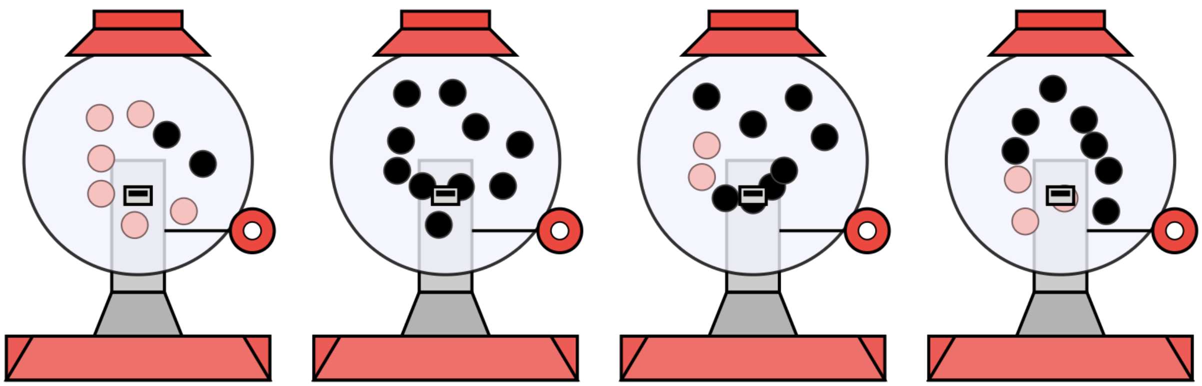
Computational Cognitive Science 2025-2026

Tutorial 5: Active learning

This tutorial provides some hands-on exposure to concepts from the active learning lectures.

Imagine you come across a curiosity shop that has four devices for sale at £100 each. Each device can automatically create a delicious cake when one pulls its lever, but the devices vary in their reliability:

- one device never works,
- one device works exactly 20 percent of the time,
- one works 30 percent of the time, and
- one works 75 percent of the time.



We don't know which device is which, but the shopkeeper will let us use the devices a few times to choose one.

Our task is to find out which machine is which.

Let's take an information-theoretic approach to this active learning problem.

Here are some helper functions.

```
plogp <- function(p) {
  # bits, not nats
  if(p==0) 0 else p*log2(p) # Consider why we can't just use p*log2(p).
}

# entropy for a categorical random variable
cat_entropy <- function(catv) {
  -sum(sapply(catv,plogp))
}
```

Let's define our hypothesis space as all of the different cake probabilities the machines before us might have from left to right. E.g., c(0,.2,.3,.75) means the machines increase in their probabilities from left to right, and c(.75,.3,.2,0) means they decrease from left to right.

Question 1: How does the code below define that hypothesis space?

```
require(combinat) # From googling "all list permutations in R"
```

```
## Loading required package: combinat

##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##     combn
```

```
probs <- c(0,.2,.3,.75)
hyps <- permn(probs)
```

Prior visualization:

```
library(ggplot2)
library(reshape2)

plot_hypothesis_distribution <- function(probs, title = "Probability Distribution over Hypotheses") {
  # Create hypothesis labels
  hyp_labels <- sapply(1:n_hyps, function(i) paste(hyps[[i]], collapse="-"))

  # Create data frame for plotting
  df <- data.frame(
    Hypothesis = factor(hyp_labels, levels = hyp_labels),
    Probability = probs
  )

  # Create and return the plot
  ggplot(df, aes(x = Hypothesis, y = Probability)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
    labs(title = title,
         x = "Hypothesis (Machine probabilities left to right)",
         y = "Probability") +
    ylim(0, 1)
}
```

