Ideas

Nicolas Luarte

On free energy

Sample space The sample space of an experiment or random trial is the set of all possible outcomes or results of that experiment; denoted Ψ

Random fluctuation An outcome of a random variable has some fluctuation, and such fluctuations are inversely proportional to the square root of the number of processes

random fluctuations are $\omega \in \Omega$, it seems that Friston uses '$\sim$' to denote an empirical realization

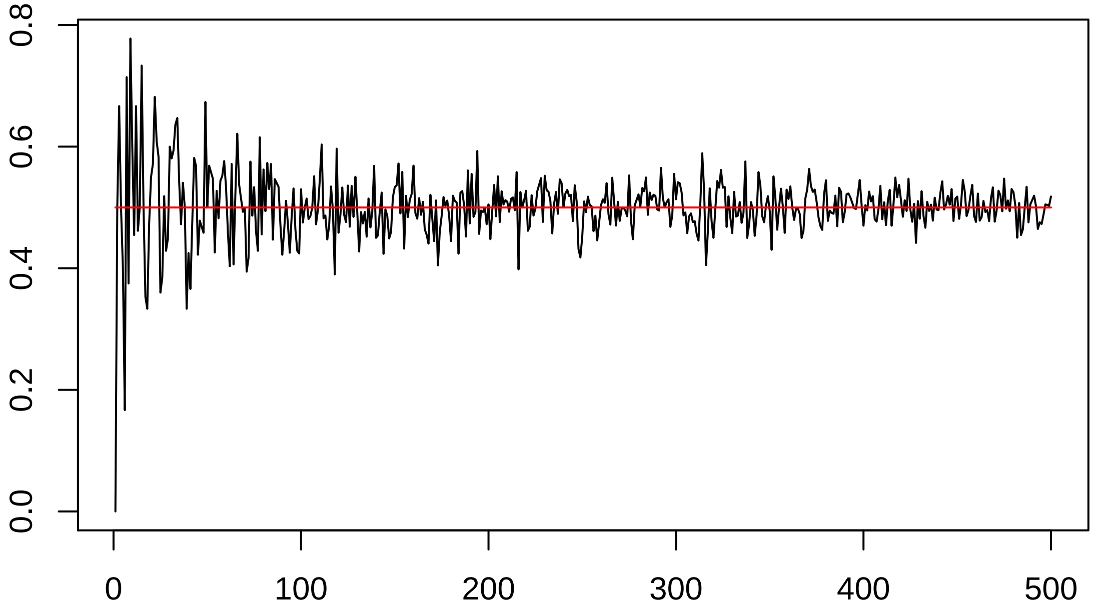
Hidden or external states

$$\Psi : \Psi \times A \times \Omega \to \mathbb{R}$$

A function which maps hidden states, actions and sample space into a 1-dimensional probability vector (?)

Example of random fluctuation

```r
library(purrr)
sample.space <- c(0, 1) # simulated coin toss
theta <- 0.5 # fair coin toss
N <- 1:500 # vector specifying number of coin flips
flips <- N %>% map(function(x) sample(sample.space,
                                      size = x,
                                      replace = TRUE,
                          prob = c(theta, 1 - theta)))
ratios <- flips %>% map(mean) # calculate the ratios
plot(unlist(ratios), type = "l", xlab = "", ylab = "")
lines(rep(0.5, length(N)), col = "red")
```



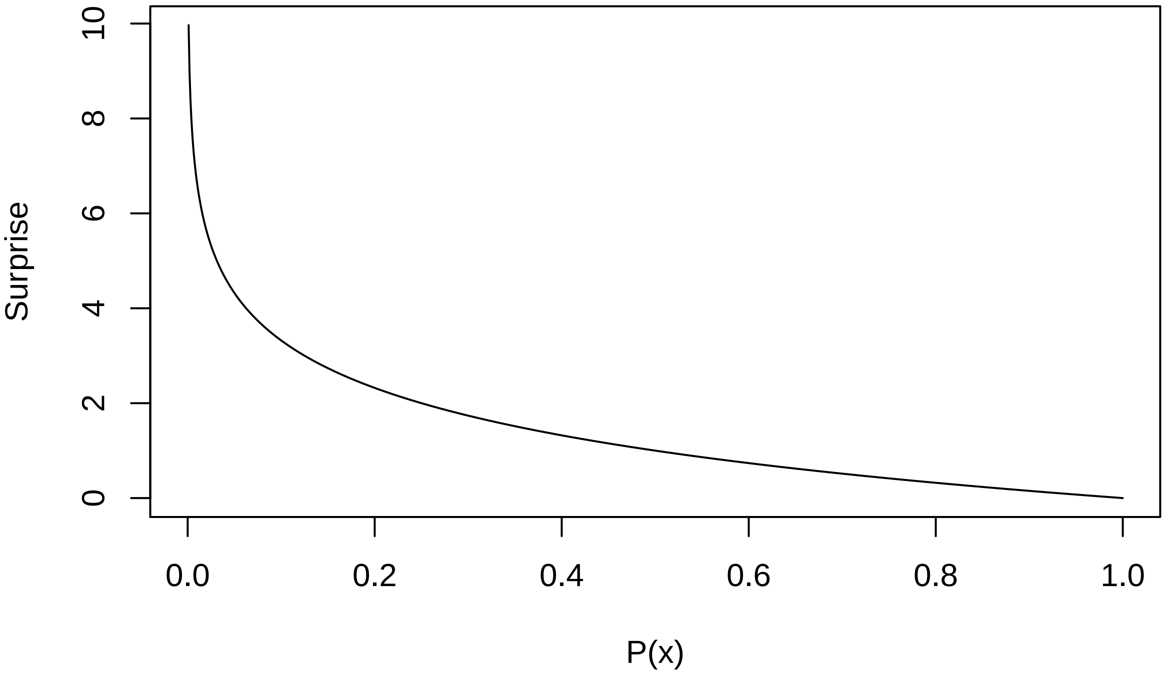
Entropy of a 'hidden markov models'

