Probabilistic Population Codes

Kristjan Kalm *kristjan.kalm@mrc-cbu.cam.ac.uk*

Medical Research Council, Cognition & Brain Sciences Unit

February 25, 2014

Overview

- How can neuronal populations encode probability distributions?

Overview

- How can neuronal populations encode probability distributions?
- Parametric approach - probabilistic population codes (PPC)

Overview

- How can neuronal populations encode probability distributions?
- Parametric approach - probabilistic population codes (PPC)
- Non-parametric - sampling

PPC -

ARTICLES

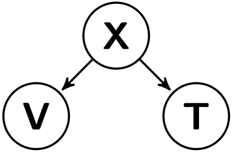
nature neuroscience

Bayesian inference with probabilistic population codes

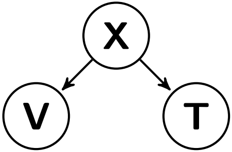
Wei Ji Ma[1,3], Jeffrey M Beck[1,3], Peter E Latham[2] & Alexandre Pouget[1]

Recent psychophysical experiments indicate that humans perform near-optimal Bayesian inference in a wide variety of tasks, ranging from cue integration to decision making to motor control. This implies that neurons both represent probability distributions and combine those distributions according to a close approximation to Bayes' rule. At first sight, it would seem that the high variability in the responses of cortical neurons would make it difficult to implement such optimal statistical inference in cortical circuits. We argue that, in fact, this variability implies that populations of neurons automatically represent probability distributions over the stimulus, a type of code we call probabilistic population codes. Moreover, we demonstrate that the Poisson-like variability observed in cortex reduces a broad class of Bayesian inference to simple linear combinations of populations of neural activity. These results hold for arbitrary probability distributions over the stimulus, for tuning curves of arbitrary shape and for realistic neuronal variability.

PPC - Perception as Bayesian inference



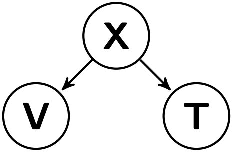
PPC - Perception as Bayesian inference



- Bayesian fusion of sensory information produces a posterior density

$$p(x|v,t) \propto p(v|x) \cdot p(t|x) \cdot p(x)$$

PPC - Perception as Bayesian inference



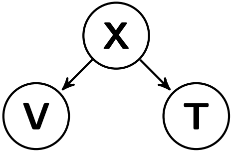
- Bayesian fusion of sensory information produces a posterior density

