac{p[z|+]}{p[z|-]} = l(z)\,, \tag{3.14}$$

so the slope of the ROC curve is equal to the likelihood ratio.

Another way of seeing that comparing the likelihood ratio to a threshold value is an optimal decoding procedure for discrimination uses a Bayesian approach based on associating a cost or penalty with getting the wrong answer. Suppose that the penalty associated with answering "minus" when the correct answer is "plus" is quantified by the loss parameter L_-. Similarly, quantify the loss for answering "plus" when the correct answer is "minus" as L_+. For convenience, we assume that there is neither loss nor gain for answering correctly. The probabilities that the correct answer is "plus" or "minus", given the firing rate r, are $P[+|r]$ and $P[-|r]$ respectively. These probabilities are related to the conditional firing-rate probability densities by Bayes theorem,

$$P[+|r] = \frac{p[r|+]P[+]}{p[r]} \quad \text{and} \quad P[-|r] = \frac{p[r|-]P[-]}{p[r]}\,. \tag{3.15}$$

The average loss expected for a "plus" answer when the firing rate is r is the loss associated with being wrong times the probability of being wrong, $\text{Loss}_+ = L_+ P[-|r]$. Similarly, the expected loss when answering "minus" is $\text{Loss}_- = L_- P[+|r]$. A reasonable strategy is to cut the losses, answering "plus" if $\text{Loss}_+ \leq \text{Loss}_-$ and "minus" otherwise. Using equation 3.15, we find that this strategy gives the response "plus" if

$$l(r) = \frac{p[r|+]}{p[r|-]} \geq \frac{L_+}{L_-}\frac{P[-]}{P[+]}\,. \tag{3.16}$$

This shows that the strategy of comparing the likelihood ratio to a threshold is a way of minimizing the expected loss. The right side of this inequality gives an explicit formula for the value of the threshold that should be used, and reflects two factors. One is the relative losses for the two sorts of possible errors. The other is the prior probabilities that the stimulus is plus or minus. Interestingly, it is possible to change the thresholds that human subjects use in discrimination tasks by manipulating these two factors.

If the conditional probability densities $p[r|+]$ and $p[r|-]$ are Gaussians with means r_+ and r_- and identical variances σ_r^2, and $P[+] = P[-] = 1/2$, the probability $P[+|r]$ is a sigmoidal function of r,

$$P[+|r] = \frac{1}{1 + \exp(-d'(r - r_{\text{ave}})/\sigma_r)}\,, \tag{3.17}$$

where $r_{\text{ave}} = (r_+ + r_-)/2$. This provides an alternate interpretation of the parameter d' that is often used in the psychophysics literature; it determines the slope of a sigmoidal function fitted to $P[+|r]$.

We have thus far considered discriminating between two quite distinct stimulus values, plus and minus. Often we are interested in discriminating between two stimulus values $s + \Delta s$ and s that are very close to one another. In this case, the likelihood ratio is

$$\frac{p[r|s + \Delta s]}{p[r|s]} \approx \frac{p[r|s] + \Delta s \, \partial p[r|s]/\partial s}{p[r|s]}$$

$$= 1 + \Delta s \frac{\partial \ln p[r|s]}{\partial s} . \qquad (3.18)$$

For small Δs, a test that compares

$$Z(r) = \frac{\partial \ln p[r|s]}{\partial s} \qquad (3.19)$$

to a threshold $(z - 1)/\Delta s$ is equivalent to the likelihood ratio test. The function $Z(r)$ is sometimes called the score. *score $Z(r)$*

3.3 Population Decoding

The use of large numbers of neurons to represent information is a basic operating principle of many nervous systems. Population coding has a number of advantages, including reduction of uncertainty due to neuronal variability and the ability to represent a number of different stimulus attributes simultaneously. Individual neurons in such a population typically have different but overlapping selectivities, so that many neurons, but not necessarily all, respond to a given stimulus. In the previous section, we discussed discrimination between stimuli on the basis of the response of a single neuron. The responses of a population of neurons can also be used for discrimination, with the only essential difference being that terms such as $p[r|s]$ are replaced by $p[\mathbf{r}|s]$, the conditional probability density of the population response $\mathbf{r}$. ROC analysis, likelihood ratio tests, and the Neyman-Pearson lemma continue to apply in exactly the same way. Discrimination is a special case of decoding in which only a few different stimulus values are considered. A more general problem is the extraction of a continuous stimulus parameter from one or more neuronal responses. In this section, we study how the value of a continuous parameter associated with a static stimulus can be decoded from the spike-count firing rates of a population of neurons.

Encoding and Decoding Direction

The cercal system of the cricket, which senses the direction of incoming air currents as a warning of approaching predators, is an interesting example of population coding involving a relatively small number of neurons. Crickets and related insects have two appendages called cerci extending
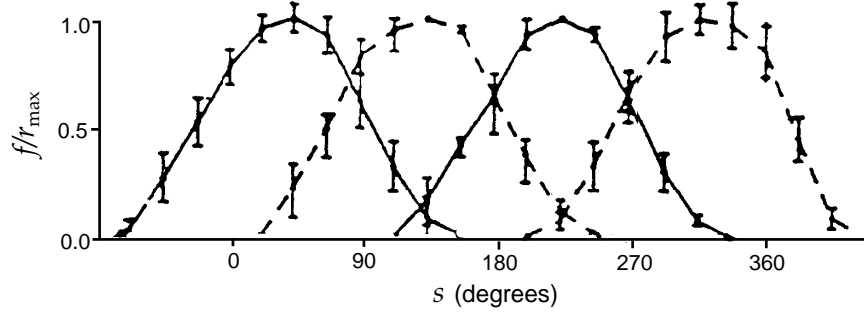
Figure 3.4 Tuning curves for the four low-velocity interneurons of the cricket cercal system plotted as a function of the wind direction s. Each neuron responds with a firing rate that is closely approximated by a half-wave rectified cosine function. The preferred directions of the neurons are located $90°$ from each other, and r_{max} values are typically around 40 Hz. Error bars show standard deviations. (Adapted from Theunissen and Miller, 1991.)

from their hind ends. These are covered with hairs that are deflected by air currents. Each hair is attached to a neuron that fires when the hair is deflected. Thousands of these primary sensory neurons send axons to a set of interneurons that relay the sensory information to the rest of the cricket's nervous system. No single interneuron of the cercal system responds to all wind directions, and multiple interneurons respond to any given wind direction. This implies that the interneurons encode the wind direction collectively as a population.

Theunissen and Miller (1991) measured both the mean and the variance of responses of cercal interneurons while blowing air currents at the cerci. At low wind velocities, information about wind direction is encoded by just four interneurons. Figure 3.4 shows average firing-rate tuning curves for the four relevant interneurons as a function of wind direction. These neurons are sensitive primarily to the angle of the wind around the vertical axis and not to its elevation above the horizontal plane. Wind speed was held constant in these experiments, so we do not discuss how it is encoded. The interneuron tuning curves are well approximated by half-wave rectified cosine functions. Neuron a (where $a = 1, 2, 3, 4$) responds with a maximum average firing rate when the angle of the wind direction is s_a, the preferred-direction angle for that neuron. The tuning curve for interneuron a in response to wind direction s, $\langle r_a \rangle = f_a(s)$, normalized to *cosine tuning*   its maximum, can be written as

$$\left(\frac{f(s)}{r_{max}} \right)_a = [(\cos(s - s_a)]_+ \,, \tag{3.20}$$

where the half-wave rectification eliminates negative firing rates. Here r_{max}, which may be different for each neuron, is a constant equal to the maximum average firing rate. The fit can be improved somewhat by introducing a small offset rate, but the simple cosine is adequate for our purposes.

To determine the wind direction from the firing rates of the cercal interneurons, it is useful to change the notation somewhat. In place of the angle s, we can represent wind direction by a spatial vector $\vec{v}$ pointing parallel to the wind velocity and having unit length $|\vec{v}| = 1$ (we use over-arrows to denote spatial vectors). Similarly, we can represent the preferred wind direction for each interneuron by a vector $\vec{c}_a$ of unit length pointing in the direction specified by the angle s_a. In this case, we can use the vector dot product to write $\cos(s - s_a) = \vec{v} \cdot \vec{c}_a$. In terms of these vectors, the average firing rate is proportional to a half-wave rectified projection of the wind direction vector onto the preferred-direction axis of the neuron,

dot product

$$\left(\frac{f(s)}{r_{\max}} \right)_a = \left[\vec{v} \cdot \vec{c}_a \right]_+ . \tag{3.21}$$

Decoding the cercal system is particularly easy because of the close relationship between the representation of wind direction it provides and a two-dimensional Cartesian coordinate system. In a Cartesian system, vectors are parameterized by their projections onto x and y axes, v_x and v_y. These projections can be written as dot products of the vector being represented, $\vec{v}$, with vectors of unit length $\vec{x}$ and $\vec{y}$ lying along the x and y axes, $v_x = \vec{v} \cdot \vec{x}$ and $v_y = \vec{v} \cdot \vec{y}$. Except for the half-wave rectification, these equations are identical to equation 3.21. Furthermore, the preferred directions of the four interneurons, like the x and y axes of a Cartesian coordinate system, lie along two perpendicular directions (figure 3.5A). Four neurons are required, rather than two, because firing rates cannot represent negative projections. The cricket discovered the Cartesian coordinate system long before Descartes did, but failed to invent negative numbers! Perhaps credit should also be given to the leech, for Lewis and Kristan (1998) have shown that the direction of touch sensation in its body segments is encoded by four neurons in a virtually identical arrangement.