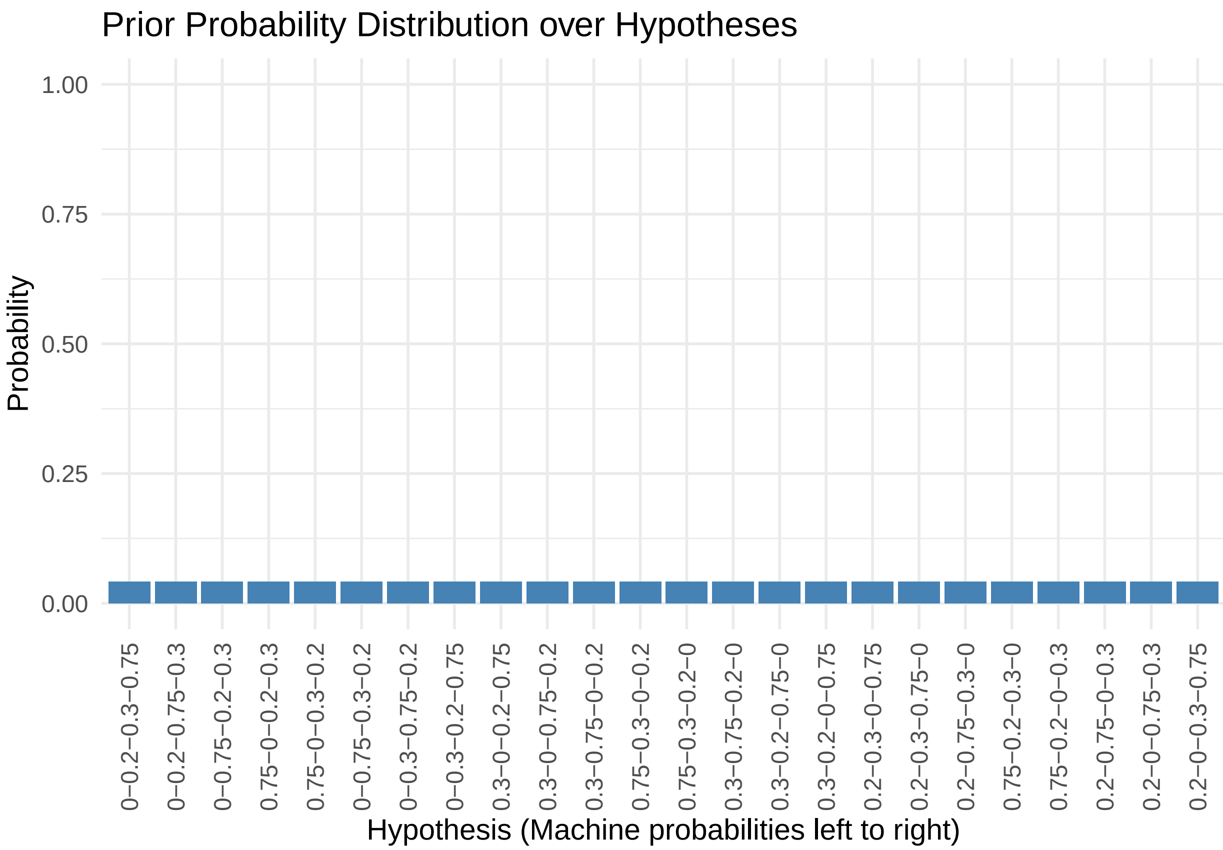
Question 2: What is our entropy, assuming we know nothing about the order of the machines?

$$Entropy = \sum_{h' \in \mathcal{H}} p(h') \log p(h')$$

```
n_hyps <- length(hyps)
prior <- rep(1/n_hyps,n_hyps) # even prior
cat_entropy(prior)
```

```
## [1] 4.584963
```

```
# Example usage with prior
plot_hypothesis_distribution(prior, "Prior Probability Distribution over Hypotheses")
```



Question 3: Suppose we turn the lever on the first machine and get nothing. What probability should we now assign our different hypotheses? Edit the code below.

Posterior:

$$p(h|d) = \frac{p(d|h)p(h)}{\sum_{h'} p(d|h')p(h')}$$

Likelihood $p(d|h)$:

