3.3 Population Decoding
Population Decoding

So far: single neuron/test, two (or more) discrete choices
Now: population of neurons/tests, estimate continuous variable:

$$p(r \mid s) \rightarrow p(r \mid s)$$

Cricket cercal system example: estimate wind direction based on activity in four interneurons

Adapted from Theunissen and Miller 1992
• cosine tuning of individual cells (with half-wave rectification), $s$ is wind direction:

\[
\left( \frac{f(s)}{r_{\text{max}}} \right)_a = \begin{bmatrix} \cos(s - s_a) \end{bmatrix}
\]

• we can also use vectors (of unit length):

\[
\left( \frac{f(s)}{r_{\text{max}}} \right)_a = [\vec{v} \cdot \vec{c}_a]
\]

• simple population vector decoding scheme:

\[
\vec{v}_{\text{pop}} = \sum_{a=1}^{4} \left( \frac{r}{r_{\text{max}}} \right)_a \vec{c}_a
\]
Population vector method for cricket example: population vector method allows to estimate wind direction fairly accurately on the basis of the activity of just four interneurons.

Salinas and Abbott, 1994
• Activity in cortical area M1 in monkeys during reaching movements:

\[
\left( \frac{f(s) - r_0}{r_{\text{max}}} \right)_a = \vec{v} \cdot \vec{c}_a
\]
population vector decoding:

\[ \vec{v}_{\text{pop}} = \sum_{a=1}^{N} \left( \frac{r - r_0}{r_{\text{max}}} \right)_a \vec{c}_a \]
Optimal Decoding Methods

Population vector: just an ad hoc method, but can sometimes work well

Optimal methods: **Bayesian**, maximum a posteriori (MAP), maximum likelihood (ML)

MAP estimation: start with Bayes rule

\[
 p[s|r] = \frac{p[r|s]p[s]}{p[r]}
\]

This allows to calculate probability of each possible stimulus \( s \) given the neural response \( r \). But it requires knowledge of \( p(r|s) \), which is (still) difficult to estimate. (In contrast, population vector method only requires knowledge of optimal stimulus for each neuron.)

Bayesian decoding:

\[
s_{\text{bayes}} = \int ds \ p[s|r]s
\]

The **MAP** estimate of \( s \) is the value \( s^* \) which maximizes \( p(s|r) \). If \( p(s) \) does not depend on \( s \) then \( s^* \) maximizes \( p(r|s) \). This is called the maximum likelihood (ML) estimate.
Optimal decoding for cricket example:

Salinas and Abbott, 1994
Problem: Suppose you want to diagnose a patient who may have a serious illness. Consider you have made a number of more or less reliable tests. How should you diagnose? (This is a classic pattern classification problem.)

Denote the presence or absence of the illness by a binary random variable $S$. Denote the results of the $n$ tests by binary random variables $T_i$, $i=1,...,n$.

We can fully describe the system by introducing the unknown joint distribution $p(S, T_1,...,T_n)$. What we need to do, however, is to decide on the most probable value of $S$ given the test results $T_1,...,T_n$, i.e. we are looking for:

$$p(S|T_1,...,T_n)$$
Apply Bayes rule:

\[
p(S|T_1,\ldots,T_n) = \frac{p(T_1,\ldots,T_n|S)p(S)}{p(T_1,\ldots,T_n)}
\]

Natural choice: pick the value of \( S \) that has the highest posterior probability:

\[
\hat{S} = \arg \max_S p(S|T_1,\ldots,T_n)
\]

\[
= \arg \max_S p(T_1,\ldots,T_n|S)p(S)
\]

I.e. the maximum a posteriori (MAP) estimate of \( S \).
Estimating the probabilities:

\[ \hat{S} = \arg \max_S p(T_1, \ldots, T_n | S) p(S) \]

Estimating \( p(S) \) presents no challenge. We just need to count what fraction of the population has the disease. But let’s look at:

\[ p(T_1, \ldots, T_n | S) \]

Since each \( T_i \) is a binary variable, for each value of \( S \) we need to specify the probabilities of \( 2^n \) possible combinations of the \( T_n \).