A vector $\vec{v}$ can be reconstructed from its Cartesian components through the component-weighted vector sum $\vec{v} = v_x \vec{x} + v_y \vec{y}$. Because the firing rates of the cercal interneurons we have been discussing are proportional to the Cartesian components of the wind direction vector, a similar sum should allow us to reconstruct the wind direction from a knowledge of the interneuron firing rates, except that four, not two, terms must be included. If r_a is the spike-count firing rate of neuron a, an estimate of the wind direction on any given trial can be obtained from the direction of the vector

$$\vec{v}_{\text{pop}} = \sum_{a=1}^{4} \left(\frac{r}{r_{\max}} \right)_a \vec{c}_a . \tag{3.22}$$

This vector is known as the population vector, and the associated decoding method is called the vector method. This decoding scheme works quite well. Figure 3.5B shows the root-mean-square difference between the direction determined by equation 3.22 and the actual wind direction that evoked the firing rates. The difference between the decoded and actual wind directions is around $6°$ except for dips at the angles corresponding to the preferred directions of the neurons. These dips are not due to the

population vector
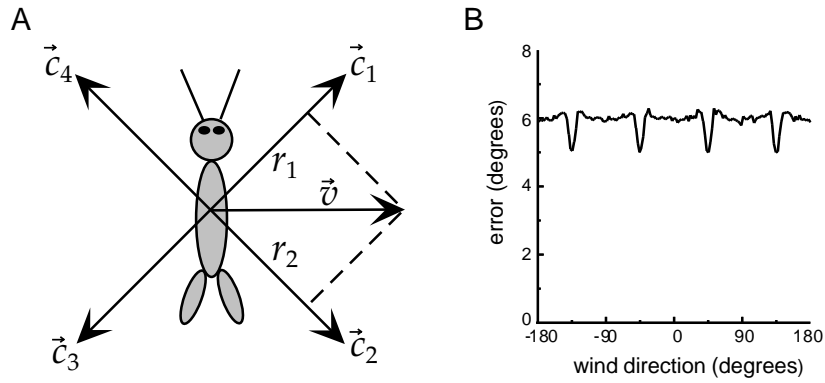
vector method

A

B

Figure 3.5 (A) Preferred directions of four cercal interneurons in relation to the cricket's body. The firing rate of each neuron for a fixed wind speed is proportional to the projection of the wind velocity vector $\vec{v}$ onto the preferred-direction axis of the neuron. The projection directions $\vec{c}_1$, $\vec{c}_2$, $\vec{c}_3$, and $\vec{c}_4$ for the four neurons are separated by $90°$, and they collectively form a Cartesian coordinate system. (B) The root-mean-square error in the wind direction determined by vector decoding of the firing rates of four cercal interneurons. These results were obtained through simulation by randomly generating interneuron responses to a variety of wind directions, with the average values and trial-to-trial variability of the firing rates matched to the experimental data. The generated rates were then decoded using equation 3.22 and compared to the wind direction used to generate them. (B adapted from Salinas and Abbott, 1994.)

fact that one of the neurons responds maximally; rather, they arise because the two neurons with tuning curves adjacent to the maximally responding neuron are most sensitive to wind direction at these points.

As discussed in chapter 1, tuning curves of certain neurons in the primary motor cortex (M1) of the monkey can be described by cosine functions of arm movement direction. Thus, a vector decomposition similar to that of the cercal system appears to take place in M1. Many M1 neurons have nonzero offset rates, r_0, so they can represent the cosine function over most or all of its range. When an arm movement is made in the direction represented by a vector of unit length, $\vec{v}$, the average firing rates for such an M1 neuron, labeled by an index a (assuming that it fires over the entire range of angles), can be written as

$$\left(\frac{\langle r \rangle - r_0}{r_{max}}\right)_a = \left(\frac{f(s) - r_0}{r_{max}}\right)_a = \vec{v} \cdot \vec{c}_a \,, \qquad (3.23)$$

where $\vec{c}_a$ is the preferred-direction vector that defines the selectivity of the neuron. Because these firing rates represent the full cosine function, it would, in principle, be possible to encode all movement directions in three dimensions using just three neurons. Instead, many thousands of M1 neurons have arm-movement-related tuning curves, resulting in a highly redundant representation. Of course, these neurons encode additional movement-related quantities; for example, their firing rates depend on the initial position of the arm relative to the body as well as on movement ve-

locity and acceleration. This complicates the interpretation of their activity as reporting movement direction in a particular coordinate system.

Unlike the cercal interneurons, M1 neurons do not have orthogonal preferred directions that form a Cartesian coordinate system. Instead, the preferred directions of the neurons appear to point in all directions with roughly equal probability. If the projection axes are not orthogonal, the Cartesian sum of equation 3.22 is not the correct way to reconstruct $\vec{v}$. Nevertheless, if the preferred directions point uniformly in all directions and the number of neurons N is sufficiently large, the population vector

$$\vec{v}_{\mathrm{pop}} = \sum_{a=1}^{N} \left(\frac{r - r_0}{r_{\max}} \right)_a \vec{c}_a \tag{3.24}$$

will, on average, point in a direction parallel to the arm movement direction vector $\vec{v}$. If we average equation 3.24 over trials and use equation 3.23, we find

$$\langle \vec{v}_{\mathrm{pop}} \rangle = \sum_{a=1}^{N} (\vec{v} \cdot \vec{c}_a) \vec{c}_a \,. \tag{3.25}$$

We leave as an exercise the proof that $\langle \vec{v}_{\mathrm{pop}} \rangle$ is approximately parallel to $\vec{v}$ if a large enough number of neurons is included in the sum, and if their preferred-direction vectors point randomly in all directions with equal probability. Later in this chapter, we discuss how corrections can be made if the distribution of preferred directions is not uniform or the number of neurons is not large. The population vectors constructed from equation 3.24 on the basis of responses of neurons in primary motor cortex, recorded while a monkey performed a reaching task, are compared with the actual directions of arm movements in figure 3.6.

Optimal Decoding Methods

The vector method is a simple decoding method that can perform quite well in certain cases, but it is neither a general nor an optimal way to reconstruct a stimulus from the firing rates of a population of neurons. In this section, we discuss two methods that can, by some measure, be considered optimal. These are called Bayesian inference and maximum a posteriori (MAP) inference. We also discuss a special case of MAP called maximum likelihood (ML) inference. The Bayesian approach involves finding the minimum of a loss function that expresses the cost of estimation errors. MAP inference and ML inference generally produce estimates that are as accurate, in terms of the variance of the estimate, as any that can be achieved by a wide class of estimation methods (so-called unbiased estimates), at least when large numbers of neurons are used in the decoding. Bayesian and MAP estimates use the conditional probability that a stimulus parameter takes a value between s and $s + \Delta s$, given that the set of N encoding neurons fired at rates given by $\mathbf{r}$. The probability density
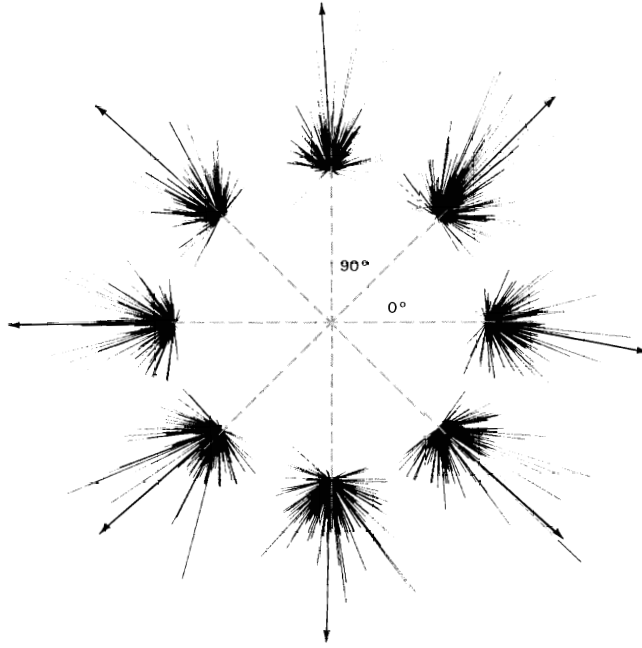
Figure 3.6 Comparison of population vectors with actual arm movement directions. Results are shown for eight different movement directions. Actual arm movement directions are radially outward at angles that are multiples of 45°. The groups of lines without arrows show the preferred-direction vectors of the recorded neurons multiplied by their firing rates. Vector sums of these terms for each movement direction are indicated by the arrows. The fact that the arrows point approximately radially outward shows that the population vector reconstructs the actual movement direction fairly accurately. (Figure adapted from Kandel et al., 1991, based on data from Kalaska et al., 1983.)

needed for a continuous stimulus parameter, $p[s|\mathbf{r}]$, can be obtained from the encoding probability density $p[\mathbf{r}|s]$ by the continuous version of Bayes theorem (equation 3.3),

$$p[s|\mathbf{r}] = \frac{p[\mathbf{r}|s]p[s]}{p[\mathbf{r}]} \ .$$

(3.26)

A disadvantage of these methods is that extracting $p[s|\mathbf{r}]$ from experimental data can be difficult. In contrast, the vector method only requires us to know the preferred stimulus values of the encoding neurons.

