Population Coding

Jonathan Pillow
PNI & Psychology

Primary Motor Cortex
(arm-reaching task)

Tuning Curve
(“Direction-tuned”)

Georgopolous 1982

Princeton NEU 501a
Dynamics & Computation module
Dec 2, 2015
readings

• Dayan & Abbott, Chap 3


stimulus  encoding  neural representation  decoding  stimulus judgment
• **Stimulus**: physical feature of the world (often 1D)
  – Orientation of a contour
  – Direction of self-motion
  – Number of students in class
  – Whether object A is bigger than object B
  – ....
• **Stimulus**: physical feature of the world (often 1D)
  – Orientation of a contour
  – Direction of self-motion
  – Number of students in class
  – Whether object A is bigger than object B
  – ....

• **Neural representation**: spike activity of neurons in response to stimulus (population code)
- **Stimulus**: physical feature of the world (often 1D)
  - Orientation of a contour
  - Direction of self-motion
  - Number of students in class
  - Whether object A is bigger than object B
  - ....

- **Neural representation**: spike activity of neurons in response to stimulus (population code)

- **Stimulus judgment**: often motor response
Primary Visual Cortex ("V1")

Hubel & Weisel, 1968

Tuning Curve ("Orientation tuned")

$s$ (orientation angle in degrees)
Population codes

Primary Motor Cortex (arm-reaching task)

Tuning Curve ("Direction-tuned")

Georgopolous 1982
neural code in motor cortex

paninski et al 2004
Tuning curve of a single neuron

Mean response as a function of the stimulus

Activity (spikes/s)

\[ f(s) \]

Stimulus \( s \)

Preferred stimulus of this neuron
Variability around the mean response

Response distribution: $p(r \mid s)$

Variability is **Poisson-like**: spike count variance proportional to mean

---

Trial 1: 7 spikes
Trial 2: 5 spikes
Trial 3: 3 spikes
Trial 4: 6 spikes
Poisson variability

- Discrete distribution (spike counts)
  \[ p(r | s) = \frac{e^{-f(s)} f(s)^r}{r!} \]

- Variance = mean = \( f(s) \).
- \( r \) is an integer, \( f(s) \) not necessarily
- Fano factor = Variance/mean = 1
  (Physiology: near Poisson, but Fano not 1)
Gaussian Variability

- Continuous distribution
  \[ p(r \mid s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(r-f(s))^2}{2\sigma^2}} \]

- Problem at small means

- Variance is not fixed \( \rightarrow \)
  \[ p(r \mid s) = \frac{1}{\sqrt{2\pi\sigma(s)^2}} e^{-\frac{(r-f(s))^2}{2\sigma(s)^2}} \]

- Very similar to Poisson for large means
Single neuron – response variability

\[ p(r \mid s) \]

Variability depends on stimulus
Population of neurons

$f_1(s), f_2(s), \ldots, f_N(s)$

$N$: number of neurons

Activity (spikes/s)

Stimulus

Different preferred stimuli
Population activity on a single trial

\[ \mathbf{r} = (r_1, r_2, \ldots, r_N) \]

Given stimulus \( S \)

Preferred stimulus

\( N \) neurons

These are now different neurons, not different stimuli!
Population activity – variability

\[ \mathbf{r} = (r_1, r_2, \ldots, r_N) \]

Response distribution (noise distribution): 
\[ p(\mathbf{r} | s) \]
Independent Poisson variability

One neuron: \[ p(r \mid s) = \frac{e^{-f(s)} f(s)^r}{r!} \]

Population: \[ p(r \mid s) = \prod_{i=1}^{N} \frac{e^{-f_i(s)} f_i(s)^{r_i}}{r_i!} \]
True or false?

• When neurons have similar tuning curves, a population pattern of activity looks like a noisy version of a tuning curve. TRUE

• The width of a bell-shaped tuning curve is related to the variance of the neuron’s responses. FALSE

• The variability of a single neuron responding to a stimulus can be exactly determined from the value of its tuning curve at that stimulus. FALSE

• The variability of population activity follows from the variability of a single neuron without any further assumptions. FALSE
Population codes in the brain

• Primary visual cortex (orientation, spatial frequency)
• MT (motion direction, velocity)
• IT (human faces, objects)
• SC (saccade direction)
• Primary motor cortex (arm movement direction)
• Hippocampus in rat (self location)
• Cercal interneurons in cricket (wind direction)
• Prefrontal cortex ( numerosity)

