Probabilistic approaches to neural computations

CA6b, Lecture 2
22/02/2013
All of our decisions are subject to uncertainty.
The Bayesian Brain

\[
P(x | s) = \frac{P(s | x) P(x)}{P(s)}
\]
Poisson Variability in Cortex

Variance of Spike Count

Mean Spike Count

Trial 1
Trial 2
Trial 3
Trial 4
Cortical spike trains are variable

From Churchland et al, Nature neuroscience 2010
Continuous variable: Population coding

Georgopoulos 1982
Population Codes

Tuning Curves

Average pattern of activity

\[ \langle s_i \rangle = f(x - x_i) = f_i(x) \]
Noisy population Codes

Pattern of activity \((s)\)

Poisson noise:

\[
p(s_i \mid x) = \frac{f_i(x)^{s_i} \exp(-f_i(x))}{s_i!}
\]

Independent:

\[
p(s \mid x) = \prod_i p(s_i \mid x)
\]
Perception as estimation

\[ \hat{x} = f(s) \]
Likelihood

Pattern of activity \((s)\)

\[
\log(p(s|x)) = \sum_i \log(f_i(x))s_i - \sum_i f_i(x)
\]
Maximum-Likelihood

Pattern of activity ($s$)

\[
\frac{\delta}{\delta x} \log(p(s|x)) = \sum_i \frac{f'_i(x)}{f_i(x)} s_i - \sum_i f'_i(x) = 0
\]
Maximum-Likelihood

Pattern of activity (s)

Very hard to compute…
How precise in a population code?

Pattern of activity ($s$)

Fisher Information:

$$I_F = \langle \left( \frac{\delta}{\delta x} \log(p(s|x)) \right)^2 \rangle$$

Cramer-Rao inequality:

$$\langle (x - \hat{x})^2 \rangle \geq \frac{1}{I_F}$$

$$I_F = \sum_i \frac{f_i'(x)^2}{f_i(x)}$$
The Bayesian Brain

$$P(x|x) = \frac{P(s|x)P(x)}{P(s)}$$
Posterior probability

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

\[ \log(p(x|s)) = L_o(x) + \sum_i \log(f_i(x))s_i - \sum_i f_i(x) \]
Probabilistic population code

Pattern of activity \( s \)

\[
\log(p(x_j | s)) = L_o(x_j) + \sum_i \log(f_i(x_j)) s_i - \sum_i f_i(x_j)
\]
Pattern of activity ($s$)

$$\log \left(p(x | s)\right) + Z$$

$$L_j = \sum_i w_{ij} s_i - \theta_j$$

Posterior

Preferred Direction (deg)

Activity

Synaptic weights

Bias
Generalization

Exponential family:

\[ p(s \mid x) = Z(s) g(x) \exp(h(x).s) \]

Linear read-out:

\[ \log(p(x_j \mid s)) = \sum_j w_{ij} s_j + \theta_j \]

Indepedendant poisson neurons

\[ w_{ij} = \log(f_i(x_j)) \quad \theta_j = \sum_i f_i(x_j) \]
For neural noise in exponential family, cue combination is equivalent to summing activities

\[
L_j^v = \sum_i w_{ij} s_i^v - \theta_j^v
\]

\[
L_j^a = \sum_l k_{lj} s_l^a - \theta_j^a
\]

\[
L_j^b = \sum_l k_{lj} s_l^a + \sum_i w_{ij} s_i^v - \theta_j^a - \theta_j^v
\]
For neural noise in exponential family, cue combination is equivalent to summing activities.

Visual input:

$\boldsymbol{p}(x | \boldsymbol{r}_{vis})$

Auditory input:

$\boldsymbol{p}(x | \boldsymbol{r}_{aud})$

Multisensory

$\boldsymbol{p}(x | \boldsymbol{r}_{bi})$
For neural noise in exponential family, cue combination is equivalent to summing activities

\[ L^v = Ws^v - \theta^v \]

\[ L^a = Ks^a - \theta^a \]

\[ Hs^b = Ks^a + Ws^v - \Theta^b \]

\[ s^b = H^{-1}Ks^a + H^{-1}Ws^v - H^{-1}\Theta^b \]
cue combination

$f_{i}^{1}(x)$

$f_{i}^{2}(x)$

$f_{i}^{3}(x)$

$p(x \mid s_{1}, s_{2})$
Generalization (Beck, Latham, Pouget 2006)

- Linear combinations correspond to optimal cue integration as long as the likelihood belongs to the exponential family:

\[ p(s \mid x) = Z(s) g(x) \exp(h(x) \cdot s) \]

In which case a linear combination can perform optimal cue combination:

\[ p(x \mid A_1 s_1 + A_2 s_2) = \frac{1}{Z} p(x \mid s_1) p(x \mid s_2) \]

At all stages, the neural responses could be decoded with a line attractor.