Bayesian inference

As mentioned in the previous paragraph, Bayesian inference is based on the minimization of a particular loss function $L(s, s_{\text{bayes}})$ that quantifies the "cost" of reporting the estimate s_{bayes} when the correct answer is s. The loss function provides a way of defining the optimality criterion for decoding analogous to the loss computation discussed previously for optimal discrimination. The value of s_{bayes} is chosen to minimize the expected loss averaged over all stimuli for a given set of rates, that is, to minimize the function $\int ds\, L(s, s_{\text{bayes}}) p[s|\mathbf{r}]$. If the loss function is the squared differ-
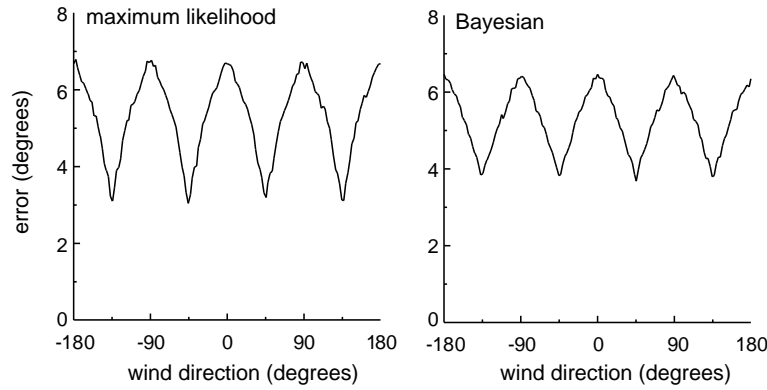
Figure 3.7 Maximum likelihood and Bayesian estimation errors for the cricket cercal system. ML and Bayesian estimates of the wind direction were compared with the actual stimulus value for a large number of simulated firing rates. Firing rates were generated as for figure 3.5B. The error shown is the root-mean-squared difference between the estimated and actual stimulus angles. (Adapted from Salinas and Abbott, 1994.)

ence between the estimate and the true value, $L(s, s_{\text{bayes}}) = (s - s_{\text{bayes}})^2$, the estimate that minimizes the expected loss is the mean

$$s_{\text{bayes}} = \int ds\, p[s|\mathbf{r}]s \,. \tag{3.27}$$

If the loss function is the absolute value of the difference, $L(s, s_{\text{bayes}}) = |s - s_{\text{bayes}}|$, then s_{bayes} is the median rather than the mean of the distribution $p[s|\mathbf{r}]$.

Maximum a posteriori (MAP) inference does not involve a loss function but instead simply chooses the stimulus value, s_{MAP}, that maximizes the conditional probability density of the stimulus, $p[s_{\text{MAP}}|\mathbf{r}]$. The MAP approach is thus to choose as the estimate s_{MAP} the most likely stimulus value for a given set of rates. If the prior or stimulus probability density $p[s]$ is independent of s, then $p[s|\mathbf{r}]$ and $p[\mathbf{r}|s]$ have the same dependence on s, because the factor $p[s]/p[\mathbf{r}]$ in equation 3.26 is independent of s. In this case, the MAP algorithm is equivalent to maximizing the likelihood function, that is, choosing s_{ML} to maximize $p[\mathbf{r}|s_{\text{ML}}]$, which is called maximum likelihood (ML) inference.

MAP inference

ML inference

Previously we applied the vector decoding method to the cercal system of the cricket. Figure 3.7 shows the root-mean-squared difference between the true and estimated wind directions for the cercal system, using ML and Bayesian methods. For the cercal interneurons, the response probability density $p[\mathbf{r}|s]$ is a product of four Gaussians with means and variances given by the data points and error bars in figure 3.4. The Bayesian estimate in figure 3.7 is based on the squared-difference loss function. Both estimates use a constant stimulus probability density $p[s]$, so the ML and MAP estimates are identical. The maximum likelihood estimate is either more or less accurate than the Bayesian estimate, depending on the angle.

The Bayesian result has a slightly smaller average error across all angles. The dips in the error curves in figure 3.7, as in the curve of figure 3.5B, appear at angles where one tuning curve peaks and two others rise from threshold (see figure 3.4). As in figure 3.5B, these dips are due to the two neurons responding near threshold, not to the maximally responding neuron. They occur because neurons are most sensitive at points where their tuning curves have maximum slopes, which in this case is near threshold (see figure 3.11).

Comparing these results with figure 3.5B shows the improved performance of these methods relative to the vector method. The vector method performs extremely well for this system, so the degree of improvement is not large. This is because the cercal responses are well described by cosine functions and their preferred directions are $90°$ apart. Much more dramatic differences occur when the tuning curves are not cosines or the preferred stimulus directions are not perpendicular.

Up to now, we have considered the decoding of a direction angle. We now turn to the more general case of decoding an arbitrary continuous stimulus parameter. An instructive example is provided by an array of N neurons with preferred stimulus values distributed uniformly across the full range of possible stimulus values. An example of such an array for Gaussian tuning curves,

$$f_a(s) = r_{\max} \exp\left(-\frac{1}{2}\left(\frac{s - s_a}{\sigma_a}\right)^2\right), \tag{3.28}$$

is shown in figure 3.8. In this example, each neuron has a tuning curve with a different preferred value s_a and potentially a different width σ_a (although all the curves in figure 3.8 have the same width). If the tuning curves are evenly and densely distributed across the range of s values, the sum of all tuning curves $\sum f_a(s)$ is approximately independent of s. The roughly flat line in figure 3.8 is proportional to this sum. The constancy of the sum over tuning curves will be useful in the following analysis.

Tuning curves give the mean firing rates of the neurons across multiple trials. In any single trial, measured firing rates will vary from their mean values. To implement the Bayesian, MAP, or ML approach, we need to know the conditional firing-rate probability density $p[\mathbf{r}|s]$ that describes this variability. We assume that the firing rate r_a of neuron a is determined by counting n_a spikes over a trial of duration T (so that $r_a = n_a/T$), and that the variability can be described by the homogeneous Poisson model discussed in chapter 1. In this case, the probability of stimulus s evoking $n_a = r_a T$ spikes, when the average firing rate is $\langle r_a \rangle = f_a(s)$, is given by (see chapter 1)

$$P[r_a|s] = \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} \exp(-f_a(s)T). \tag{3.29}$$

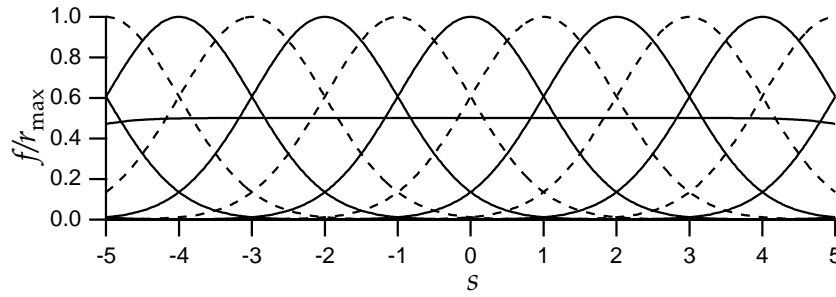If we assume that each neuron fires independently, the firing-rate proba-

Figure 3.8 An array of Gaussian tuning curves spanning stimulus values from -5 to 5. The peak values of the tuning curves fall on the integer values of s and the tuning curves all have $\sigma_a = 1$. For clarity, the curves are drawn alternately with dashed and solid lines. The approximately flat curve with value near 0.5 is 1/5 the sum of the tuning curves shown, indicating that this sum is approximately independent of s.

bility for the population is the product of the individual probabilities,

$$P[\mathbf{r}|s] = \prod_{a=1}^{N} \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} \exp\left(-f_a(s)T\right). \tag{3.30}$$

The assumption of independence simplifies the calculations considerably.

The filled circles in figure 3.9 show a set of randomly generated firing rates for the array of Gaussian tuning curves in figure 3.8 for $s = 0$. This figure also illustrates a useful way of visualizing population responses: plotting the responses as a function of the preferred stimulus values. The dashed curve in figure 3.9 is the tuning curve for the neuron with $s_a = 0$. Because the tuning curves are functions of $|s - s_a|$, the values of the dashed curve at $s_a = -5, -4, \ldots, 5$ are the mean activities of the cells with preferred values at those locations for a stimulus at $s = 0$.

To apply the ML estimation algorithm, we only need to consider the terms in $P[\mathbf{r}|s]$ that depend on s. Because equation 3.30 involves a product, it is convenient to take its logarithm and write

$$\ln P[\mathbf{r}|s] = T \sum_{a=1}^{N} r_a \ln\left(f_a(s)\right) + \ldots, \tag{3.31}$$

where the ellipsis represents terms that are independent or approximately independent of s, including, as discussed above, $\sum f_a(s)$. Because maximizing a function and maximizing its logarithm are equivalent, we can use the logarithm of the conditional probability in place of the actual probability in ML decoding.