For large \( n \) there will be more parameters than we’ll ever see patients! We cannot accurately estimate these probabilities from realistic amounts of training data.

This is an example of the “curse of dimensionality”.

[46x536]Estimating the probabilities:

[46x409]Estimating \( p(S) \) presents no challenge. We just need to count what fraction of the population has the disease. But let’s look at:

[46x254]Since each \( T_i \) is a binary variable, for each value of \( S \) we need to specify the probabilities of \( 2^n \) possible combinations of the \( T_n \).

[46x156]For large \( n \) there will be more parameters than we’ll ever see patients! We cannot accurately estimate these probabilities from realistic amounts of training data.

[46x39]This is an example of the “curse of dimensionality”.

(46x536)
Conditional independence to the rescue:

If the test results are all conditionally independent given S, then our problem is greatly simplified. We may write:

\[
p(T_1, \ldots, T_n|S) = p(T_1|S) \times \cdots \times p(T_n|S) \\
= \prod_{i=1}^{n} p(T_i|S)
\]

Note that each \(p(T_i|S)\) is easy to estimate. It’s just the fraction of times test \(T_i\) is positive given the disease. We only have to measure \(n\) parameters, which is tractable.

The idea of exploiting conditional dependencies is extended to more complex probabilistic models in the field of *graphical models* or *Bayes nets*.
Notes:

Often it is not clear if the conditional independence assumption holds or not. But since it so greatly simplifies the problem, it is frequently made anyway - hence the name *naïve Bayes*. Often the results appear to be very good.

Note that we restricted ourselves to binary random variables in this example, but the conditional independence assumption is useful in the case of continuous R.V.s as well.
Decoding arbitrary stimulus parameter by a population of

**Instructive example:**
Array of N neurons, preferred stimulus value uniformly distributed with Gaussian tuning curves:

\[ f_a(s) = r_{\text{max}} \exp \left( -\frac{1}{2} \left( \frac{s - s_a}{\sigma_a} \right)^2 \right) \]

\[ \sum_a f_a(s) \approx \text{const.} \]
Recall from Chapter 1:

A: Recording from neuron in primary visual cortex (V1, area 17, striate cortex) in monkey when presented with moving bars of light falling over the neuron’s receptive field.

B: **Gaussian tuning curve** fitted to the responses.
Poisson neurons:

Average rate to stimulus $s$ determined by tuning curve $\langle r_a \rangle = f_a(s)$

Probability of stimulus evoking $n_a = r_a T$ spikes

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

Assume independent neurons (as in naïve Bayes):

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

To apply ML, use logarithm and only consider terms depending on stimulus $s$ (see black board 3.2):

$$\ln P[r|s] = T \sum_{a=1}^{N} r_a \ln (f_a(s)) + \ldots$$
ML estimate

Find maximum of r.h.s. by setting derivative to zero:  \[
\sum_{a=1}^{N} r_a \frac{f'_a(s_{\text{ML}})}{f_a(s_{\text{ML}})} = 0
\]

For Gaussians we can use:  \[
f'_a(s)/f_a(s) = (s_a - s)/\sigma_a^2
\]

and thus obtain:  \[
s_{\text{ML}} = \frac{\sum r_a s_a/\sigma_a^2}{\sum r_a/\sigma_a^2}
\]

and if all tuning curves have the same width:

\[
s_{\text{ML}} = \frac{\sum r_a s_a}{\sum r_a}
\]

Intuitive explanation: firing rate weighted average of preferred values of encoding neurons
MAP estimate: include prior information

Prior information is taken into account
\[ \ln p[s|r] = \ln P[r|s] + \ln p[s] - \ln P[r] \]

We now have
\[ \ln p[s|r] = T \sum_{a=1}^{N} r_a \ln \left( f_a(s) \right) + \ln p[s] + \ldots \]

