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

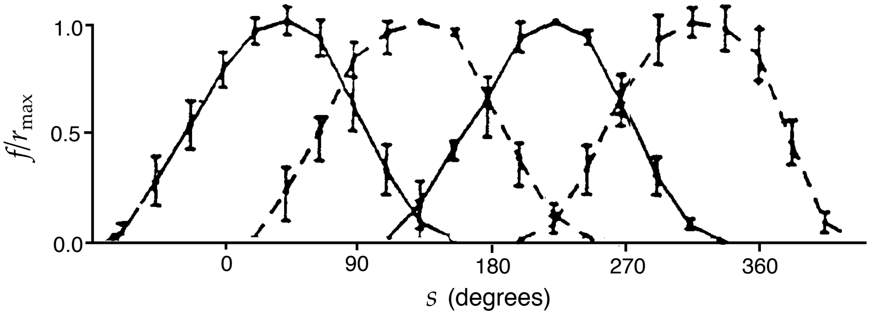

Figure 3.4 Tuning curves for the four low-velocity interneurons of the cricket cercal system plotted as a function of the wind direction s. Each neuron responds with a firing rate that is closely approximated by a half-wave rectified cosine function. The preferred directions of the neurons are located 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 its maximum, can be written as

cosine tuning

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

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

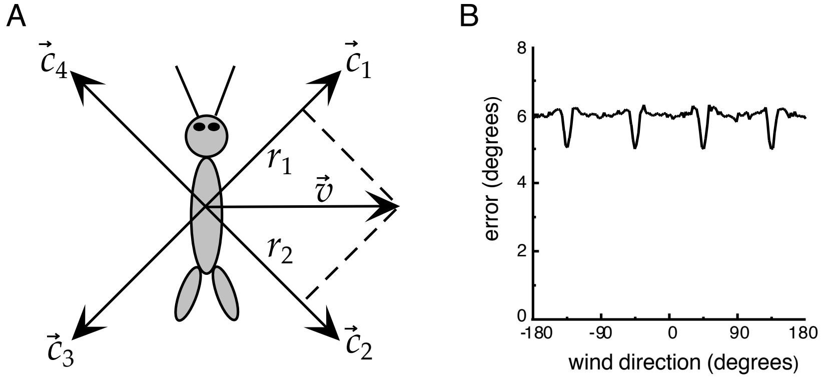

Figure 3.5 (A) Preferred directions of four cercal interneurons in relation to the cricket's body. The firing rate of each neuron for a fixed wind speed is proportional to the projection of the wind velocity vector $\vec{v}$ onto the preferred-direction axis of the neuron. The projection directions $\vec{c}_1$, $\vec{c}_2$, $\vec{c}_3$, 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

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

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 velocity 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

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

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

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

We leave as an exercise the proof that $\langle \vec{v}_{\text{pop}} \rangle$ 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

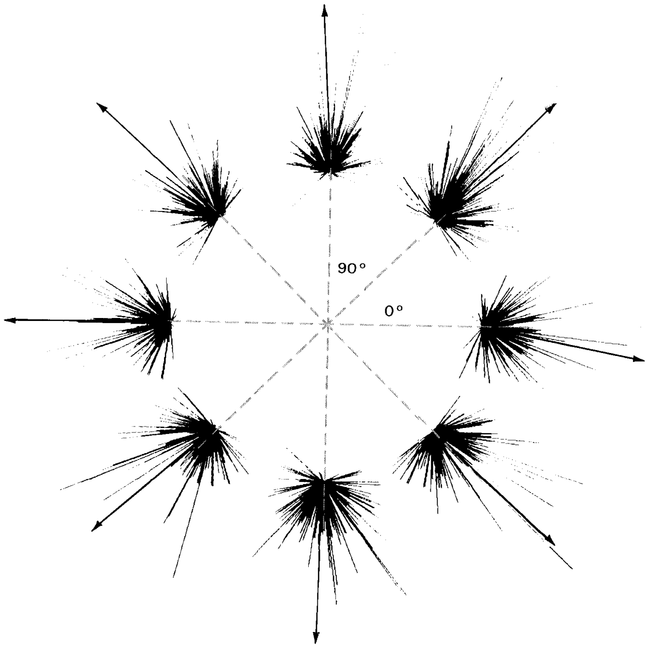

Figure 3.6 Comparison of population vectors with actual arm movement directions. Results are shown for eight different movement directions. Actual arm movement directions are radially outward at angles that are multiples of 45°. The groups of lines without arrows show the preferred-direction vectors 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., 2000, 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),

$$p[s|\mathbf{r}] = \frac{p[\mathbf{r}|s]p[s]}{p[\mathbf{r}]} . \tag{3.26}$$

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 difference between the estimate and the true value, $L(s, s_{\text{bayes}}) = (s - s_{\text{bayes}})^2$,