The ML estimated stimulus, s_{ML}, is the stimulus that maximizes the right side of equation 3.31. Setting the derivative to 0, we find that s_{ML} is deter-
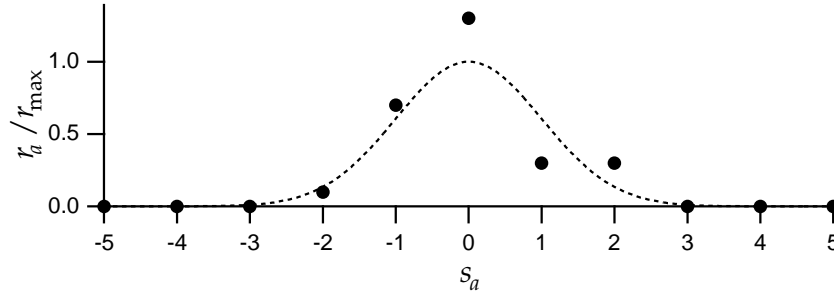
Figure 3.9 Simulated responses of 11 neurons with the Gaussian tuning curves shown in figure 3.8 to a stimulus value of 0. Firing rates for a single trial, generated using the Poisson model, are plotted as a function of the preferred-stimulus values of the different neurons in the population (filled circles). The dashed curve shows the tuning curve for the neuron with $s_a = 0$. Its heights at integer values of s_a are the average responses of the corresponding cells. It is possible to have $r_a > r_{\max}$ (point at $s_a = 0$) because $r_{\max}$ is the maximum average firing rate, not the maximum firing rate.

mined by

$$\sum_{a=1}^{N} r_a \frac{f_a'(s_{\mathrm{ML}})}{f_a(s_{\mathrm{ML}})} = 0\,, \tag{3.32}$$

where the prime denotes a derivative. If the tuning curves are the Gaussians of equation 3.28, this equation can be solved explicitly using the result $f_a'(s)/f_a(s) = (s_a - s)/\sigma_a^2$,

$$s_{\mathrm{ML}} = \frac{\sum r_a s_a/\sigma_a^2}{\sum r_a/\sigma_a^2}\,. \tag{3.33}$$

If all the tuning curves have the same width, this reduces to

$$s_{\mathrm{ML}} = \frac{\sum r_a s_a}{\sum r_a}\,, \tag{3.34}$$

which is a simple estimation formula with an intuitive interpretation as the firing-rate weighted average of the preferred values of the encoding neurons. The numerator of this expression is reminiscent of the population vector.

Although equation 3.33 gives the ML estimate for a population of neurons with Poisson variability, it has some undesirable properties as a decoding algorithm. Consider a neuron with a preferred stimulus value s_a that is much greater than the actual stimulus value s. Because $s_a \gg s$, the average firing rate of this neuron is essentially 0. For a Poisson distribution, zero rate implies zero variability. If, however, this neuron fires one or more spikes on a trial due to a non-Poisson source of variability, this will cause a large error in the estimate because of the large weighting factor s_a.

The MAP estimation procedure is similar in spirit to the ML approach, but the MAP estimate, s_{MAP}, may differ from s_{ML} if the probability density $p[s]$ depends on s. The MAP algorithm allows us to include prior knowledge about the distribution of stimulus values in the decoding estimate. As noted above, if the $p[s]$ is constant, the MAP and ML estimates are identical. In addition, if many neurons are observed, or if a small number of neurons is observed over a long trial period, even a nonconstant stimulus distribution has little effect and $s_{\text{MAP}} \approx s_{\text{ML}}$.

The MAP estimate is computed from the distribution $p[s|\mathbf{r}]$ determined by Bayes theorem. In terms of the logarithms of the probabilities, $\ln p[s|\mathbf{r}] = \ln P[\mathbf{r}|s] + \ln p[s] - \ln P[\mathbf{r}]$. The last term in this expression is independent of s and can be absorbed into the ignored s-independent terms, so we can write, as in equation 3.31,

$$\ln p[s|\mathbf{r}] = T \sum_{a=1}^{N} r_a \ln \left(f_a(s) \right) + \ln p[s] + \ldots . \tag{3.35}$$

Maximizing this determines the MAP estimate,

$$T \sum_{a=1}^{N} \frac{r_a f_a'(s_{\text{MAP}})}{f_a(s_{\text{MAP}})} + \frac{p'[s_{\text{MAP}}]}{p[s_{\text{MAP}}]} = 0 . \tag{3.36}$$

If the stimulus or prior distribution is itself Gaussian with mean s_{prior} and variance σ_{prior}, and we use the Gaussian array of tuning curves, equation 3.36 yields

$$s_{\text{MAP}} = \frac{T \sum r_a s_a / \sigma_a^2 + s_{\text{prior}} / \sigma_{\text{prior}}^2}{T \sum r_a / \sigma_a^2 + 1 / \sigma_{\text{prior}}^2} . \tag{3.37}$$

Figure 3.10 compares the conditional stimulus probability densities $p[s|\mathbf{r}]$ for a constant stimulus distribution (solid curve) and for a Gaussian stimulus distribution with $s_{\text{prior}} = -2$ and $\sigma_{\text{prior}} = 1$, using the firing rates given by the filled circles in figure 3.9. If the stimulus distribution is constant, $p[s|\mathbf{r}]$ peaks near the true stimulus value of 0. The effect of a nonconstant stimulus distribution is to shift the curve toward the value -2, where the stimulus probability density has its maximum, and to decrease its width by a small amount. The estimate is shifted to the left because the prior distribution suggests that the stimulus is more likely to take negative values than positive ones, independent of the evoked response. The decreased width is due to the added information that the prior distribution provides. The curves in figure 3.10 can be computed from equations 3.28 and 3.35 as Gaussians with variances $1/(T \sum r_a / \sigma_a^2)$ (constant prior) and $1/(T \sum r_a / \sigma_a^2 + 1 / \sigma_{\text{prior}}^2)$ (Gaussian prior).

The accuracy with which an estimate s_{est} describes a stimulus s can be characterized by two important quantities, its bias $b_{\text{est}}(s)$ and its variance $\sigma_{\text{est}}^2(s)$. The bias is the difference between the average of s_{est} across trials that use the stimulus s and the true value of the stimulus (i.e., s), *bias*

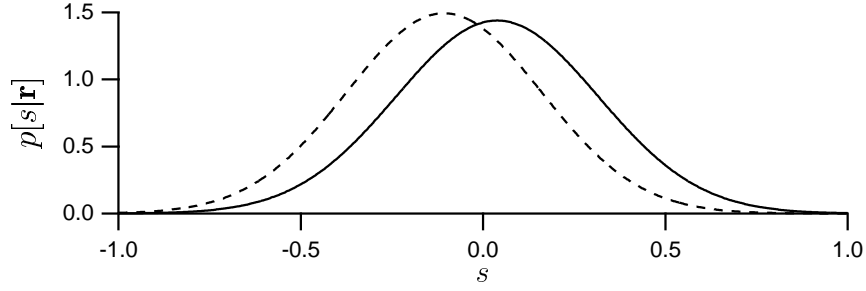$$b_{\text{est}}(s) = \langle s_{\text{est}} \rangle - s . \tag{3.38}$$

Figure 3.10 Probability densities for the stimulus, given the firing rates shown in figure 3.9 and assuming the tuning curves of figure 3.8. The solid curve is $p[s|\mathbf{r}]$ when the prior distribution of stimulus values is constant and the true value of the stimulus is $s = 0$. The dashed curve is for a Gaussian prior distribution with a mean of -2 and variance of 1, again with the true stimulus being $s = 0$. The peaks of the solid and dashed curves are at $s = 0.0385$ and $s = -0.107$, respectively.

Note that the bias depends on the true value of the stimulus. An estimate is termed unbiased if $b_{\text{est}}(s) = 0$ for all stimulus values.

variance   The variance of the estimator, which quantifies how much the estimate varies about its mean value, is defined as

$$\sigma^2_{\text{est}}(s) = \langle (s_{\text{est}} - \langle s_{\text{est}} \rangle)^2 \rangle . \tag{3.39}$$

The bias and variance can be used to compute the trial-average squared estimation error, $\langle (s_{\text{est}} - s)^2 \rangle$. This is a measure of the spread of the estimated values about the true value of the stimulus. Because $s = \langle s_{\text{est}} \rangle - b_{\text{est}}(s)$, we

estimation error   can write the squared estimation error as

$$\langle (s_{\text{est}} - s)^2 \rangle = \langle (s_{\text{est}} - \langle s_{\text{est}} \rangle + b_{\text{est}}(s))^2 \rangle = \sigma^2_{\text{est}}(s) + b^2_{\text{est}}(s) . \tag{3.40}$$

In other words, the average squared estimation error is the sum of the variance and the square of the bias. For an unbiased estimate, the average squared estimation error is equal to the variance of the estimator.

Fisher Information

Decoding can be used to limit the accuracy with which a neural system encodes the value of a stimulus parameter because the encoding accuracy cannot exceed the accuracy of an optimal decoding method. Of course, we must be sure that the decoding technique used to establish such a bound is truly optimal, or else the result will reflect the limitations of the decoding procedure, not bounds on the neural system being studied. The Fisher information is a quantity that provides one such measure of encoding accuracy. Through a bound known as the Cramér-Rao bound, the Fisher information limits the accuracy with which any decoding scheme can extract an estimate of an encoded quantity.

