Part 2.2

Population Coding

How neural activities relate to an animal’s behavior
Reminder: matrix inversion

A vector: \[ \vec{a} = \left( \begin{array}{c} a_1 \\ a_2 \end{array} \right) \]

A matrix: \[ B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \]
**Reminder: matrix inversion**

**A vector:**

\[ \vec{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \]

**A matrix:**

\[ B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \]

**Matrix calculus:**

\[ \vec{c} = B\vec{a} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \]

\[ c_1 = b_{11}a_1 + b_{12}a_2 \]

\[ c_2 = b_{21}a_1 + b_{22}a_2 \]
Reminder: matrix inversion

\[ B^{-1} B = BB^{-1} = I \]

\[ I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

\[ \vec{c} = B\vec{a} \quad B^{-1}\vec{c} = \vec{a} \]

\[ c_1 = b_{11}a_1 + b_{12}a_2 \]
\[ c_2 = b_{21}a_1 + b_{22}a_2 \]

Solve...

\[ a_1 = \frac{b_{21}}{b_{21}b_{12} - b_{22}b_{11}} c_1 + \frac{b_{11}}{b_{21}b_{12} - b_{22}b_{11}} c_2 \]

\[ B_{11}^{-1} \quad B_{12}^{-1} \]
Reminder: inner product

\[ a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \]

length
\[ ||a|| = \sqrt{a_1^2 + a_2^2} \]
\[ ||b|| = \sqrt{b_1^2 + b_2^2} \]
Reminder: inner product

\[ \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \]

\[ \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \]

length

\[ \|\mathbf{a}\| = \sqrt{a_1^2 + a_2^2} \]

\[ \|\mathbf{b}\| = \sqrt{b_1^2 + b_2^2} \]

inner product

\[ \mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 \]
Reminder: inner product

\[ a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \]

\[ b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \]

length

\[ \|a\| = \sqrt{a_1^2 + a_2^2} \]

\[ \|b\| = \sqrt{b_1^2 + b_2^2} \]

inner product

\[ a \cdot b = a_1 b_1 + a_2 b_2 \]

angle

\[ \cos(\phi) = \frac{a \cdot b}{\|a\| \cdot \|b\|} \]
Reminder: inner product

\[ \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \]
\[ \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \]

**length**

\[
\| \mathbf{a} \| = \sqrt{a_1^2 + a_2^2}
\]
\[
\| \mathbf{b} \| = \sqrt{b_1^2 + b_2^2}
\]

**inner product**

\[
\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2
\]

**angle**

\[
\cos(\phi) = \frac{\mathbf{a} \cdot \mathbf{b}}{\| \mathbf{a} \| \cdot \| \mathbf{b} \|}
\]

\[
\phi = 90^\circ \iff \mathbf{a} \cdot \mathbf{b} = 0
\]
Motion discrimination task
Motion discrimination task

0% coherence  50% coherence  100% coherence
Decoding with one neuron

![Graph showing neuron activity for left and right motions.](image)
Decoding with one neuron

number of events

neuron 1 (spikes/sec)

possible decision threshold

motion
left right
Measurement of responses in MT

Graph showing the relationship between coherence (%) and fraction correct. The graph includes data points for behavior (solid circles) and neuronal responses (open circles), with a best-fit line. The x-axis represents coherence (%) ranging from 0.1 to 100, and the y-axis represents the fraction correct ranging from 0.5 to 1.0.

Bar charts showing the number of trials for different coherence levels:
- Null Direction: coherence = 12.8%
- Preferred Direction: coherence = 3.2%
- coherence = 0.8%

Histograms showing the firing rate (Hz) for different coherence levels.
Measurement of responses in MT

- Graph showing the fraction correct against coherence (%).
  - Black dots: behavior.
  - White circles: neuronal.

- Histograms showing firing rate vs. number of trials for different levels of coherence:
  - Null Direction: coherence = 12.8%
  - Preferred Direction: coherence = 0.8%
Decoding with two neurons

![Graph showing the relationship between neuron 1 and neuron 2 activity, indicating motion left.](image-url)
Decoding with two neurons
Decoding with two neurons
Decoding with two neurons

Decision threshold

Motion
left
right
Decoding with two neurons

Decision threshold

motion
left    right


tangent vector
\[ b = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

normal vector
\[ a = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]

\[ a \cdot b = 0 \]
Decoding with two neurons

Decision threshold

motion
left right

\[ x = a \cdot r \]
Decoding with two neurons

\[
a = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}
\]
Decoding with two neurons

Decision threshold

motion
left right

neuron 2 (spikes/sec)

neuron 1 (spikes/sec)

\[ x = a \cdot r = a_1 r_1 + a_2 r_2 \]

\[ a = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]
Decoding with two neurons

\[ x = a \cdot r = a_1 r_1 + a_2 r_2 = \frac{1}{\sqrt{2}} \left( r_1 - r_2 \right) \]

\[ a = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]
Decoding with two neurons