Why population coding and not single-neuron coding? ...
Decoding population activity

\[ \mathbf{r} = (r_1, r_2, \ldots, r_N) \]

some operation

decoder estimator read-out

\[ \hat{S} \]
Winner-take-all decoder

\[ r = (r_1, r_2, \ldots, r_N) \]

\[ \hat{s} = s_j : j = \arg\max_i r_i \]
Center-of-mass decoder

\[ \mathbf{r} = (r_1, r_2, \ldots, r_N) \]

Activity

\[ S_1, S_2, \ldots, S_N \]

weight factors

preferred stimuli of neurons

\[ \hat{S} = \frac{\sum_i r_i S_i}{\sum_i r_i} \]
Template-matching decoder

The template has the shape of a tuning curve.

$$s = \arg\min_s \sum_{i=1}^{N} (r_i - f_i(s))^2$$
Maximum-likelihood decoder

• Does not only use $r$ and $f(s)$, but also noise distribution $p(r|s)$ (e.g. independent Poisson)

• Find the value of $s$ that maximizes the likelihood of $s$, i.e. $p(r|s)$

\[
\hat{s} = \text{argmax}_{s} p(r|s)
\]

• Experimental disadvantage: need to know (or assume) $p(r|s)$
Population Coding: summary so far

- tuning curves (function of “stimulus”)
- population response (function of “preferred stimulus”)
- noise distributions

- population decoders (winner-take-all, center-of-mass, template-matching, maximum-likelihood, Bayesian)
Bayesian Estimation
Formal treatment: Bayesian Estimation

\[
\begin{align*}
\theta & \quad \rightarrow \quad p(m|\theta) \quad \rightarrow \quad m^* \quad \rightarrow \quad f(m) \quad \rightarrow \quad \hat{\theta}^*
\end{align*}
\]

where \( f(m) \) minimizes some cost function over the posterior:

\[
p(\theta|m) = \frac{1}{p(m)} p(m|\theta) p(\theta)
\]

Bayes’ rule

\[
e.g., \quad \hat{\theta}^* = \arg\max_\theta p(\theta|m^*)
\]

maximum a posteriori (MAP) estimate
Three Ingredients for Bayesian Estimation:

1. Likelihood $p(m|\theta)$
2. Prior $p(\theta)$
3. Loss function $L(\hat{\theta}, \theta)$

jointly determine the posterior $p(\theta|m)$

- specifies how to generate an estimate $\hat{\theta}$
- "cost" of making an estimate $\hat{\theta}$
- if the true value is $\theta$
- specifies how to generate an estimate from the full posterior

Bayesian estimator is defined as:

$$\hat{\theta}(m) = \arg \min_{\hat{\theta}} \int L(\hat{\theta}, \theta) p(\theta|m) d\theta$$

"Bayes’ risk"
Typical Loss functions and Bayesian estimators

1. \( L(\hat{\theta}, \theta) = (\hat{\theta} - \theta)^2 \)  
   squared error cost function

need to find \( \hat{\theta} \) minimizing the expected loss:

\[
\int (\hat{\theta} - \theta)^2 p(\theta|m) d\theta
\]

Differentiate with respect to \( \hat{\theta} \) and set to zero:

\[
\int 2(\hat{\theta} - \theta)p(\theta|m)d\theta = 0
\]

\[
\int \hat{\theta} p(\theta|m)d\theta = \int \theta p(\theta|m)d\theta
\]

\[
\hat{\theta} = \int \theta p(\theta|m)d\theta \quad \text{“posterior mean”}
\]

also known as Bayes’ Least Squares (BLS) estimator
Typical Loss functions and Bayesian estimators

2. \( L(\hat{\theta}, \theta) = 1 - \delta(\hat{\theta} - \theta) \) “zero-one” loss (1 unless \( \hat{\theta} = \theta \))

expected loss:

\[
\int (1 - \delta(\hat{\theta} - \theta)) p(\theta|m) d\theta
\]

\[= 1 - p(\hat{\theta}|m)\]

which is minimized by:

\[\hat{\theta} = \arg\max_{\theta} p(\theta|m)\]

- posterior maximum (or “mode”).
- known as maximum a posteriori (MAP) estimate.
Note: posterior maximum and mean not always the same!
Typical Loss functions and Bayesian estimators

3. \( L(\hat{\theta}, \theta) = |\hat{\theta} - \theta| \) "L1" loss

   \[
   \text{expected cost: } \int |\hat{\theta} - \theta| \ p(\theta|m) \, d\theta
   \]

Homework problem: What is the Bayesian estimator for this cost function?
Simple Example: Gaussian noise & prior, single neuron, linear tuning curve: \( f(\theta) = \theta \)

1. Likelihood \( p(m | \theta) \)  
   additive Gaussian noise  
   
   \[
   m^* = \theta + \text{GN} \\
   p(m | \theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(m - \theta)^2}{2\sigma^2}}
   \]