Cramér-Rao bound   The Cramér-Rao bound limits the variance of any estimate s_{est} according

to (appendix B)

$$\sigma_{\text{est}}^2(s) \geq \frac{\left(1 + b_{\text{est}}'(s)\right)^2}{I_{\text{F}}(s)} \, , \tag{3.41}$$

where $b_{\text{est}}'(s)$ is the derivative of $b_{\text{est}}(s)$. If we assume here that the firing rates take continuous values and that their distribution in response to a stimulus s is described by the conditional probability density $p[\mathbf{r}|s]$, the quantity $I_{\text{F}}(s)$ in equation 3.41 is the Fisher information of the firing-rate distribution, which is related to $p[\mathbf{r}|s]$ (assuming the latter is sufficiently smooth) by

Fisher information

$$I_{\text{F}}(s) = \left\langle -\frac{\partial^2 \ln p[\mathbf{r}|s]}{\partial s^2} \right\rangle = \int d\mathbf{r} \, p[\mathbf{r}|s] \left(-\frac{\partial^2 \ln p[\mathbf{r}|s]}{\partial s^2} \right) . \tag{3.42}$$

The reader can verify that the Fisher information can also be written as

$$I_{\text{F}}(s) = \left\langle \left(\frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \right)^2 \right\rangle = \int d\mathbf{r} \, p[\mathbf{r}|s] \left(\frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \right)^2 . \tag{3.43}$$

The Cramér-Rao bound sets a limit on the accuracy of any unbiased estimate of the stimulus. When $b_{\text{est}}(s) = 0$, equation 3.40 indicates that the average squared estimation error is equal to σ_{est}^2 and, by equation 3.41, this satisfies the bound $\sigma_{\text{est}}^2 \geq 1/I_{\text{F}}(s)$. Provided that we restrict ourselves to unbiased decoding schemes, the Fisher information sets an absolute limit on decoding accuracy, and it thus provides a useful limit on encoding accuracy. Although imposing zero bias on the decoding estimate seems reasonable, the restriction is not trivial. In general, minimizing the decoding error in equation 3.40 involves a trade-off between minimizing the bias and minimizing the variance of the estimator. In some cases, biased schemes may produce more accurate results than unbiased ones. For a biased estimator, the average squared estimation error and the variance of the estimate are not equal, and the estimation error can be either larger or smaller than $1/I_{\text{F}}(s)$.

The limit on decoding accuracy set by the Fisher information can be attained by a decoding scheme we have studied, the maximum likelihood method. In the limit of large numbers of encoding neurons, and for most firing-rate distributions, the ML estimate is unbiased and saturates the Cramér-Rao bound. In other words, the variance of the ML estimate is given asymptotically (for large N) by $\sigma_{\text{ML}}^2(s) = 1/I_{\text{F}}(s)$. Any unbiased estimator that saturates the Cramér-Rao lower bound is called efficient. Furthermore, $I_{\text{F}}(s)$ grows linearly with N, and the ML estimate obeys a central limit theorem, so that $N^{1/2}(s_{\text{ML}} - s)$ is Gaussian distributed with a variance that is independent of N in the large N limit. Finally, in the limit $N \to \infty$, the ML estimate is asymptotically consistent, in the sense that $P[|s_{\text{ML}} - s| > \epsilon] \to 0$ for any $\epsilon > 0$.

efficiency

*asymptotic
consistency*

As equation 3.42 shows, the Fisher information is a measure of the expected curvature of the log likelihood at stimulus value s. Curvature is

important because the likelihood is expected to be at a maximum near the true stimulus value s that caused the responses. If the likelihood is very curved, and thus the Fisher information is large, responses typical for the stimulus s are much less likely to occur for slightly different stimuli. Therefore, the typical response provides a strong indication of the value of the stimulus. If the likelihood is fairly flat, and thus the Fisher information is small, responses common for s are likely to occur for slightly different stimuli as well. Thus, the response does not as clearly determine the stimulus value. The Fisher information is purely local in the sense that it does not reflect the existence of stimulus values completely different from s that are likely to evoke the same responses as those evoked by s itself. However, this does not happen for the sort of simple population codes we consider. Shannon's mutual information measure, discussed in chapter 4, takes such possibilities into account.

The Fisher information for a population of neurons with uniformly arrayed tuning curves (the Gaussian array in figure 3.8, for example) and Poisson statistics can be computed from the conditional firing-rate probability in equation 3.30. Because the spike-count rate is described here by a probability rather than a probability density, we use the discrete analog of equation 3.42,

$$I_{\mathrm{F}}(s) = \left\langle -\frac{\partial^2 \ln P[\mathbf{r}|s]}{\partial s^2} \right\rangle = T \sum_{a=1}^{N} \left(\langle r_a \rangle \left(\left(\frac{f_a'(s)}{f_a(s)} \right)^2 - \frac{f_a''(s)}{f_a(s)} \right) + f_a''(s) \right).$$

(3.44)

Note that we have used the full expression, equation 3.30, in deriving this result, not the truncated form of $\ln P[\mathbf{r}|s]$ in equation 3.31. We next make the replacement $\langle r_a \rangle = f_a(s)$, producing the final result

$$I_{\mathrm{F}}(s) = T \sum_{a=1}^{N} \frac{(f_a'(s))^2}{f_a(s)}.$$

(3.45)

In this expression, each neuron contributes an amount to the Fisher information proportional to the square of its tuning curve slope and inversely proportional to the average firing rate for the particular stimulus value being estimated. Highly sloped tuning curves give firing rates that are sensitive to the precise value of the stimulus. Figure 3.11 shows the contribution to the sum in equation 3.45 from a single neuron with a Gaussian tuning curve, the neuron with $s_a = 0$ in figure 3.8. For comparison purposes, a dashed curve proportional to the tuning curve is also plotted. Note that the Fisher information vanishes for the stimulus value that produces the maximum average firing rate, because $f_a'(s) = 0$ at this point. The firing rate of a neuron at the peak of its tuning curve is relatively unaffected by small changes in the stimulus. Individual neurons carry the most Fisher information in regions of their tuning curves where average firing rates are rapidly varying functions of the stimulus value, not where the firing rate is highest.

The Fisher information can be used to derive an interesting result on the optimal widths of response tuning curves. Consider a population of neu-
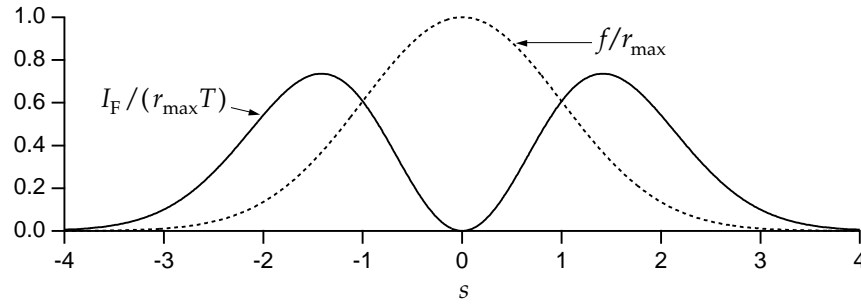
Figure 3.11 The Fisher information for a single neuron with a Gaussian tuning curve with $s = 0$ and $\sigma_a = 1$, and Poisson variability. The Fisher information (solid curve) has been divided by $r_{max}T$, the peak firing rate of the tuning curve times the duration of the trial. The dashed curve shows the tuning curve scaled by r_{max}. Note that the Fisher information is greatest where the slope of the tuning curve is highest, and vanishes at $s = 0$, where the tuning curve peaks.

rons with tuning curves of identical shapes, distributed evenly over a range of stimulus values as in figure 3.8. Equation 3.45 indicates that the Fisher information will be largest if the tuning curves of individual neurons are rapidly varying (making the square of their derivatives large), and if many neurons respond (making the sum over neurons large). For typical neuronal response tuning curves, these two requirements are in conflict with one another. If the population of neurons has narrow tuning curves, individual neural responses are rapidly varying functions of the stimulus, but few neurons respond. Broad tuning curves allow many neurons to respond, but the individual responses are not as sensitive to the stimulus value. To determine whether narrow or broad tuning curves produce the more accurate encodings, we consider a dense distribution of Gaussian tuning curves, all with $\sigma_a = \sigma_r$. Using such curves in equation 3.45, we find

$$I_F(s) = T \sum_{a=1}^{N} \frac{r_{max}(s - s_a)^2}{\sigma_r^4} \exp\left(-\frac{1}{2}\left(\frac{s - s_a}{\sigma_r}\right)^2\right). \qquad (3.46)$$

This expression can be approximated by replacing the sum over neurons with an integral over their preferred stimulus values and multiplying by a density factor ρ_s. The factor ρ_s is the density with which the neurons cover the range of stimulus values, and it is equal to the number of neurons with preferred stimulus values lying within a unit range of s values. Replacing the sum over a with an integral over a continuous preferred stimulus parameter ξ (which replaces s_a), we find

sums→integrals

$$I_F(s) \approx \rho_s T \int_{-\infty}^{\infty} d\xi \frac{r_{max}(s - \xi)^2}{\sigma_r^4} \exp\left(-\frac{1}{2}\left(\frac{s - \xi}{\sigma_r}\right)^2\right)$$

$$= \frac{\sqrt{2\pi}\,\rho_s \sigma_r r_{max} T}{\sigma_r^2} \,. \qquad (3.47)$$

We have expressed the final result in this form because the number of neurons that respond to a given stimulus value is roughly $\rho_s \sigma_r$, and the Fisher

information is proportional to this number divided by the square of the tuning curve width. Combining these factors, the Fisher information is inversely proportional to σ_r, and the encoding accuracy increases with narrower tuning curves.

The advantage of using narrow tuning curves goes away if the stimulus is characterized by more than one parameter. Consider a stimulus with D parameters and suppose that the response tuning curves are products of identical Gaussians for each of these parameters. If the tuning curves cover the D-dimensional space of stimulus values with a uniform density ρ_s, the number of responding neurons for any stimulus value is proportional to $\rho_s \sigma_r^D$ and, using the same integral approximation as in equation 3.47, the Fisher information is

$$I_{\mathrm{F}} = \frac{(2\pi)^{D/2} \rho_s \sigma_r^D r_{\max} T}{D \sigma_r^2} = \frac{(2\pi)^{D/2} \rho_s \sigma_r^{D-2} r_{\max} T}{D} \,. \qquad (3.48)$$

This equation, which reduces to the result given above if $D = 1$, allows us to examine the effect of tuning curve width on encoding accuracy. The trade-off between the encoding accuracy of individual neurons and the number of responding neurons depends on the dimension of the stimulus space. Narrowing the tuning curves (making σ_r smaller) increases the Fisher information for $D = 1$, decreases it for $D > 2$, and has no impact if $D = 2$.