- HMMs are used to find the most likely hidden state sequence that produces a given sequence of observations
- The entropy of a random variable provides a measure of its uncertainty
- A Markov Chain S_i, represented by an $N \times N$ stochastic matrix A, which describes the transition probabilities $a_{ij} = P(S_t = j|S_{t-1} = i)$ between the N states of the model, together with a probability distribution π, where $\pi_i = P(S_1 = i)$
- A set of probability distributions, one for each hidden state, $b_i(o_j) = P(O_t = o_j|S_t = i)$, that model the emission of such observations (it models the true hidden state with an 'emission' given a certain state i at time t). If there are M possible distinct observations (true hidden states), we accommodate the probability distributions b_i in the rows of an $N \times M$ matrix B. Note that M is not necessarily bounded

Surprise

- Less likely messages provide more information

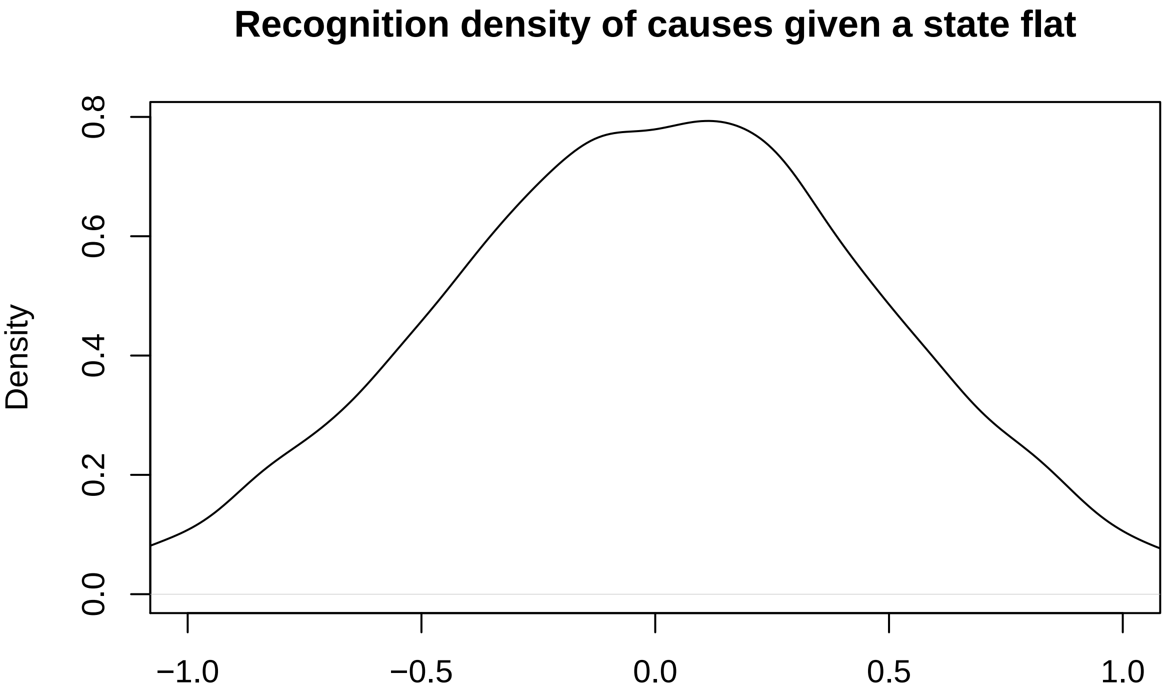
```r
probs <- seq(from = 0, to = 1, by = 0.001)
surprise_vector <- log2((1 / probs))
plot(x = probs, y = surprise_vector, type = "l", ylab = "Surprise", xlab = "P(x)")
```