Derivative to zero:
\[ T \sum_{a=1}^{N} r_a f_a'(s_{\text{MAP}}) \frac{p'[s_{\text{MAP}}]}{p[s_{\text{MAP}}]} = 0 \]

which leads to:
\[ s_{\text{MAP}} = \frac{T \sum r_a s_a / \sigma_a^2 + s_{\text{prior}} / \sigma_{\text{prior}}^2}{T \sum r_a / \sigma_a^2 + 1 / \sigma_{\text{prior}}^2} \]

Solid: for constant stimulus distribution
Dashed: for Gaussian, \( s_{\text{prior}} = -2, \sigma_{\text{prior}} = 1 \)
Cricket cercal system
Bias and Variance of Estimate

Bias is difference between true stimulus and average estimate of the stimulus. **Unbiased estimator**: bias is zero for all stimuli:

\[ b_{\text{est}}(s) = \langle s_{\text{est}} \rangle - s \]

Variance quantifies how much the estimate varies about the mean estimate:

\[ \sigma^2_{\text{est}}(s) = \langle (s_{\text{est}} - \langle s_{\text{est}} \rangle)^2 \rangle \]

Average squared estimation error (see black board 3.3):

\[ \langle (s_{\text{est}} - s)^2 \rangle = \sigma^2_{\text{est}}(s) + b^2_{\text{est}}(s) \]
Fisher Information

Significance:

Limits accuracy with which any decoding scheme can extract an estimate of the encoded quantity.

Provides measure of possible encoding accuracy.

No encoding scheme’s accuracy can exceed the accuracy of the optimal decoding scheme.
Definition Fisher information:

\[ I_F(s) = \left\langle -\frac{\partial^2 \ln p[r|s]}{\partial s^2} \right\rangle = \int dr \ p[r|s] \left( -\frac{\partial^2 \ln p[r|s]}{\partial s^2} \right) \]

Equivalent:

\[ I_F(s) = \left\langle \left( \frac{\partial \ln p[r|s]}{\partial s} \right)^2 \right\rangle = \int dr \ p[r|s] \left( \frac{\partial \ln p[r|s]}{\partial s} \right)^2 \]

Depends on stimulus s, but dependence is purely local

Cramer-Rao bound limits variance \( \sigma^2_{est} \) of any estimate \( s_{est} \):

\[ \sigma^2_{est}(s) \geq \frac{(1 + b'_{est}(s))^2}{I_F(s)} \]

proved in appendix

with bias

\[ b_{est}(s) = \langle s_{est} \rangle - s \]

Without bias:

\[ \sigma^2_{est} \geq 1/I_F(s) \]
Poisson neurons

\[ I_F(s) = T \sum_{a=1}^{N} \frac{(f'_a(s))^2}{f_a(s)} \]

Each neuron contributes an amount. High slopes contribute most. Peaks contribute little.

Interpretation: At high slopes the firing rates are sensitive to precise value of stimulus.
3.4 Spike-Train Decoding
Spike train decoding

**Goal:**
Decoding temporally varying stimuli

**Problem:**
Cannot be properly decoded from spike counts, since temporal resolution may be insufficient

**Solution:**
Use individual spikes to reconstruct stimulus.
H1 neuron in fly visual system

assume from now on that time average of stimulus is zero

adapted from Rieke et al. 1997
Issues:
• At time $t$, decode stimulus
• Can only use spikes at time points $t_i < t$ !
• But spike at $t_i$ can only be caused by stimulus $s$ up to time $t_i$
• Estimating $s$ at times $> t_i$ possible if stimulus correlated in time
• Estimating the stimulus at time $t$ (instantaneous decoding) is hard! (not many spikes available)
• Easier: decode at time $t - \tau_0$
• The larger $\tau_0$ the easier the decoding (more spikes available), but decoding ought to be fast as well $\rightarrow$ compromise
**Approach:**
Construct stimulus estimate as linear sum over all spikes

\[
\text{\textit{s}}_{\text{est}}(t - \tau_0) = \sum_{i=1}^{n} K(t - t_i) - \langle r \rangle \int_{-\infty}^{\infty} d\tau K(\tau)
\]

\[
\text{\textit{s}}_{\text{est}}(t - \tau_0) = \int_{-\infty}^{\infty} d\tau \left( \rho(t - \tau) - \langle r \rangle \right) K(\tau)
\]

with \( \rho(t) = \sum \delta(t - t_i) \)