Alternative neural code for uncertainty: sampling
Alternative neural code for uncertainty: sampling

Berkes et al, Science 2011
Alternative neural code for uncertainty: sampling
Representation of uncertainty

Probabilistic population code

Increases response gain

Sampling code

Decreases response variance
Sampling: Cue combination hard, Marginalisation easy

\[
\hat{p}(x|s) = \sum_t x_s^t \\
\hat{p}(y|s) = \sum_x p(y|x) \hat{p}(x|s) = \sum_{x,t} w_{xy} x_s^t
\]
PPC: cue combination easy, marginalization hard

\[ f_i^1(x) \]

\[ f_i^2(x) \]

\[ f_i^3(x) \]
Temporal integration

- Events start and end unpredictably, motor effects move...

- We need to estimate the states of relevant variable on-line.
Stimulus: $x_{t-dt} \rightarrow x_t \rightarrow x_{t+dt}$

Input spikes:

$S_{t-dt} \rightarrow S_t \rightarrow S_{t+dt}$

Graphical representation:

- Nodes: $X_t$, $L_t$, $S^1_t$, $S^2_t$
- Edges: $X_t \rightarrow S^1_t$, $X_t \rightarrow S^2_t$, $S^1_t \rightarrow L_t$, $S^2_t \rightarrow L_t$

Time series:

- $X_t$: Pink, periodic spikes
- $S^1_t$: Blue, sparse spikes
- $S^2_t$: Blue, dense spikes
Probability of a spike during interval $dt$:

$$p(s \mid x = 1) = \frac{(q_1 dt)^s \exp(-q_1 dt)}{s!}$$

$$p(s \mid x = 0) = \frac{(q_0 dt)^s \exp(-q_0 dt)}{s!}$$

Log Likelihood Ratio:

$$\log \left( \frac{p(s \mid x = 1)}{p(s \mid x = 0)} \right)$$
Probability of a spike during interval $dt$:

$$p(s | x = 1) = \frac{(q_1 dt)^s \exp(-q_1 dt)}{s!}$$

$$p(s | x = 0) = \frac{(q_0 dt)^s \exp(-q_0 dt)}{s!}$$

Log Likelihood Ratio:

$$\log \left( \frac{p(s | x = 1)}{p(s | x = 0)} \right) = \log \left( \frac{q_1}{q_0} \right)^s + q_0 - q_1$$
Probability of a spike during interval $dt$:

\[ p(s \mid x = 1) = \frac{(q_1 dt)^s \exp(-q_1 dt)}{s!} \]
\[ p(s \mid x = 0) = \frac{(q_0 dt)^s \exp(-q_0 dt)}{s!} \]

Log Likelihood Ratio:

\[ \log \left( \frac{p(s \mid x = 1)}{p(s \mid x = 0)} \right) = w s - \theta \]
Likelihood

Probability of a spike during interval $dt$:

$$p(s_i \mid x = 1) = \frac{(q^i dt)^{s_i} \exp(-q^i dt)}{s_i!}$$

$$p(s_i \mid x = 0) = \frac{(q^0 dt)^{s_i} \exp(-q^0 dt)}{s_i!}$$

$$\log \left( \frac{p(s_1 \ldots s_N \mid x = 1)}{p(s_1 \ldots s_N \mid x = 0)} \right) = \sum_{i=1}^{N} \omega_i s_i - \theta$$
Temporal integration

Rate of switching on: \[ r_{on} = \frac{1}{dt} p(x_{t+dt} = 1 | x_t = 0) \]

Rate of switching off: \[ r_{off} = \frac{1}{dt} p(x_{t+dt} = 0 | x_t = 1) \]

\[ q^i_1 = \frac{1}{dt} P(s^i_t = 1 | x_t = 1) \]
\[ q^i_0 = \frac{1}{dt} P(s^i_t = 1 | x_t = 0) \]

\[ L_t = \log \left( \frac{p(x_t = 1 | s_{0\rightarrow t+dt})}{p(x_t = 0 | s_{0\rightarrow t+dt})} \right) \]
Temporal integration

\[ \frac{\partial L}{\partial t} = r_{on} \left( 1 + e^{-L} \right) - r_{off} \left( 1 + e^{L} \right) + \sum_i w_i s_t^i - \theta \]

Deneve, 2001
Synaptic integration = Bayesian filtering?

\[
\frac{\partial L}{\partial t} = -\varphi'(L_t) + \sum_i w_i s_t^i
\]

Leak  Synaptic input
Accumulation of evidence results in linear ramps followed by a saturation.