$$p(x|v,t) \propto p(v|x) \cdot p(t|x) \cdot p(x)$$

- Brain is a Bayesian decoder

$$p(s|r_1,r_2) \propto p(r_1|s) \cdot p(r_2|s) \cdot p(s)$$

PPC - Perception as Bayesian inference



- Bayesian fusion of sensory information produces a posterior density

$$p(x|v,t) \propto p(v|x) \cdot p(t|x) \cdot p(x)$$

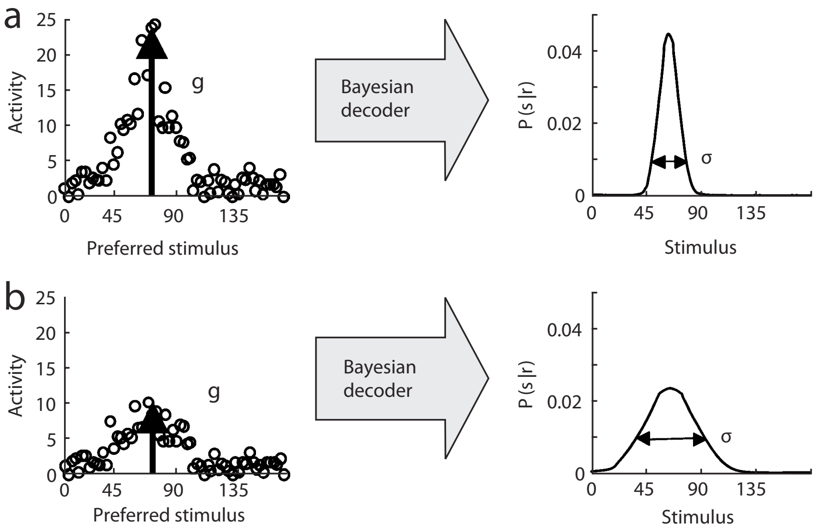
- Brain is a Bayesian decoder

$$p(s|r_1, r_2) \propto p(r_1|s) \cdot p(r_2|s) \cdot p(s)$$

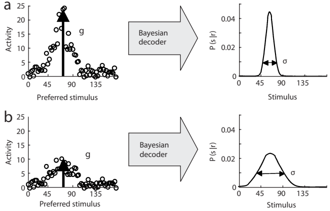
- Neurons have tuning curves

$$p(r|s) = \mathcal{N}(\mu, \sigma^2)$$

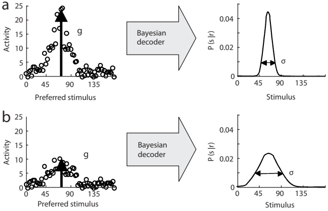
PPC - Tuning curves



PPC - Tuning curves

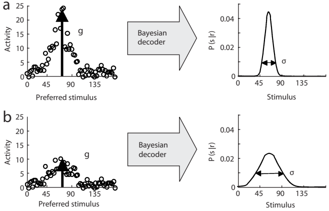


PPC - Tuning curves



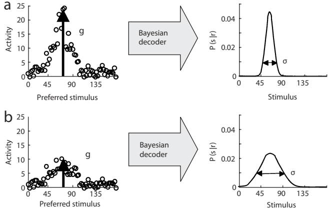
- How do neurons represent probability distributions?

PPC - Tuning curves



- How do neurons represent probability distributions?
- How do neurons compute the posterior?

PPC - Tuning curves

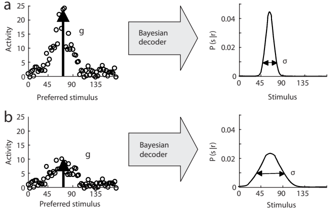


- How do neurons represent probability distributions?
- How do neurons compute the posterior?

- Simple posterior over the stimulus

$$p(s|r) \propto p(r|s) \cdot p(s)$$

PPC - Tuning curves



- How do neurons represent probability distributions?
- How do neurons compute the posterior?

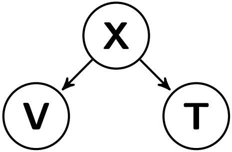
- Simple posterior over the stimulus

$$p(s|r) \propto p(r|s) \cdot p(s)$$

- Independent Poisson neural variability

$$p(s|r) \propto \prod_i \frac{e^{-f_i} f_i^{r_i}}{r_i!} \cdot p(s)$$

PPC - Perception as Bayesian inference



- Bayesian fusion of sensory information produces a posterior density

$$p(x|v,t) \propto p(v|x) \cdot p(t|x) \cdot p(x)$$

- Brain is a Bayesian decoder

$$p(s|r_1, r_2) \propto p(r_1|s) \cdot p(r_2|s) \cdot p(s)$$

- Neurons have tuning curves

$$p(r|s) = \mathcal{N}(\mu, \sigma^2)$$

Vision and touch

For a Gaussian likelihood with mean m_d and precision λ_d and a Gaussian prior with mean m_0 and precision λ_0 the posterior is a Gaussian with

$$m = \frac{\lambda_d}{\lambda} m_d + \frac{\lambda_0}{\lambda} m_0$$

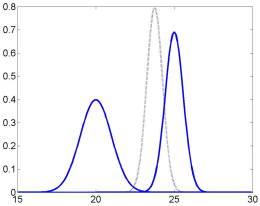
$$\lambda = \lambda_d + \lambda_0$$

Vision and touch

For a Gaussian likelihood with mean m_d and precision λ_d and a Gaussian prior with mean m_0 and precision λ_0 the posterior is a Gaussian with

$$m = \frac{\lambda_d}{\lambda} m_d + \frac{\lambda_0}{\lambda} m_0$$

$$\lambda = \lambda_d + \lambda_0$$



The two solid curves show the probability densities for the prior $m_0 = 20$, $\lambda_0 = 1$ and the likelihood $m_d = 25$ and $\lambda_d = 3$. The dotted curve shows the posterior distribution with $m = 23.75$ and $\lambda = 4$. The posterior is closer to the likelihood because the likelihood has higher precision.