2. Prior \( p(\theta) \)  
   zero-mean Gaussian  
   
   \[
   p(\theta) = \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{\theta^2}{2\gamma^2}}
   \]

3. Loss function: doesn’t matter (all agree here)
   
   posterior distribution \( p(\theta | m) = \mathcal{N} \left( \frac{\gamma^2}{\sigma^2 + \gamma^2} m, \frac{\sigma^2 \gamma^2}{\sigma^2 + \gamma^2} \right) \)
   
   MAP estimate variance
\[ m^* = \theta + GN \]
$m^* = \theta + GN$
$$m^* = \theta + \mathcal{GN}$$

$$p(m|\theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(m-\theta)^2}{2\sigma^2}}$$
\[ m^* = \theta + \mathcal{N} \]

\[ p(m|\theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(m-\theta)^2}{2\sigma^2}} \]
Prior

\[
p(\theta) = \frac{1}{\sqrt{2\pi}\gamma} e^{-\frac{\theta^2}{2\gamma^2}}
\]
Computing the posterior

\[ p(m|\theta) \cdot p(\theta) \propto p(\theta|m) \]
Making an Bayesian Estimate:

likelihood
\[ p(m|\theta) \]

prior
\[ p(\theta) \]

posterior
\[ p(\theta|m) \]

\[ \hat{\theta}_{ML} \]

\[ \hat{\theta}_{MAP} \]
High Measurement Noise: large bias

- **Likelihood**: $p(m|\theta)$
- **Prior**: $p(\theta)$
- **Posterior**: $p(\theta|m)$

Diagram showing visual representations of the likelihood, prior, and posterior distributions, along with the maximum likelihood estimate ($\hat{\theta}_{ML}$) and the maximum a posteriori estimate ($\hat{\theta}_{MAP}$). The diagram indicates a larger bias in the posterior distribution due to high measurement noise.
Low Measurement Noise: small bias

likelihood
\[ p(m | \theta) \]

prior
\[ p(\theta) \]

posterior
\[ p(\theta | m) \]

\( \hat{\theta}_{ML} \)

\( \hat{\theta}_{MAP} \)
Bayesian Estimation:

- Likelihood and prior combine to form posterior
- Bayesian estimate is always biased towards the prior (from the ML estimate)
Application #1: Biases in Motion Perception

Which grating moves faster?
Application #1: Biases in Motion Perception

Which grating moves faster?
Explanation from Weiss, Simoncelli & Adelson (2002):

- In the limit of a zero-contrast grating, likelihood becomes infinitely broad ⇒ percept goes to zero-motion.

- Claim: explains why people actually speed up when driving in fog!
Decoder properties: bias, variance, Fisher information
How to judge a decoder?

Estimate distribution

Probability

$p(\hat{s} | s)$
Good decoders

- Unbiased
- Low variance
- Can be implemented by a neural network.

\[ p(\hat{s} \mid s) \]
Bias

• difference between the stimulus and the (expected) estimate

\[
\text{bias}(\hat{s}) = \mathbb{E}[^{\hat{s} | s}] - s
\]

“unbiased” - bias is zero
variance

• expected square of the estimate minus its mean

\[
\text{var}(\hat{s}) = \mathbb{E}[(\hat{s} - \mathbb{E}[\hat{s}])^2]
\]
Consider an estimator which reports “5” regardless of the stimulus $s$

- what is the bias?
- what is the variance?
“Probabilistic Population Codes” (PPC)

• can we do more than just make a point estimate of the stimulus?
• can we compute the full posterior distribution with neurons?

Ma et al 2006, Nature Neurosci
General Question: How does the brain represent probability distributions?

(also: why would it want to?)

Two basic schemes:

1) Probabilistic Population Coding (PPC) (Ma, Beck, Pouget, Latham) - basic extension of “tuning curve” codes.

2) “Sampling Hypothesis” (Fiser, Berkes, Orbán, & Lengyel 2010 TINS) - “generative models” setup
Probabilistic Population Codes

“Poisson-like” neural noise - exponential family with linear sufficient statistics

\[ P(\vec{r}|s) = \frac{1}{Z} \exp \left[ \vec{r} \cdot \vec{w}_s \right] \]

on board:
1) demonstrate that Poisson is in this class
2) show cue combination example
“Probabilistic Population Codes” (PPC)

- more spiking gives you narrower posteriors

\[
P(s | \vec{r}) \propto P(\vec{r} | s) = \frac{1}{Z} \exp \left[ \vec{r} \cdot \vec{w}_s \right]
\]
more importantly: can do Bayesian computations with spikes