Decision threshold

motion
left  right

\[ x = a \cdot r \]
\[ = a_1 r_1 + a_2 r_2 \]
\[ = \frac{1}{\sqrt{2}} \left( r_1 - r_2 \right) \]
\[ x > 0 \iff r_1 > r_2 \]
(Last week)

Decision rule: $r_1 > r_2$
The importance of correlations

Decision threshold

motion
left right

essentially error-free decoding
The importance of correlations

Decision threshold

motion
left → right

identical single-neuron histograms, but now there are decoding errors!!
Covariance and Correlation

$r_{1,i}$: firing rate of first neuron in i-th trial
$r_{2,i}$: firing rate of second neuron in i-th trial

(focus on trials with rightward motion)
Covariance and Correlation

\( r_{1,i} \): firing rate of first neuron in i-th trial
\( r_{2,i} \): firing rate of second neuron in i-th trial

average:

\[
\bar{r}_1 = \frac{1}{N} \sum_{i=1}^{N} r_{1,i} \\
\bar{r}_2 = \frac{1}{N} \sum_{i=1}^{N} r_{2,i}
\]
Covariance and Correlation

\( r_{1,i} \): firing rate of first neuron in i-th trial
\( r_{2,i} \): firing rate of second neuron in i-th trial

average:

\[
\bar{r}_1 = \frac{1}{N} \sum_{i=1}^{N} r_{1,i} \quad \quad \quad \bar{r}_2 = \frac{1}{N} \sum_{i=1}^{N} r_{2,i}
\]

variance:

\[
\text{Var}(r_1) = \frac{1}{N - 1} \sum_{i=1}^{N} (r_{1,i} - \bar{r}_1)^2 \quad \quad \quad \text{Var}(r_2) = \ldots
\]
Covariance and Correlation

(focus on trials with rightward motion)

variance:
\[ \text{Var}(r_1) = \frac{1}{N - 1} \sum_{i=1}^{N} (r_{1,i} - \bar{r}_1)^2 \quad \text{Var}(r_2) = \ldots \]

covariance:
\[ \text{Cov}(r_1, r_2) = \frac{1}{N - 1} \sum_{i=1}^{N} (r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2) \]
Covariance and Correlation

(variance: \[ \text{Var}(r_1) = \frac{1}{N - 1} \sum_{i=1}^{N} (r_{1,i} - \bar{r}_1)^2 \quad \text{Var}(r_2) = \ldots \])

covariance: \[ \text{Cov}(r_1, r_2) = \frac{1}{N - 1} \sum_{i=1}^{N} (r_{1,i} - \bar{r}_1)(r_{2,i} - \bar{r}_2) \]

Correlation coefficient: \[ r = \frac{\text{Cov}(r_1, r_2)}{\sqrt{\text{Var}(r_1) \text{Var}(r_2)}} \]
\[ \Sigma = \begin{bmatrix} \text{Var}(r_1) & \text{Cov}(r_1, r_2) \\ \text{Cov}(r_1, r_2) & \text{Var}(r_2) \end{bmatrix} \]

Covariance Matrix
Covariance and Correlation

Correlation coefficient:
\[ r = \frac{\text{Cov}(r_1, r_2)}{\sqrt{\text{Var}(r_1)\text{Var}(r_2)}} \in [-1, 1] \]

(focus on trials with rightward motion)

http://en.wikipedia.org
Covariance and Correlation

Correlation coefficient:

\[ r = \frac{\text{Cov}(r_1, r_2)}{\sqrt{\text{Var}(r_1)\text{Var}(r_2)}} \in [-1, 1] \]

(focus on trials with rightward motion)

http://en.wikipedia.org
Covariance and Correlation

Correlation coefficient: “second-order correlations”

$$r = \frac{\text{Cov}(r_1, r_2)}{\sqrt{\text{Var}(r_1)\text{Var}(r_2)}} \in [-1, 1]$$

(focus on trials with rightward motion)

“higher-order correlations”

http://en.wikipedia.org
The importance of correlations

$r = 0.79$

$r = 0.82$
The importance of correlations

\[ r = -0.91 \]

\[ r = -0.89 \]
The importance of correlations
The importance of correlations
The importance of correlations

[Diagram showing scatter plot and histogram with color coding for motion (left and right)]
The importance of correlations