Optimal Discrimination

In the first part of this chapter, we considered discrimination between two values of a stimulus. An alternative to the procedures discussed there is simply to decode the responses and discriminate on the basis of the estimated stimulus values. Consider the case of discriminating between s and $s + \Delta s$ for small Δs. For large N, the average value of the difference between the ML estimates for the two stimulus values is equal to Δs (because the estimate is unbiased) and the variance of each estimate (for small Δs) is $1/I_{\mathrm{F}}(s)$. Thus, the discriminability, defined in equation 3.4, for the ML-based test is

ML discriminability

$$d' = \Delta s \sqrt{I_{\mathrm{F}}(s)} \,. \qquad (3.49)$$

The larger the Fisher information, the higher the discriminability. We leave as an exercise the proof that for small Δs, this discriminability is the same as that of the likelihood ratio test $Z(\mathbf{r})$ defined in equation 3.19.

Discrimination by ML estimation requires maximizing the likelihood, and this may be computationally challenging. The likelihood ratio test described previously may be simpler, especially for Poisson variability, because, for small Δs, the likelihood ratio test Z defined in equation 3.19 is a linear function of the firing rates,

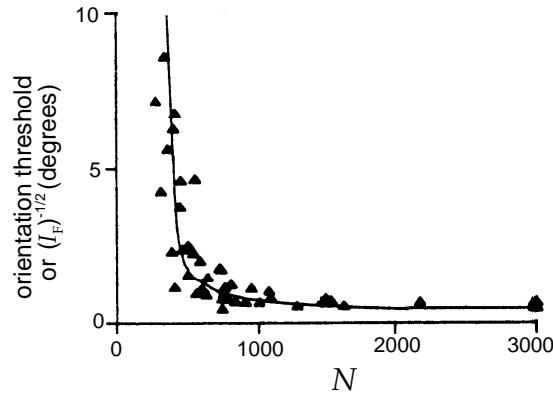$$Z = T \sum_{a=1}^{N} r_a \frac{f_a'(s)}{f_a(s)} \,. \qquad (3.50)$$

Figure 3.12 Comparison of Fisher information and discrimination thresholds for orientation tuning. The solid curve is the minimum standard deviation of an estimate of orientation angle from the Cramér-Rao bound, plotted as a function of the number of neurons (N) involved in the estimation. The triangles are data points from an experiment that determined the threshold for discrimination of the orientation of line images by human subjects as a function of line length and eccentricity. An effective number of neurons involved in the task was estimated for the different line lengths and eccentricities, using the cortical magnification factor discussed in chapter 2. (Adapted from Paradiso, 1988.)

Figure 3.12 shows an interesting comparison of the Fisher information for orientation tuning in the primary visual cortex with human orientation discrimination thresholds. Agreement like this can occur for difficult tasks, like discrimination at threshold, where the performance of a subject may be limited by basic constraints on neuronal encoding accuracy.

3.4 Spike-Train Decoding

The decoding methods we have considered estimate or discriminate static stimulus values on the basis of spike-count firing rates. Spike-count firing rates do not provide sufficient information for reconstructing a stimulus that varies during the course of a trial. Instead, we can estimate such a stimulus from the sequence of firing times t_i for $i = 1, 2, \ldots, n$ of the spikes that it evokes. One method for doing this is similar to the Wiener kernel approach used to estimate the firing rate from the stimulus in chapter 2, and to approximate a firing rate using a sliding window function in chapter 1. For simplicity, we restrict our discussion to the decoding of a single neuron. We assume, as we did in chapter 2, that the time average of the stimulus being estimated is 0.

In spike-train decoding, we attempt to construct an estimate of the stimulus at time t from the sequence of spikes evoked up to that time. There are paradoxical aspects of this procedure. The firing of an action potential at time t_i is only affected by the stimulus $s(t)$ prior to that time, $t < t_i$, and yet, in spike decoding, we attempt to extract information from this
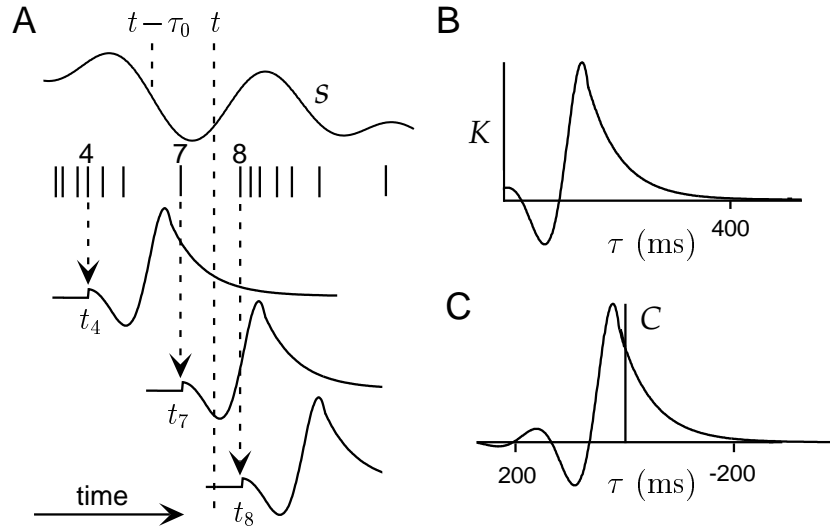
Figure 3.13 Illustration of spike-train decoding. (A) The top trace denotes a stimulus that evokes the spike train appearing below it (second trace from top). At time t an estimate is being made of the stimulus at time $t - \tau_0$. The estimate is obtained by summing the values of the kernels where they cross the dashed line labeled t, for spikes up to and including spike 7. Two such kernels are shown in the third and fourth traces from the top. The real estimate is obtained by summing similar contributions from all of the spikes. The kernel is 0 for negative values of its argument, so spikes for $i \geq 8$ do not contribute to the estimate at this time (e.g., fifth trace from top). (B) The kernel used in A. This has been truncated to zero value for negative values of τ. (C) The spike-triggered average corresponding to the kernel in B, assuming no spike-train correlations. Note that C has been plotted with the τ axis reversed, following the convention established in chapter 1. With this convention, K in panel B is simply a shifted and truncated version of the curve appearing here. In this case $\tau_0 = 160$ ms.

action potential about the value of the stimulus at a later time $t > t_i$. That is, the evoked spikes tell us about the past behavior of the stimulus and, in spike decoding, we attempt to use this information to predict the current stimulus value. Clearly, this requires that the stimulus have some form of temporal correlation so that past behavior provides information about the current stimulus value. To make the decoding task easier, we

prediction delay τ_0 can introduce a prediction delay, τ_0, and attempt to construct, from spikes occurring prior to time t, an estimate of the stimulus at time $t - \tau_0$ (see figure 3.13A). Such a delayed estimate uses a combination of spikes that could have been fired in response to the stimulus $s(t - \tau_0)$ being estimated (those for which $t - \tau_0 < t_i < t$; spike 7 in figure 3.13A), and spikes that occurred too early to be affected by the value of $s(t - \tau_0)$ (those for which $t_i < t - \tau_0$; spikes 1-6 in figure 3.13A), but that can contribute to its estimation on the basis of stimulus correlations. The estimation task gets easier as τ_0 is increased, but this delays the decoding and makes the result less behaviorally relevant. We will consider decoding with an arbitrary delay and later discuss how to set a specific value for τ_0.

stimulus estimate The stimulus estimate is constructed as a linear sum over all spikes. A

spike occurring at time t_i contributes a kernel $K(t - t_i)$, and the total estimate is obtained by summing over all spikes,

$$s_{\text{est}}(t - \tau_0) = \sum_{i=1}^{n} K(t - t_i) - \langle r \rangle \int_{-\infty}^{\infty} d\tau \, K(\tau) . \qquad (3.51)$$

The last term, with $\langle r \rangle = \langle n \rangle / T$ the average firing rate over the trial, is included to impose the condition that the time average of s_{est} is 0, in agreement with the time-average condition on s. The sum in equation 3.51 includes all spikes, so the constraint that only those spikes occurring prior to the time t (spikes 1-7 in figure 3.13A) should be included must be imposed by requiring $K(t - t_i) = 0$ for $t - t_i \leq 0$. A kernel satisfying this constraint is termed causal. We ignore the causality constraint for now and construct an acausal kernel, but we will return to issues of causality later in the discussion. Figure 3.13A shows how spikes contribute to a stimulus estimate, using the kernel shown in figure 3.13B.