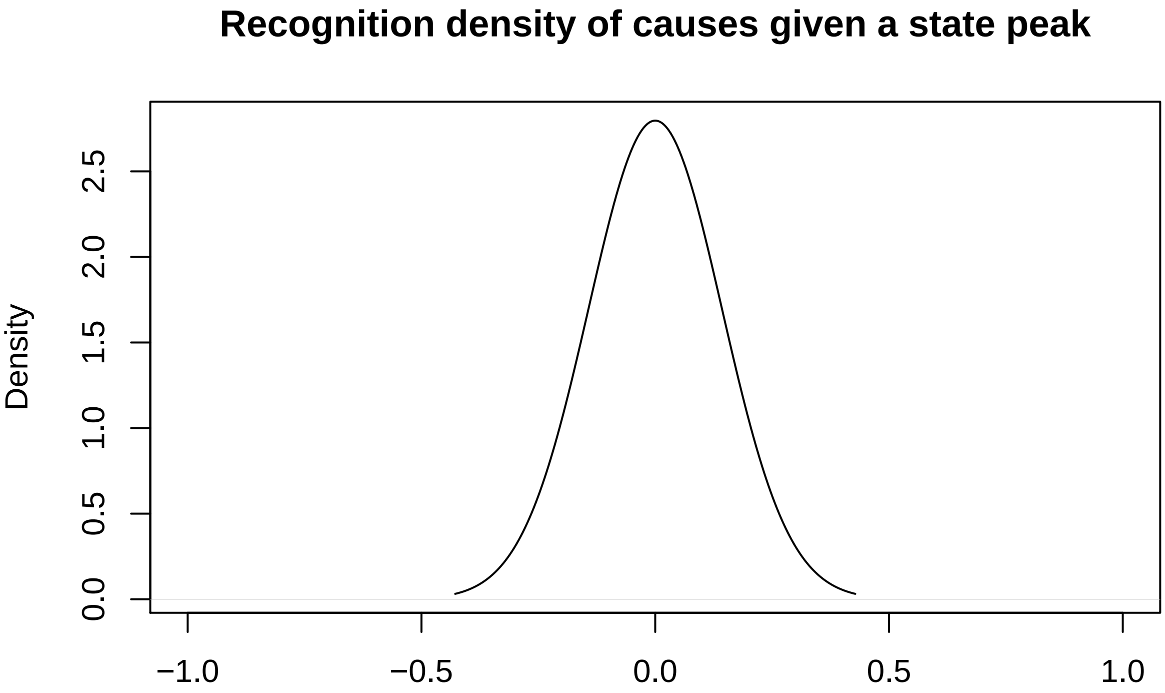
Recognition density

- First formulation expresses free energy as energy minus entropy. Free energy can be evaluated by an agent because the energy is the surprise about the joint occurrence of sensations and their perceived causes, whereas the entropy is simply the of the agent's own recognition density
- The energy is the surprise about the joint occurrence of sensations
- free energy as energy minus entropy

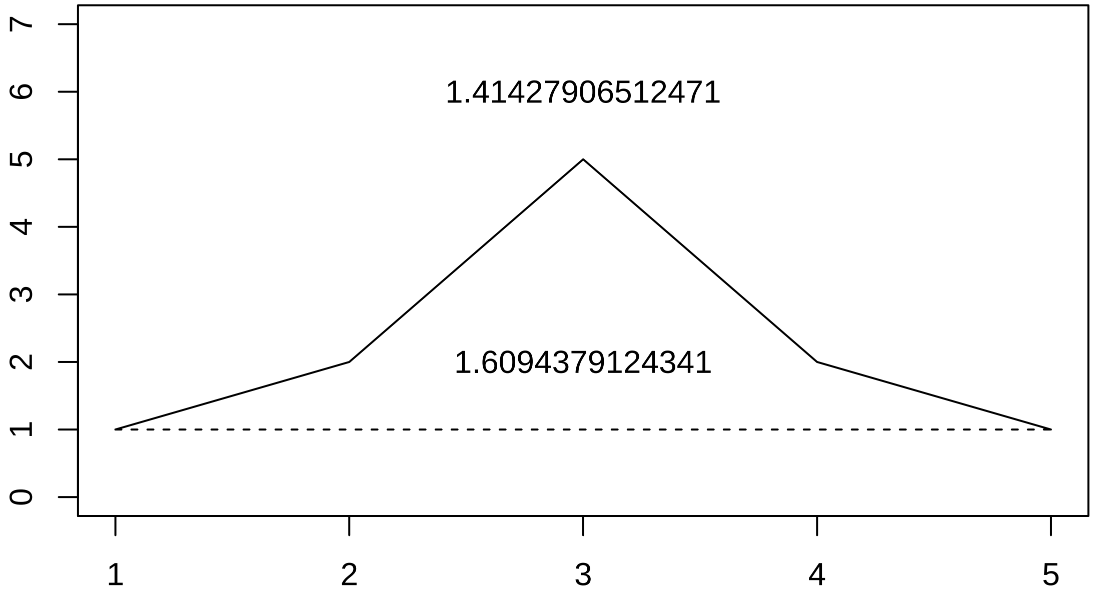
```
library(entropy)
recognition_density_flat <- density(rnorm(10000, sd = 0.5))
recognition_density_peak <- density(rnorm(10000, sd = 0))
plot(recognition_density_flat,
             xlim = c(-1, 1),
             xlab = "",
             main = "Recognition density of causes given a state flat")
```