**example: cue combination**

\[
P(s|r_1) = \frac{1}{Z} \exp [r_1 \cdot h_1(s)]
\]

\[
P(s|r_2) = \frac{1}{Z} \exp [r_2 \cdot h_2(s)]
\]

**combined posterior**

\[
P(s|r_1, r_2) \propto p(r_1|s)p(r_2|s)
\]

\[
\propto \exp [r_1 \cdot h_1(s) + r_2 \cdot h_2(s)]
\]

\[
= \exp [(r_1 + r_2) \cdot h(s)]
\]

\[
= \exp [r_3 \cdot h(s)]
\]

**combined representation**

no need to explicitly decode
Dynamic codes: what about time?

linear readout model: \[ M = WN \]

\[
\begin{align*}
\text{muscles} & \quad \begin{bmatrix} 1 \\ m \end{bmatrix} & \quad \begin{bmatrix} \text{time} \\ T \end{bmatrix} \\
\text{neuron} & \quad \begin{bmatrix} 1 \\ n \end{bmatrix} & \quad \begin{bmatrix} \text{neuron-muscle} \\ \text{weights} \end{bmatrix} \\
\text{neurons} & \quad \begin{bmatrix} 1 \\ n \end{bmatrix} & \quad \begin{bmatrix} \text{time} \\ T \end{bmatrix}
\end{align*}
\]
Dynamic codes: what about time?

linear readout model: \( M = WN \)

Approach 1:

1. Dimensionality reduction: PCA

2. Regression: \( \hat{W} = \arg\min_W ||M - WN||^2 \)

- Kaufman, Churchland, Ryu, & Shenoy - NN 2014 (pre- vs. peri-movement activity)
data schematic

hand position

time

neural activity (rate)

- T: target
- G: go cue

- preparatory period
- movement period

- Kaufman, Churchland, Ryu, & Shenoy - NN 2014 (pre- vs. peri-movement activity)
The core logic of this analysis is to use electromyography (EMG) to quantify movement-related activity. As in the model, the analysis in equation (1) uses a linear combination for identifying which dimensions are output-null versus output-potent. We therefore designed a mathematical method for estimating from EMG recordings. We applied a two-dimensional view to preparatory activity of left neuron. Targ, target onset; Move, movement onset. (Fig. 3a)

Two caveats are worth stressing. First, a two-dimensional view may be insufficient to fully test the hypothesis. Second, to properly interpret these results, one would wish to have some independent means of differentiating movement-related activity from preparatory activity of left neuron. Targ, target onset; Move, movement onset. (Fig. 3b)

With the dimensions identified, we predicted that preparatory activity should avoid leaking into the output-potent dimensions and should mainly be present in output-null dimensions. To avoid this, we subtracted neurons' responses (see Fig. 3c).

- Kaufman, Churchland, Ryu, & Shenoy - NN 2014 (pre- vs. peri-movement activity)

Population analysis

Illustrative pair:
Approach 2: Dynamic models of neural activity

\[
\begin{bmatrix}
1 & \text{time} & T \\
m
\end{bmatrix}
\begin{bmatrix}
1 \\
m
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
n
\end{bmatrix}
\begin{bmatrix}
\text{neuron-muscle weights}
\end{bmatrix}
\]

Dynamics matrix

\[ \vec{r}_t = A \vec{r}_{t-1} + \text{noise} \]

Q: what would the eigenvalues of A tell us about this system?
Approach 2: Dynamic models of neural activity

Dynamics matrix

\[ \vec{r}_t = A \vec{r}_{t-1} + \text{noise} \]

Q: what would the eigenvalues of A tell us about this system?
Looking for oscillatory dynamics (where they ought not exist)

“jPCA” - find low-rank A matrix with purely imaginary eigenvalues

Churchland, Cunningham et al
Nature 2012

Churchland et al, Neuron 2010

~1/8 real time
Looking for oscillatory dynamics (where they ought not exist)

~1/8 real time

Churchland et al, Neuron 2010
Looking for oscillatory dynamics (where they ought not exist)

~1/8 real time

Churchland et al, Neuron 2010
Population Coding

• tuning curves, noise distributions

• population decoders (winner-take-all, center-of-mass, template-matching, maximum-likelihood, Bayesian)

• Bayesian decoding: 3 ingredients (likelihood, prior, loss function), explains perceptual biases

• PPC - probabilistic population codes: use tuning curves and “exponential family noise with linear sufficient statistics” to represent the posterior

• Dynamic codes:
  - dimensionality reduction (PCA) (dimensionality reduction)
  - regression (linear decoding)
  - auto-regressive models (dynamics)