Equation 3.51 can be written in a compact way by using the neural response function $\rho(t) = \sum \delta(t - t_i)$ introduced in chapter 1,

$$s_{\text{est}}(t - \tau_0) = \int_{-\infty}^{\infty} d\tau \, (\rho(t - \tau) - \langle r \rangle) \, K(\tau) . \qquad (3.52)$$

Using this form of the estimate, the construction of the optimal kernel K proceeds very much like the construction of the optimal kernel for predicting firing rates in chapter 2. We choose K so that the squared difference between the estimated stimulus and the actual stimulus, averaged over both time and trials,

$$\frac{1}{T} \int_{0}^{T} dt \left\langle \left(\int_{-\infty}^{\infty} d\tau \, (\rho(t - \tau) - \langle r \rangle) \, K(\tau) - s(t - \tau_0) \right)^2 \right\rangle , \qquad (3.53)$$

is minimized. The calculation proceeds as in appendix A of chapter 2, and the result is that K obeys the equation

$$\int_{-\infty}^{\infty} d\tau' \, Q_{\rho\rho}(\tau - \tau') K(\tau') = Q_{rs}(\tau - \tau_0) , \qquad (3.54)$$

where $Q_{\rho\rho}$ is the spike-train autocorrelation function,

$$Q_{\rho\rho}(\tau - \tau') = \frac{1}{T} \int_{0}^{T} dt \, \langle (\rho(t - \tau) - \langle r \rangle)(\rho(t - \tau') - \langle r \rangle) \rangle , \qquad (3.55)$$

as defined in chapter 1. Q_{rs} is the correlation of the firing rate and the stimulus, which is related to the spike-triggered average C, both introduced in chapter 1,

$$Q_{rs}(\tau - \tau_0) = \langle r \rangle C(\tau_0 - \tau) = \frac{1}{T} \left\langle \sum_{i=1}^{n} s(t_i + \tau - \tau_0) \right\rangle . \qquad (3.56)$$

At this point in the derivation of the optimal linear kernel for firing-rate prediction in chapter 2, we chose the stimulus to be uncorrelated so that an

integral equation similar to 3.54 would be simplified. This could always be done because we have complete control over the stimulus in this type of experiment. However, we do not have similar control of the neuron, and must deal with whatever spike-train autocorrelation function it gives us. If the spike train is uncorrelated, which tends to happen at low rates,

$$Q_{\rho\rho}(\tau) = \langle r \rangle \delta(\tau) \,, \tag{3.57}$$

and we find from equation 3.54 that

$$K(\tau) = \frac{1}{\langle r \rangle} Q_{rs}(\tau - \tau_0) = C(\tau_0 - \tau) = \frac{1}{\langle n \rangle} \left\langle \sum_{i=1}^{n} s(t_i + \tau - \tau_0) \right\rangle . \tag{3.58}$$

This is the average value of the stimulus at time $\tau - \tau_0$ relative to the appearance of a spike. Because $\tau - \tau_0$ can be either positive or negative, stimulus estimation, unlike firing-rate estimation, involves both forward and backward correlation and the average values of the stimulus both before and after a spike. Decoding in this way follows a simple rule: every time a spike appears, we replace it with the average stimulus surrounding a spike, shifted by an amount τ_0 (figure 3.13).

The need for either stimulus correlations or a nonzero prediction delay is clear from equation 3.58. Correlations between a spike and subsequent stimuli can arise, in a causal system, only from correlations between the stimulus and itself. If these are absent, as for white noise, $K(\tau)$ will be 0 for $\tau > \tau_0$. For causal decoding, we must also have $K(\tau) = 0$ for $\tau < 0$. Thus, if $\tau_0 = 0$ and the stimulus is uncorrelated, $K(\tau) = 0$ for all values of τ.

optimal kernel When the spike-train autocorrelation function is not a δ function, an acausal solution for K can be expressed as an inverse Fourier transform,

$$K(\tau) = \frac{1}{2\pi} \int d\omega \, \tilde{K}(\omega) \exp(-i\omega\tau) \,, \tag{3.59}$$

where, as shown in appendix C,

$$\tilde{K}(\omega) = \frac{\tilde{Q}_{rs}(\omega) \exp(i\omega\tau_0)}{\tilde{Q}_{\rho\rho}(\omega)} \,. \tag{3.60}$$

Here $\tilde{Q}_{rs}$ and $\tilde{Q}_{\rho\rho}$ are the Fourier transforms of Q_{rs} and $Q_{\rho\rho}$. The numerator in this expression reproduces the expression $Q_{rs}(\tau - \tau_0)$ in equation 3.58. The role of the denominator is to correct for any autocorrelations in the response spike train. Such correlations introduce a bias in the decoding, and the denominator in equation 3.60 corrects for this bias.

If we ignore the constraint of causality, then, because the occurrence of a spike cannot depend on the behavior of a stimulus in the very distant past, we can expect $K(\tau)$ from equation 3.58 to vanish for sufficiently negative values of $\tau - \tau_0$. For most neurons, this will occur for $\tau - \tau_0$ more negative than minus a few hundred ms. The decoding kernel can therefore be made

small for negative values of τ by choosing τ_0 large enough, but this may require a fairly large prediction delay. We can force exact adherence to the causality constraint for $\tau < 0$ by replacing $K(\tau)$ by $\Theta(\tau)K(\tau)$, where   *causality constraint*
$\Theta(\tau)$ is defined such that $\Theta(\tau) = 1$ for $\tau > 0$ and $\Theta(\tau) = 0$ for $\tau < 0$. The causality constraint was imposed in this way in figure 3.13B. When it is multiplied by $\Theta(\tau)$, the restricted K is no longer the optimal decoding kernel, but it may be close to optimal.

Another way of imposing causality on the decoding kernel is to expand $K(\tau)$ as a weighted sum of causal basis functions (functions that vanish for negative arguments and span the space of functions satisfying the causal constraint). The optimal weights are then determined by minimizing the estimation error. This approach has the advantage of producing a truly optimal kernel for any desired value of τ_0. A simpler but nonoptimal approach is to consider a fixed functional form for $K(\tau)$ that vanishes for $\tau \leq 0$ and is characterized by a number of free parameters that can be determined by minimizing the decoding error. Finally, the optimal causal kernel, also called the Wiener-Hopf filter, can be obtained by a technique that involves so-called spectral factorization of $\tilde{Q}_{\rho\rho}(\omega)$.

Figure 3.14 shows an example of spike-train decoding for the H1 neuron of the fly discussed in chapter 2. The top panel gives two reconstruction kernels, one acausal and one causal, and the bottom panel compares the reconstructed stimulus velocity with the actual stimulus velocity. The middle panel in figure 3.14 points out one further wrinkle in the procedure. Flies have two H1 neurons, one on each side of the body, that respond to motion in opposite directions. As is often the case, half-wave rectification prevents a single neuron from encoding both directions of motion. In the experiment described in the figure, rather than recording from both H1 neurons, Bialek et al. (1991) recorded from a single H1 neuron, but presented both the stimulus $s(t)$ and its negative, $-s(t)$. The two rows of spikes in the middle panel show sample traces for each of these presentations. This procedure provides a reasonable approximation of recording both H1 neurons, and produces a neuron/anti-neuron pair of recordings similar to the one that we discussed in connection with motion discrimination from area MT neurons. The stimulus is then decoded by summing the kernel $K(t - t_i)$ for all spike times t_i of the recorded H1 neuron and summing $-K(t - t_j)$ for all spike times t_j of its anti-neuron partner.

The fly has only two H1 neurons from which it must extract information about visual motion, so it seems reasonable that stimulus reconstruction using the spike-train decoding technique can produce quite accurate results (figure 3.14). It is perhaps more surprising that accurate decoding, at least in the sense of percent correct discriminations, can be obtained from single neurons out of the large population of MT neurons responding to visual motion in the monkey. Of course, the reconstruction of a time-dependent stimulus from H1 responses is more challenging than the binary discrimination done with MT neurons. Furthermore, it is worth remembering that in all the examples we have considered, including decoding wind direction from the cercal system and arm movement direction
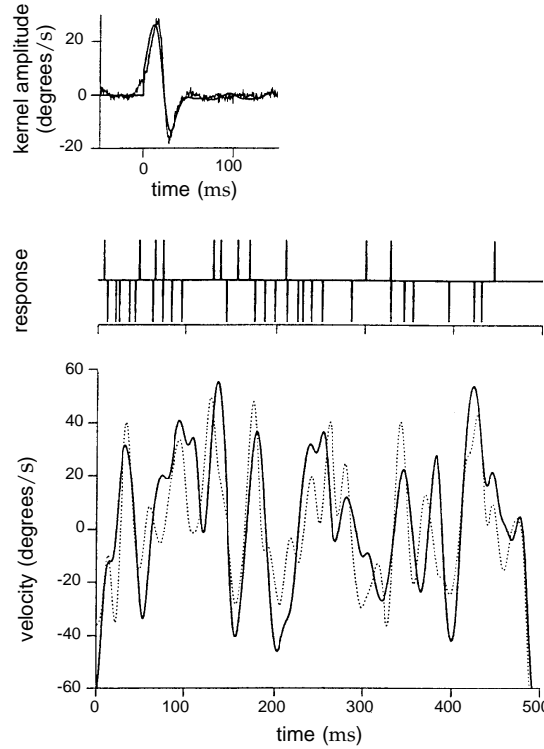
Figure 3.14 Decoding the stimulus from an H1 neuron of the fly. The upper panel is the decoding kernel. The jagged curve is the optimal acausal filter, and the smooth curve is a kernel obtained by expanding in a causal set of basis functions. In both cases, the kernels are shifted by $\tau_0 = 40$ ms. The middle panel shows typical responses of the H1 neuron to the stimuli $s(t)$ (upper trace) and $-s(t)$ (bottom trace). The dashed line in the lower panel shows the actual stimulus, and the solid line is the estimated stimulus from the optimal linear reconstruction using the acausal filter. (Adapted from Rieke et al., 1997.)

from a population of M1 neurons, the stimuli used are extremely simple compared with the naturally occurring stimuli that must be interpreted during normal behavior.

3.5 Chapter Summary

We have considered the decoding of stimulus characteristics from the responses they evoke, including discrimination between stimulus values, the decoding of static stimuli on the basis of population responses, and the decoding of dynamic stimulus parameters from spike trains. Discrimination was studied using the receiver operating characteristic, likelihood ra-

tio tests, and the Neyman-Pearson lemma. For static parameter decoding we introduced the vector method; Bayesian, maximum a posteriori, and maximum likelihood inference; the Fisher information; and the Cramér-Rao lower bound. We also showed how to use ideas from Wiener filtering to reconstruct an approximation of a time-varying stimulus from the spike trains it evokes.