**Interpretation:**
- Estimate stimulus at \( t - \tau_0 \) as linear sum of Kernels K placed at spike times i
- Causality ignored (treated later); sum is over all spikes

 guarantees that average estimated stimulus is zero
\[ s_{\text{est}}(t - \tau_0) = \sum_{i=1}^{n} K(t - t_i) - \langle r \rangle \int_{-\infty}^{\infty} d\tau K(\tau) \]

What to expect from this reconstruction?

- Kernel \( K \) might be related to spike-triggered average
- Shifted by \( \tau_0 \)
- Reverse in \( \tau \)
Estimation procedure: Minimize mean squared error (over time and trials)

\[
\frac{1}{T} \int_0^T dt \left\langle \left( \int_{-\infty}^\infty d\tau \left( \rho(t - \tau) - \langle r \rangle \right) K(\tau) - s(t - \tau_0) \right)^2 \right\rangle
\]

Set functional derivative to zero! Result:

\[
\int_{-\infty}^\infty d\tau' Q_{\rho\rho}(\tau - \tau') K(\tau') = Q_{rs}(\tau - \tau_0)
\]

with spike train autocorrelation

\[
Q_{\rho\rho}(\tau - \tau') = \frac{1}{T} \int_0^T dt \left\langle (\rho(t - \tau) - \langle r \rangle)(\rho(t - \tau') - \langle r \rangle) \right\rangle
\]

and the correlation of firing rate and stimulus

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

which is related to the spike triggered average: 

\[
C(\tau) = \left\langle \frac{1}{n} \sum_{i=1}^n s(t_i - \tau) \right\rangle
\]
Computing $K$ (approximation):

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

Similar equation as with receptive field estimation. There we used white noise stimulus to simplify. In general we cannot assume

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

But if we do so anyway, we obtain

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

which is essentially the reflected, shifted spike triggered average

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

**Interpretation:**
Decoding: every time a spike appears, we replace it with the average stimulus surrounding a spike, shifted by an amount \( \tau_0 \).
Computing $K$ (exactly):

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

Solution given by inverse Fourier transform

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

where

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

The numerator reproduces the previous expression $Q_{rs}(\tau - \tau_0)$

The denominator corrects for biases due to autocorrelations in the response spike train
Computing $K$ (exactly):

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

Take Fourier transform of both sides:

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

With $\tau \rightarrow \tau + \tau_0$

the r.h.s. is: $\exp(i\omega \tau_0) \int_{-\infty}^{\infty} d\tau \exp(i\omega \tau)Q_{rs}(\tau) = \exp(i\omega \tau_0)\tilde{Q}_{rs}(\omega)$

The l.h.s is $\tilde{Q}_{\rho \rho}(\omega)\tilde{K}(\omega)$

Thus:

$$\tilde{Q}_{\rho \rho}(\omega)\tilde{K}(\omega) = \exp(i\omega \tau_0)\tilde{Q}_{rs}(\omega)$$
Impose causality:
Simple (rough) method: Replace $K(\tau)$ with $K(\tau)\Theta(\tau)$
Impose causality:
More elaborate: expand kernel in causal set of basis functions

Adapted from Rieke et al. 1997
3.5 Chapter Summary
Chapter 3, concepts to remember:

- Bayes’ Theorem
- Discrimination
- Signal Detection Theory
- Receiver Operating Characteristic
- discriminability $d'$
- Bayesian decoding, MAP, ML
- Naive Bayes
- Fisher Information
- Cramér-Rao bound
- Spike-train decoding