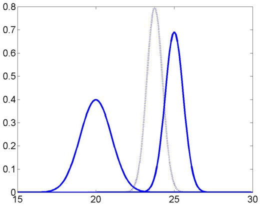
$$23.75 = \frac{3}{4}25 + \frac{1}{4}18$$

Vision and touch

For a Gaussian likelihood with mean m_d and precision λ_d and a Gaussian prior with mean m_0 and precision λ_0 the posterior is a Gaussian with

$$m = \frac{\lambda_d}{\lambda} m_d + \frac{\lambda_0}{\lambda} m_0$$

$$\lambda = \lambda_d + \lambda_0$$



- Precisions add
- The resulting mean is the sum of the means, each weighted by their relative precision

Vision and touch

$$m_{vt} = \frac{\lambda_v}{\lambda_{vt}} m_v + \frac{\lambda_t}{\lambda_{vt}} m_t$$

$$\lambda_{vt} = \lambda_v + \lambda_t$$

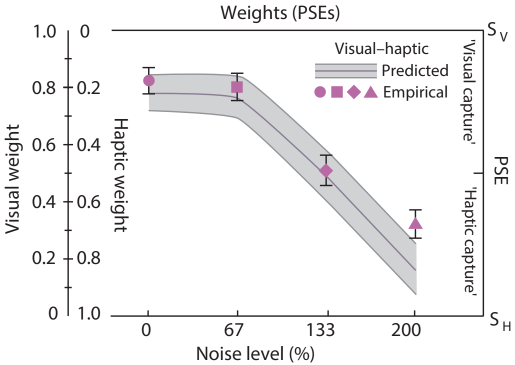
$$m_{vt} = w_v m_v + w_t m_t$$

Vision and touch

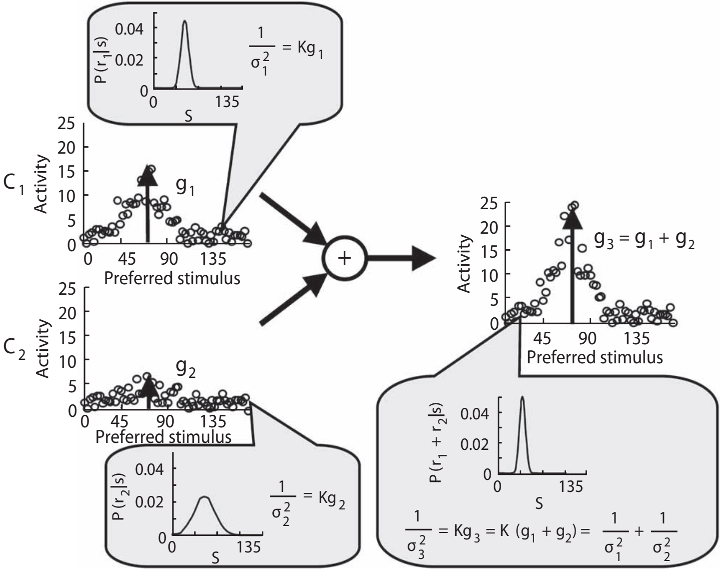
$$m_v t = \frac{\lambda_v}{\lambda_v t} m_v + \frac{\lambda_t}{\lambda_v t} m_t$$

$$\lambda_v t = \lambda_v + \lambda_t$$

$$m_v t = w_v m_v + w_t m_t$$



Vision and touch



Summary

- The summation is Bayes-optimal, if all distributions are Poisson-like

Summary

- The summation is Bayes-optimal, if all distributions are Poisson-like
- Besides linear summation works also with integrate-and-fire neurons

Summary

- The summation is Bayes-optimal, if all distributions are Poisson-like
- Besides linear summation works also with integrate-and-fire neurons

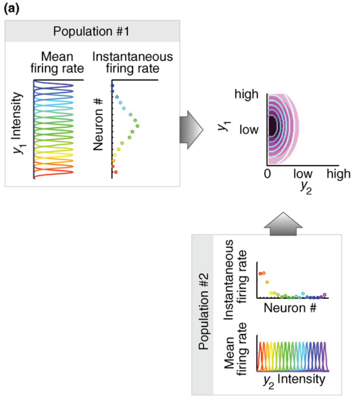
Table I. Comparing characteristics of the two main modeling approaches to probabilistic neural repre