How to find \( \vec{a} \)?
Linear Decoding with $N$ neurons

projection

\[ x = a \cdot r \]

decision rule:

\[ x > \theta \]

decision boundary:

hyperplane orthogonal to $a$
Linear Decoding with N neurons

projection
\[ x = a \cdot r \]
decision rule:
\[ x > \theta \]
decision boundary:
hyperplane orthogonal to \( a \)
Linear Decoding with N neurons

projection
\[ x = a \cdot r \]

decision rule:
\[ x > \theta \]

decision boundary:
hyperplane orthogonal to \( a \)

How to find \( a \) for real data?
- linear discriminant analysis
- support vector machines

It’s a “classification” problem with two classes (left/right)
Linear Discriminant Analysis (LDA)

Covariance Matrix:

\[
\sum = \begin{bmatrix}
\text{Var}(r_1) & \text{Cov}(r_1, r_2) \\
\text{Cov}(r_1, r_2) & \text{Var}(r_2)
\end{bmatrix}
\]

Mean responses:

\[
\vec{r}_{\text{right}} = \begin{pmatrix}
\bar{r}_1 \\
\bar{r}_2
\end{pmatrix}_{\text{right}} \\
\vec{r}_{\text{left}} = \begin{pmatrix}
\bar{r}_1 \\
\bar{r}_2
\end{pmatrix}_{\text{left}}
\]
Linear Discriminant Analysis (LDA)

Average neural responses when motion is right

\[ \tilde{a} = \sum^{-1} \left( \vec{r}_{\text{right}} - \vec{r}_{\text{left}} \right) \]

Inverse Covariance matrix

Average neural responses when motion is left
The cercal system of the cricket
Four interneurons
Four interneurons

Tuning curves: \( f_a(s) = r_{\text{max}} \cos(s - s_a) \)
Preferred directions

Tuning curves: \[ f_a(s) = r_{\text{max}} \cos(s - s_a) \]

vertical wind direction \( s \) (degrees)

\[ s_1 = 45^\circ \quad s_3 = 225^\circ \]
\[ s_2 = 135^\circ \quad s_4 = 315^\circ \]
Preferred directions in 2D

$s_2 = 135^\circ$

$s_1 = 45^\circ$

$s_3 = 225^\circ$

$s_4 = 315^\circ$
Preferred directions in 2D

\[ c_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \]

\[ c_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

\[ c_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \]

\[ c_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]
Wind direction

\[ c_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \]

\[ c_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

\[ c_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \]

\[ c_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]

\[ \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \quad \|\mathbf{v}\| = 1 \]
Wind direction

\[ \mathbf{v} \cdot \mathbf{c}_a = \cos(s - s_a) \]

up/down

\[ \mathbf{v} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}, \quad \|\mathbf{v}\| = 1 \]

left/right

\[ \mathbf{c}_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \]

\[ \mathbf{c}_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

\[ \mathbf{c}_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ -1 \end{pmatrix} \]

\[ \mathbf{c}_4 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \]
Four interneurons

Tuning curves: \[ f_a(s) = r_{\text{max}} \left[ \cos(s - s_a) \right]_+ \]
\[ = r_{\text{max}} \left[ \mathbf{v} \cdot \mathbf{c}_a \right]_+ \]
Four interneurons

Tuning curves: \[ f_a(s) = r_{\text{max}} \left[ \cos(s - s_a) \right]_+ \]
\[ = r_{\text{max}} \left[ \mathbf{v} \cdot \mathbf{c}_a \right]_+ \]

How can we read-out the wind direction?
The population vector

c_2

v

c_1

up/down

left/right
The population vector

\[ \text{length} = c_2 \cdot v \]

\[ \text{length} = c_1 \cdot v \]
The population vector

Reconstruction:
\[ \mathbf{v} = (c_1 \cdot \mathbf{v})c_1 + (c_2 \cdot \mathbf{v})c_2 \]

length = \( c_2 \cdot \mathbf{v} \)

length = \( c_1 \cdot \mathbf{v} \)
The population vector

Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_1 \cdot \mathbf{v}) \mathbf{c}_1 + (\mathbf{c}_2 \cdot \mathbf{v}) \mathbf{c}_2 \]

Alternative Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_2 \cdot \mathbf{v}) \mathbf{c}_2 + (\mathbf{c}_3 \cdot \mathbf{v}) \mathbf{c}_3 \]
The population vector

Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_1 \cdot \mathbf{v})\mathbf{c}_1 + (\mathbf{c}_2 \cdot \mathbf{v})\mathbf{c}_2 \]

Alternative Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_2 \cdot \mathbf{v})\mathbf{c}_2 + (\mathbf{c}_3 \cdot \mathbf{v})\mathbf{c}_3 \]

Note:
neuron’s tuning curve
\[ f(s_a) = r_{\text{max}} \left[ \mathbf{v} \cdot \mathbf{c}_a \right]_+ \]
only encodes positive numbers!
The population vector

Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_1 \cdot \mathbf{v}) \mathbf{c}_1 + (\mathbf{c}_2 \cdot \mathbf{v}) \mathbf{c}_2 \]

Alternative Reconstruction:
\[ \mathbf{v} = (\mathbf{c}_2 \cdot \mathbf{v}) \mathbf{c}_2 + (\mathbf{c}_3 \cdot \mathbf{v}) \mathbf{c}_3 \]

Note:
neuron’s tuning curve
\[ f(s_a) = r_{\text{max}} \left[ \mathbf{v} \cdot \mathbf{c}_a \right]_+ \]
only encodes positive numbers!