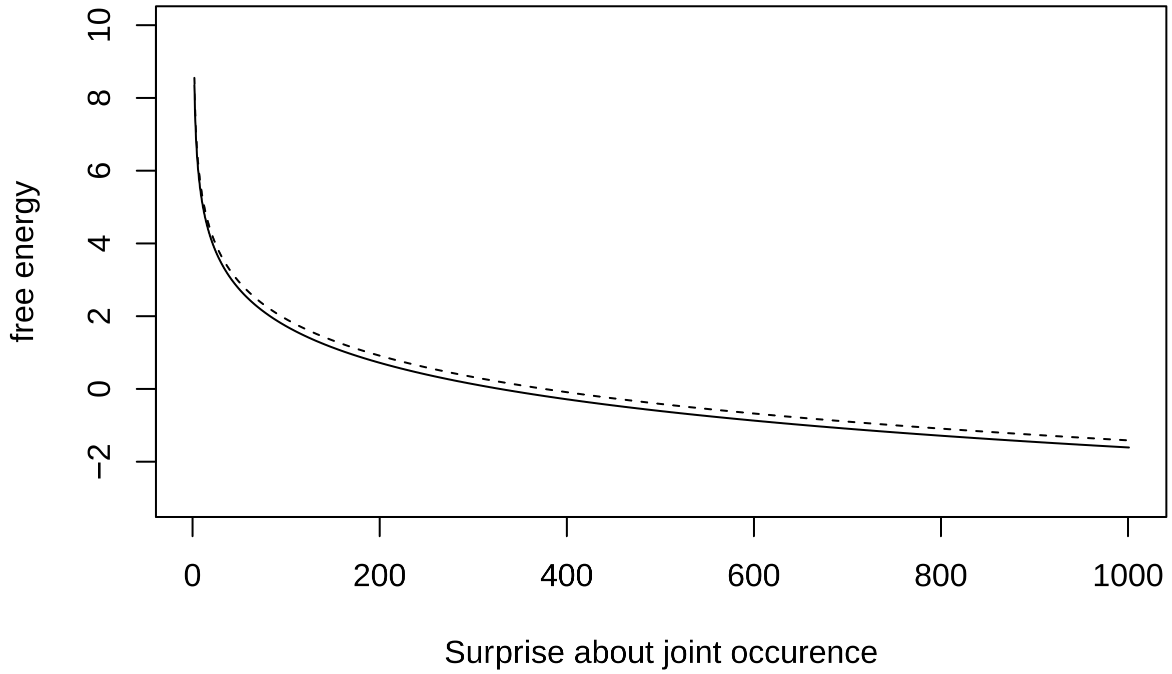
```r
plot(recognition_density_peak,
             xlim = c(-1, 1),
             xlab = "",
             main = "Recognition density of causes given a state peak")
```



```
## Calculate the entropy for peak or flat density
library(entropy)
flat_density <- c(1, 1, 1, 1, 1)
peaked_density <- c(1, 2, 5, 2, 1)
plot(flat_density, type = "l", ylim = c(0, 7), lty = 2, ylab = "", xlab = "")
lines(peaked_density)
text(x = 3, y = 6, label = entropy(peaked_density))
text(x = 3, y = 2, label = entropy(flat_density))
```



```
## plot free energy
## joint occurrence (probability) of sensations and perceived causes
joint_ocurrence <- seq(from = 0, to = 1, by = 0.001)
energy <- log2((1 / joint_ocurrence))
free_energy_flat <- energy - entropy(flat_density)
free_energy_peaked <- energy - entropy(peaked_density)
plot(free_energy_peaked,
     ylim = c(-3, 10),
     type = "l", lty = 2,
        ylab = "free energy",
        xlab = "Surprise about joint occurence")
lines(free_energy_flat)
```



If you start with lower entropy (peaked density, smooth line), the free energy is lower