3.6 Appendices

A: The Neyman-Pearson Lemma

Consider the difference $\Delta\beta$ in the power of two tests that have identical sizes α. One uses the likelihood ratio $l(r)$, and the other uses a different test function $h(r)$. For the test $h(r)$ using the threshold z_h,

$$\alpha_h(z_h) = \int dr \, p[r|-]\Theta(h(r) - z_h) \text{ and } \beta_h(z_h) = \int dr \, p[r|+]\Theta(h(r) - z_h) \,.$$
(3.61)

Similar equations hold for the $\alpha_l(z_l)$ and $\beta_l(z_l)$ values for the test $l(r)$ using the threshold z_l. We use the Θ function, which is 1 for positive and 0 for negative values of its argument, to impose the condition that the test is greater than the threshold. Comparing the β values for the two tests, we find

$$\Delta\beta = \beta_l(z_l) - \beta_h(z_h) = \int dr \, p[r|+]\Theta(l(r) - z_l) - \int dr \, p[r|+]\Theta(h(r) - z_h) \,.$$
(3.62)

The range of integration where $l(r) \geq z_l$ and also $h(r) \geq z_h$ cancels between these two integrals, so, in a more compact notation, we can write

$$\Delta\beta = \int dr \, p[r|+] \left(\Theta(l(r) - z_l)\Theta(z_h - h(r)) - \Theta(z_l - l(r))\Theta(h(r) - z_h)\right) \,.$$
(3.63)

Using the definition $l(r) = p[r|+]/p[r|-]$, we can replace $p[r|+]$ with $l(r)p[r|-]$ in this equation, giving

$$\Delta\beta = \int dr \, l(r)p[r|-]\left(\Theta(l(r) - z_l)\Theta(z_h - h(r)) - \Theta(z_l - l(r))\Theta(h(r) - z_h)\right) \,.$$
(3.64)

Then, due to the conditions imposed on $l(r)$ by the Θ functions within the integrals, replacing $l(r)$ by z can neither decrease the value of the integral resulting from the first term in the large parentheses, nor increase the value arising from the second. This leads to the inequality

$$\Delta\beta \geq z \int dr \, p[r|-] \left(\Theta(l(r) - z_l)\Theta(z_h - h(r)) - \Theta(z_l - l(r))\Theta(h(r) - z_h)\right) \,.$$
(3.65)

Putting back the region of integration that cancels between these two terms (for which $l(r) \geq z_l$ and $h(r) \geq z_h$), we find

$$\Delta \beta \geq z \left[\int dr \, p[r|-]\Theta(l(r)-z_l) - \int dr \, p[r|-]\Theta(h(r)-z_h) \right]. \quad (3.66)$$

By definition, these integrals are the sizes of the two tests, which are equal by hypothesis. Thus $\Delta \beta \geq 0$, showing that no test can be better than the likelihood ratio $l(r)$, at least in the sense of maximizing the power for a given size.

B: The Cramér-Rao Bound

Cauchy-Schwarz inequality

The Cramér-Rao lower bound for an estimator s_{est} is based on the Cauchy-Schwarz inequality, which states that for any two quantities A and B,

$$\langle A^2 \rangle \langle B^2 \rangle \geq \langle AB \rangle^2. \quad (3.67)$$

To prove this inequality, note that

$$\left\langle \left(\langle B^2 \rangle A - \langle AB \rangle B \right)^2 \right\rangle \geq 0 \quad (3.68)$$

because it is the average value of a square. Computing the square gives

$$\langle B^2 \rangle^2 \langle A^2 \rangle - \langle AB \rangle^2 \langle B^2 \rangle \geq 0, \quad (3.69)$$

from which the inequality follows directly.

Consider the inequality of equation 3.67 with $A = \partial \ln p / \partial s$ and $B = s_{est} - \langle s_{est} \rangle$. From equations 3.43 and 3.39, we have $\langle A^2 \rangle = I_F$ and $\langle B^2 \rangle = \sigma_{est}^2$. The Cauchy-Schwarz inequality then gives

$$\sigma_{est}^2(s) I_F \geq \left\langle \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} (s_{est} - \langle s_{est} \rangle) \right\rangle^2. \quad (3.70)$$

To evaluate the expression on the right side of the inequality 3.70, we differentiate the defining equation for the bias (equation 3.38),

$$s + b_{est}(s) = \langle s_{est} \rangle = \int d\mathbf{r} \, p[\mathbf{r}|s] s_{est}, \quad (3.71)$$

with respect to s to obtain

$$\begin{aligned} 1 + b'_{est}(s) &= \int d\mathbf{r} \, \frac{\partial p[\mathbf{r}|s]}{\partial s} s_{est} \\ &= \int d\mathbf{r} \, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} s_{est} \\ &= \int d\mathbf{r} \, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} (s_{est} - \langle s_{est} \rangle). \end{aligned} \quad (3.72)$$

The last equality follows from the identity

$$\int d\mathbf{r}\, p[\mathbf{r}|s] \frac{\partial \ln p[\mathbf{r}|s]}{\partial s} \langle s_{\text{est}} \rangle = \langle s_{\text{est}} \rangle \int d\mathbf{r}\, \frac{\partial p[\mathbf{r}|s]}{\partial s} = 0 \qquad (3.73)$$

because $\int d\mathbf{r}\, p[\mathbf{r}|s] = 1$. The last line of equation 3.72 is just another way of writing the expression being squared on the right side of the inequality 3.70, so combining this result with the inequality gives

$$\sigma_{\text{est}}^2(s) I_{\text{F}} \geq (1 + b_{\text{est}}'(s))^2 \,, \qquad (3.74)$$

which, when rearranged, is the Cramér-Rao bound of equation 3.41.

C: The Optimal Spike-Decoding Filter

The optimal linear kernel for spike-train decoding is determined by solving equation 3.54. This is done by taking the Fourier transform of both sides of the equation, that is, multiplying both sides by $\exp(i\omega\tau)$ and integrating over τ,

$$\int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) \int_{-\infty}^{\infty} d\tau' \, Q_{\rho\rho}(\tau - \tau') K(\tau') = \int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) Q_{rs}(\tau - \tau_0) \,. \tag{3.75}$$

By making the replacement of integration variable $\tau \to \tau + \tau_0$, we find that the right side of this equation is

$$\exp(i\omega\tau_0) \int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) Q_{rs}(\tau) = \exp(i\omega\tau_0) \tilde{Q}_{rs}(\omega) \,, \qquad (3.76)$$

where $\tilde{Q}_{rs}(\omega)$ is the Fourier transform of $Q_{rs}(\tau)$. The integral of the product of two functions that appears on the left side of equations 3.54 and 3.75 is a convolution. As a result of the theorem on the Fourier transforms of convolutions (see the Mathematical Appendix),

$$\int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) \int_{-\infty}^{\infty} d\tau' \, Q_{\rho\rho}(\tau - \tau') K(\tau') = \tilde{Q}_{\rho\rho}(\omega) \tilde{K}(\omega) \,, \qquad (3.77)$$

where $\tilde{Q}_{\rho\rho}(\omega)$ and $\tilde{K}(\omega)$ are the Fourier transforms of $Q_{\rho\rho}(\tau)$ and $K(\tau)$ respectively:

$$\tilde{Q}_{\rho\rho}(\omega) = \int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) Q_{\rho\rho}(\tau) \quad \text{and} \quad \tilde{K}(\omega) = \int_{-\infty}^{\infty} d\tau \, \exp(i\omega\tau) K(\tau) \,. \tag{3.78}$$

Putting the left and right sides of equation 3.75 together as we have evaluated them, we find that

$$\tilde{Q}_{\rho\rho}(\omega) \tilde{K}(\omega) = \exp(i\omega\tau_0) \tilde{Q}_{rs}(\omega) \,. \qquad (3.79)$$

Equation 3.60 follows directly from this result, and equation 3.59 then determines $K(\tau)$ as the inverse Fourier transform of $\tilde{K}(\omega)$.

3.7   Annotated Bibliography

Statistical analysis of discrimination, various forms of decoding, the Neyman-Pearson lemma, the Fisher information, and the Cramér-Rao lower bound can be found in **Cox & Hinckley (1974)**. Receiver operator characteristics and signal detection theory are described comprehensively in **Green & Swets (1966)** and Graham (1989). Our account of spike-train decoding follows that of **Rieke et al. (1997)**. Spectral factorization is discussed in Poor (1994). **Newsome et al. (1989)** and **Salzman et al. (1992)** present important results concerning visual motion discrimination and recordings from area MT, and **Shadlen et al. (1996)** provides a theoretically oriented review.

The vector method of population decoding has been considered in the context of a number of systems, and references include Humphrey et al. (1970), Georgopoulos et al. (1986 & 1988), van Gisbergen et al. (1987), and Lee et al. (1988). Various theoretical aspects of population decoding, such as vector and ML decoding and the Fisher information, that comprise our account were developed by Paradiso (1988), Baldi and Heiligenberg (1988), Vogels (1990), Snippe & Koenderink (1992), Zohary (1992), Seung & Sompolinsky (1993), Touretzky et al. (1993), Salinas & Abbott (1994), Sanger (1994, 1996), Snippe (1996), and Oram et al. (1998). Population codes are also known as coarse codes in the connectionist literature (Hinton, 1981). In our discussion of the effect of tuning curve widths on the Fisher information, we followed Zhang and Sejnowski (1999), but see also Snippe & Koenderink (1992) and Hinton (1984).