For any wind direction \( \mathbf{v} \):
\[ \mathbf{v} = \sum_{a=1}^{4} \left[ \mathbf{c}_a \cdot \mathbf{v} \right]_+ \mathbf{c}_a \]
Four interneurons

Tuning curves: \[ f_a(s) = r_{\text{max}} \left[ \cos(s - s_a) \right]_+ \]

\[ = r_{\text{max}} \left[ \mathbf{v} \cdot \mathbf{c}_a \right]_+ \]

How can we read-out the wind direction?

use the identity: \( \mathbf{v} = \sum_{a=1}^{4} [\mathbf{c}_a \cdot \mathbf{v}]_+ \mathbf{c}_a \) to write:

\[ \mathbf{v}_{\text{pop}} = \sum_{a=1}^{4} \frac{f_a(s)}{r_{\text{max}}} \mathbf{c}_a \]

- population vector (linear vector read-out)
- Preferred direction
- Neural response
Population vector in M1

preferred direction = 150°

Single cell responses to arm movement
Georgopoulos et al, 1982
Population vector in MI

Sum on all neurons:

\[ \mathbf{v} = \sum_{i=1}^{N} r_i \mathbf{c}_i = r_1 \mathbf{c}_1 + \ldots + r_N \mathbf{c}_N \]

Population responses and linear vector read-out
Georgopoulos et al, 1982
Population vector in MI

Sum on all neurons:

\[ \mathbf{v} = \sum_{i=1}^{N} r_i \mathbf{c}_i = r_1 \mathbf{c}_1 + \ldots + r_N \mathbf{c}_N \]

Population responses and linear vector read-out
Georgopoulos et al, 1982

Usually not optimal... Should take into account the covariance
Neural Decoding in General

Decoding: given a neural response, what was the stimulus?

\[ p(s|\mathbf{r}) \]

Sensory Stimuli

neural system

Neural Responses
Neural Encoding

sensory stimuli $S$

$p(s)$ prior distribution of stimuli

neural system

$p(r|s)$ conditional dist. of responses

neural responses $r$
Neural Encoding/Decoding

Joint probability

\[ p(r, s) = p(r|s)p(s) \]

Sensory stimuli

S

Prior distribution of stimuli

\[ p(s) \]

Neural responses

r

Conditional dist. of responses

\[ p(r|s) \]
Neural Encoding/Decoding

joint probability

\[ p(r, s) = p(r|s)p(s) \]
\[ = p(s|r)p(r) \]

sensory stimuli \( S \)

\( p(s) \) prior distribution of stimuli

neural responses \( r \)

\( p(r|s) \) conditional dist. of responses
Neural Encoding/Decoding

**Joint probability**

\[
p(r, s) = p(r | s)p(s) = p(s | r)p(r)
\]

**Bayes’ theorem**

\[
p(s | r) = \frac{p(r | s)p(s)}{p(r)}
\]
Neural Encoding/Decoding

joint probability

\[ p(r, s) = p(r|s)p(s) = p(s|r)p(r) \]

Bayes’ theorem

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

posterior

prior
Neural Encoding/Decoding

Joint probability

$$p(r, s) = p(r|s)p(s) = p(s|r)p(r)$$

Bayes’ theorem

$$p(s|r) = \frac{p(r|s)p(s)}{p(r)}$$

- **Posterior**
- **Normalization factor**
- **Prior**

$$p(r) = \int ds \, p(r|s)p(s)$$
Neural Encoding/Decoding

Joint probability

\[ p(r, s) = p(r \mid s)p(s) = p(s \mid r)p(r) \]

Bayes’ theorem

\[ p(s \mid r) = \frac{p(r \mid s)p(s)}{p(r)} \]

Posterior

Normalization factor

Prior

\[ p(r) = \int ds \ p(r \mid s)p(s) \]
Distribution over all possible stimuli... Very hard to measure in general!

Bayes’ theorem

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

- posterior
- normalization factor
- prior

\[
p(r) = \int ds \, p(r|s)p(s)
\]
Next week: Decision making
The week after that: Neural encoding

Homework 3 AND 4 due next week.

Paper version in class or get it from website.

Solutions posted next week on website.

Please bring paper version of homework answers.