A. Slow dynamics = small leak

B. Fast dynamics = strong leak

\[ \frac{\partial L}{\partial t} = -\varphi'(L_t) + \sum w_i s^i_t \]

Leak \hspace{1cm} Synaptic input
Accumulation of evidence results in linear ramps followed by a saturation.

Stimulus never changes (zero probability of transition) = no leak

\[ \frac{\partial L}{\partial t} = \sum_i w_i s_i^t \]

Synaptic input
Evidence for “explicit” encoding of probability in neural activity: Shadlen et al

Rightward motion: saccade right

Leftward motion: saccade left
Evidence for “explicit” encoding of probability in neural activity: Shadlen et al

Medio-temporal cortex (MT):

Lateral intra-parietal cortex (LIP)
LIP/MT model (sequential probability test)

Right > Left

Saccade right

Saccade Left

\[ L_t = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]

\[ L_t = \log \left( \frac{p(\text{left})}{p(\text{right})} \right) \]
LIP/MT model (sequential probability test)

**MT**
- Right > Left

**LIP**
- Saccade right
- Saccade Left

\[
L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right)
\]

\[
L_l = \log \left( \frac{p(\text{left})}{p(\text{right})} \right)
\]
LIP/MT model (sequential probability test)

Right > Left

Saccade right

Left > Right

Saccade Left

$L_i = \log \left( \frac{p(\text{right})}{p(\text{left})} \right)$
LIP/MT model (sequential probability test)

L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right)

L_r = \log \left( \frac{p(\text{left})}{p(\text{right})} \right)
LIP/MT model (sequential probability test)

$L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right)$

$L_l = \log \left( \frac{p(\text{left})}{p(\text{right})} \right)$

Right $>$ Left

Saccade right

Saccade Left

Left $>$ Right
LIP/MT model (sequential probability test)

Right > Left

Saccade right

Saccade Left

Left > Right

$L_i = \log \left( \frac{p(right)}{p(left)} \right)$

$L_i = \log \left( \frac{p(left)}{p(right)} \right)$
LIP/MT model (sequential probability test)

\[ L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]

\[ L_l = \log \left( \frac{p(\text{left})}{p(\text{right})} \right) \]
LIP/MT model (sequential probability test)

Right > Left

Saccade right

\[ L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]

Left > Right

Saccade Left

\[ L_r = \log \left( \frac{p(\text{left})}{p(\text{right})} \right) \]
Bayesian model fits behavior?

To some extent…

![Graph showing accuracy and reaction time against motion strength](image)

![Graph showing log odds against time](image)

Right Decision threshold

Log odds

Left

LIP
Importance of using the right temporal statistics

Left > Right

Right > Left
LIP/MT model

Rightward motion

Leftward motion

Right > Left

Left > Right

Firing rate

Throw a coin

\[ L_r = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]
LIP/MT model

Rightward motion

Leftward motion

Right > Left

Left > Right

Integrate and fire

\[ L_t = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]
LIP/MT model

Right>Left

Rightward motion

Left>Right

Leftward motion

\[ L_i = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]

Integrate and fire
LIP/MT model

Right > Left

Rightward motion

Leftward motion

Left > Right

\[ L_i = \log \left( \frac{p(\text{right})}{p(\text{left})} \right) \]

Integrate and fire
Deterministic firing

\[ \frac{\partial L}{\partial t} = -\varphi'(L_t) + \sum_i w_i s_i^t \]

\[ \frac{\partial G}{\partial t} = -\varphi'(G_t) + O_t \]
Mechanism for spike generation in Bayesian neurons

Integrate input (what I know)

\[ O_t \]

Integrate output (what I told)

\[ \frac{\partial L}{\partial t} = -\varphi'(L_i) + \sum w_is_i^t \]

\[ \frac{\partial G}{\partial t} = -\varphi'(G_i) + g_oO_i \]

(What I told in the past)

(What I know)
Mechanism for spike generation in Bayesian neurons

- Integrate input (what I know)
- Integrate output (what I told)

Integrator with adapting threshold

Integrate and fire with adaptive time constant

\[ V_t = L_t - G_t \]
Implication 2: Firing rate statistics are similar to Poisson, but these fluctuations purely reflects input noise.

The spike train is a deterministic function of the input.
Spikes are binary decisions. The tradeoff between accuracy (detection small fluctuations) and speed (delay, output firing rate) is controlled by the reset.
Using a deterministic firing process gives tremendous advantage compared to a rate model by limiting information loss.
Implication: Spikes signal an increase in probability

A spike signals an increase in the probability of a variable.

Deneve, 2001