	PPCs
Neurons correspond to	Parameters
Network dynamics required (beyond the first layer)	Deterministic
Representable distributions	Must correspond to a particular parametric form
Critical factor in accuracy of encoding a distribution	Number of neurons
Instantaneous representation of uncertainty	Complete, the whole distribution is represented at any time
Number of neurons needed for representing multimodal distributions	Scales exponentially with the number of dimensions
Implementation ofl earning	Unknown

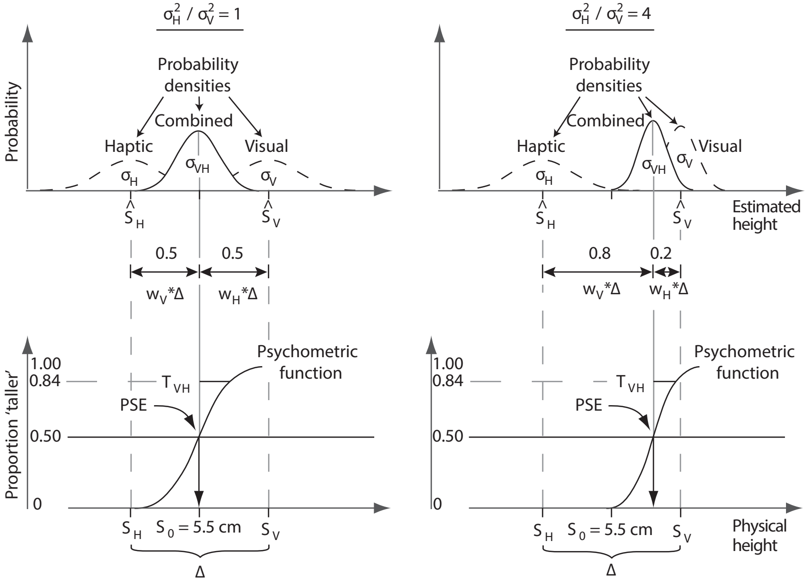
Summary

Table I. Comparing characteristics of the two main modeling approaches to probabilistic neural representations

	PPCs	Sampling-based
Neurons correspond to	Parameters	Variables
Network dynamics required (beyond the first layer)	Deterministic	Stochastic (self-consistent)
Representable distributions	Must correspond to a particular parametric form	Can be arbitrary
Critical factor in accuracy of encoding a distribution	Number of neurons	Time allowed for sampling
Instantaneous representation of uncertainty	Complete, the whole distribution is represented at any time	Partial, a sequence of samples is required
Number of neurons needed for representing multimodal distributions	Scales exponentially with the number of dimensions	Scales linearly with the number of dimensions
Implementation ofl earning	Unknown	Well-suited



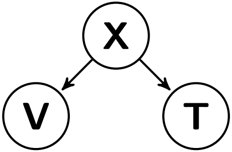
Vision and touch



Vision and touch

$$p(v|x) = \mathcal{N}(\mu, \sigma^2)$$

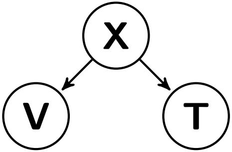
$$p(t|x) = \mathcal{N}(\mu, \sigma^2)$$



Vision and touch

$p(v|x) = \mathcal{N}(\mu, \sigma^2)$

$p(t|x) = \mathcal{N}(\mu, \sigma^2)$



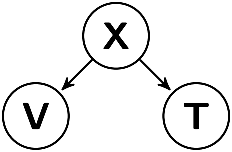
Ernst and Banks use precision instead of variance. Precision is inverse variance

$\lambda = \frac{1}{\sigma^2}$

Vision and touch

$$p(v|x) = \mathcal{N}(\mu, \sigma^2)$$

$$p(t|x) = \mathcal{N}(\mu, \sigma^2)$$



Ernst and Banks use precision instead of variance. Precision is inverse variance
$\lambda = \frac{1}{\sigma^2}$

For a Gaussian likelihood with mean m_d and precision λ_d and a Gaussian prior with mean m_0 and precision λ_0
the posterior is a Gaussian with

$$m = \frac{\lambda_d}{\lambda} m_d + \frac{\lambda_0}{\lambda} m_0$$

$$\lambda = \lambda_d + \lambda_0$$