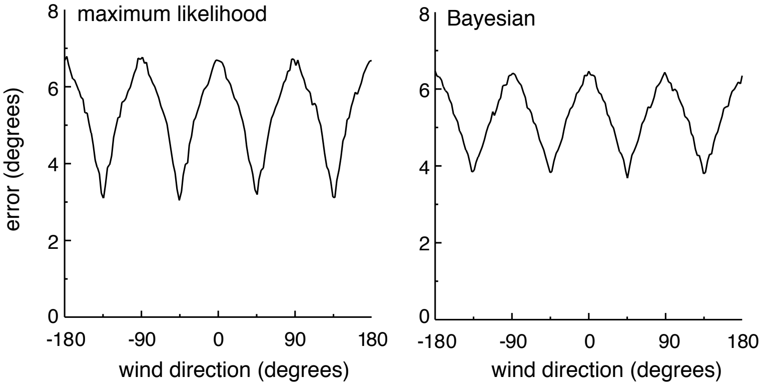

Figure 3.7 Maximum likelihood and Bayesian estimation errors for the cricket cercal system. ML and Bayesian estimates of the wind direction were compared with the actual stimulus value for a large number of simulated firing rates. Firing rates were generated as for figure 3.5B. The error shown is the root-mean-squared difference between the estimated and actual stimulus angles. (Adapted from Salinas and Abbott, 1994)

the estimate that minimizes the expected loss is the mean

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

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

MAP inference

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.

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 $n_a = r_aT$ spikes, when the average firing rate is $\langle r_a \rangle = f_a(s)$, is given by (see chapter 1)

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

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

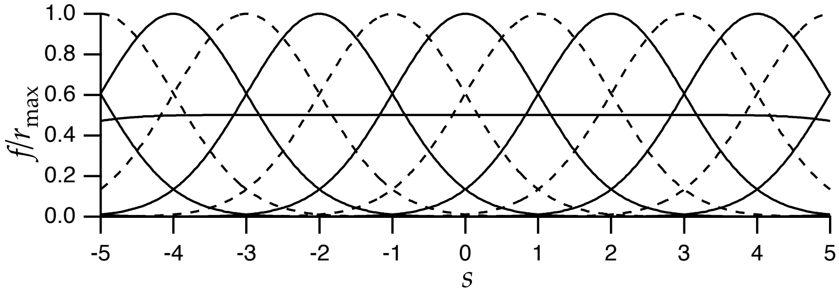

Figure 3.8 An array of Gaussian tuning curves spanning stimulus values from -5 to 5. The peak values of the tuning curves fall on the integer values of s and the tuning curves all have $\sigma_a = 1$. For clarity, the curves are drawn alternately with dashed and solid lines. The approximately flat curve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

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

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

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

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-

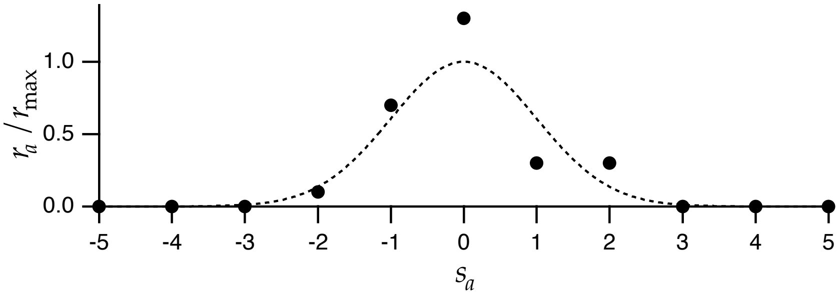

Figure 3.9 Simulated responses of 11 neurons with the Gaussian tuning curves shown in figure 3.8 to a stimulus value of 0. Firing rates for a single trial, generated using the Poisson model, are plotted as a function of the preferred-stimulus values of the different neurons in the population (filled 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,

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

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*

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

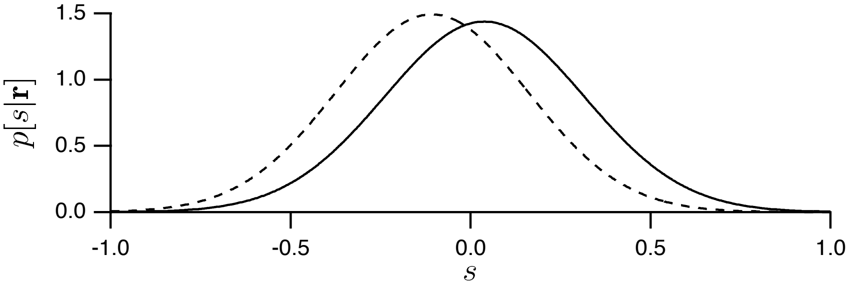

Figure 3.10 Probability densities for the stimulus, given the firing rates shown in figure 3.9 and assuming the tuning curves of figure 3.8. The solid curve is $p[s|\mathbf{r}]$ when the prior distribution of stimulus values is constant and the true value of the stimulus is $s = 0$. The dashed curve 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

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

estimation error

The bias and variance can be used to compute the trial-average squared estimation error, $\left\langle (s_{\text{est}} - s)^2 \right\rangle$. 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 can write the squared estimation error as

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

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,

$$I_{\mathrm{F}}(s) = \left\langle -\frac{\partial^2 \ln P[\mathbf{r}|s]}{\partial s^2} \right\rangle = T \sum_{a=1}^{N} \left(\langle r_a \rangle \left(\left(\frac{f_a'(s)}{f_a(s)} \right)^2 - \frac{f_a''(s)}{f_a(s)} \right) + f_a''(s) \right) . \tag{3.44}$$

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

$$I_{\mathrm{F}}(s) = T \sum_{a=1}^{N} \frac{\left(f_a'(s)\right)^2}{f_a(s)} . \tag{3.45}$$

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-

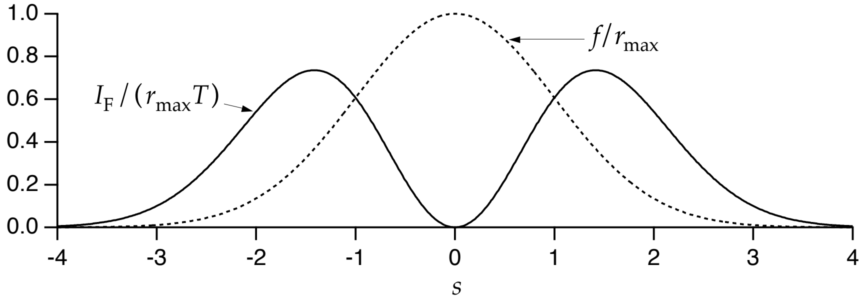

Figure 3.11 The Fisher information for a single neuron with a Gaussian tuning curve with $s = 0$ and $\sigma_a = 1$, and Poisson variability. The Fisher information (solid curve) has been divided by $r_{\max}T$, the peak firing rate of the tuning curve times the duration of the trial. The dashed curve 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

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

sums→integrals

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

$$\begin{aligned} I_F(s) &\approx \rho_s T \int_{-\infty}^{\infty} d\xi \, \frac{r_{\max}(s-\xi)^2}{\sigma_r^4} \exp\left(-\frac{1}{2}\left(\frac{s-\xi}{\sigma_r}\right)^2\right) \\ &= \frac{\sqrt{2\pi}\rho_s\sigma_r r_{\max}T}{\sigma_r^2} \, . \end{aligned} \qquad (3.47)$$

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

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

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 $1/I_{\text{F}}(s)$. Thus, the discriminability, defined in equation 3.4, for the ML-based test is

ML discriminability

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

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,

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

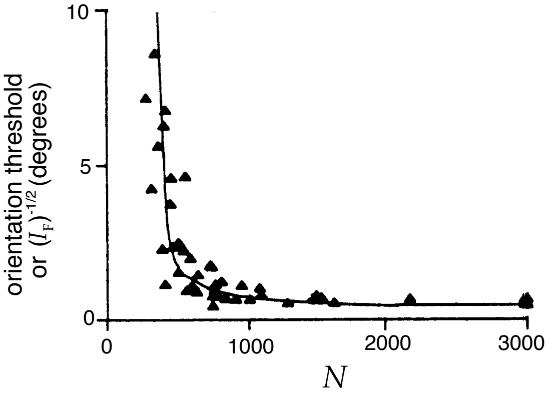

Figure 3.12 Comparison of Fisher information and discrimination thresholds for orientation tuning. The solid curve is the minimum standard deviation of an estimate of orientation angle from the Cramér-Rao bound, plotted as a function of the number of neurons (N) involved in the estimation. The triangles 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