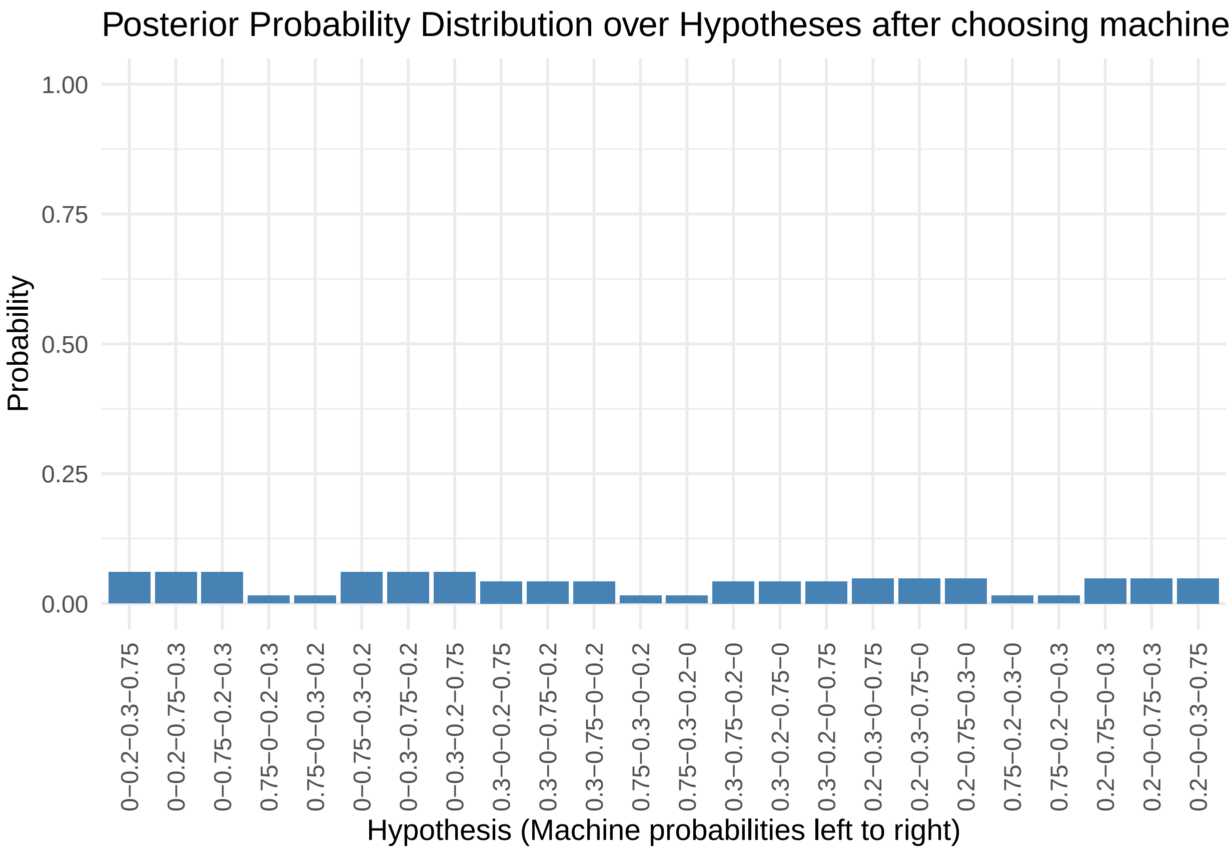
if cake: probability of the choice given the current hypothesis

if no cake: 1- probability of the choice given the current hypothesis

```
update_probs <- function(prior,choice,outcome) {
  # prior (vector): one prior probability per hypothesis
  # choice (int): index of the machine (1-4) on which the lever was turned
  # outcome (int): whether we get cake (1) or not (0)
  likelihood <- function(this_hyp,this_choice,this_outcome) {
    p <- this_hyp[[this_choice]]
    if (this_outcome == 1) p else 1-p
  }
  likes <- sapply(hyps,function(h) {likelihood(h,choice,outcome)})  # likelihoods
  unnp <- likes*prior # unnormalized posterior
  z <- sum(unnp) # normalization constant
  unnp/z
}

post <- update_probs(prior,choice=1,outcome=0)
```

```r
post_plot <- plot_hypothesis_distribution(post, "Posterior Probability Distribution over Hypotheses afte
post_plot
```



Question 4: What is our new entropy? Did we learn much? Why or why not?

```r
print('Posterior entropy:')
```

```
## [1] "Posterior entropy:"
```

```r
print(cat_entropy(post))
```

```
## [1] 4.450846
```

```r
 print('Prior entropy:')
```

```
## [1] "Prior entropy:"
```

```r
 print(cat_entropy(prior))
```

```
## [1] 4.584963
```

```r
print(cat_entropy(prior) - cat_entropy(post))
```

```
## [1] 0.1341161
```

Question 5: What is the expected probability getting a cake if we choose machine 2 next?

```r
# First we want the probability of each outcome in each hyp
# Then we take the weighted sum over hypothesis probabilities (Bayesian hypothesis averaging)
expected_outcome <- function(i,ph) {
  # ph (vector): posterior, one prior probability per hypothesis
```

```r
  # i (int): machine index
  sum(sapply(hyps,function(x) {x[i]})*ph)
}
print(expected_outcome(2,post))
```

```
## [1] 0.3490909
```

Question 6: What is the expected entropy if we choose machine 2 next?

```r
expected_entropy <- function(choice,p_hyps) {
  # First determine the probability of winning given your current beliefs
  pwin <- expected_outcome(choice,p_hyps)

  # Then determine the posterior if you do make that choice
  win_post <- update_probs(p_hyps,choice,1)
  lose_post <- update_probs(p_hyps,choice,0)

  # Return the entropy of the posteriors weighted by how likely we expected each of the outcomes
  cat_entropy(win_post)*pwin + cat_entropy(lose_post)*(1-pwin)
}

print(expected_entropy(2,post))
```

```
## [1] 4.15306
```

Question 7: Suppose we do choose machine 2 and it yields a cake. What is our current entropy and what are the expected entropies after each of our next possible choices?

```r
sapply(c(1,2,3,4),function(x) {expected_entropy(x,post)})
```

```
## [1] 4.231594 4.153060 4.153060 4.153060
```

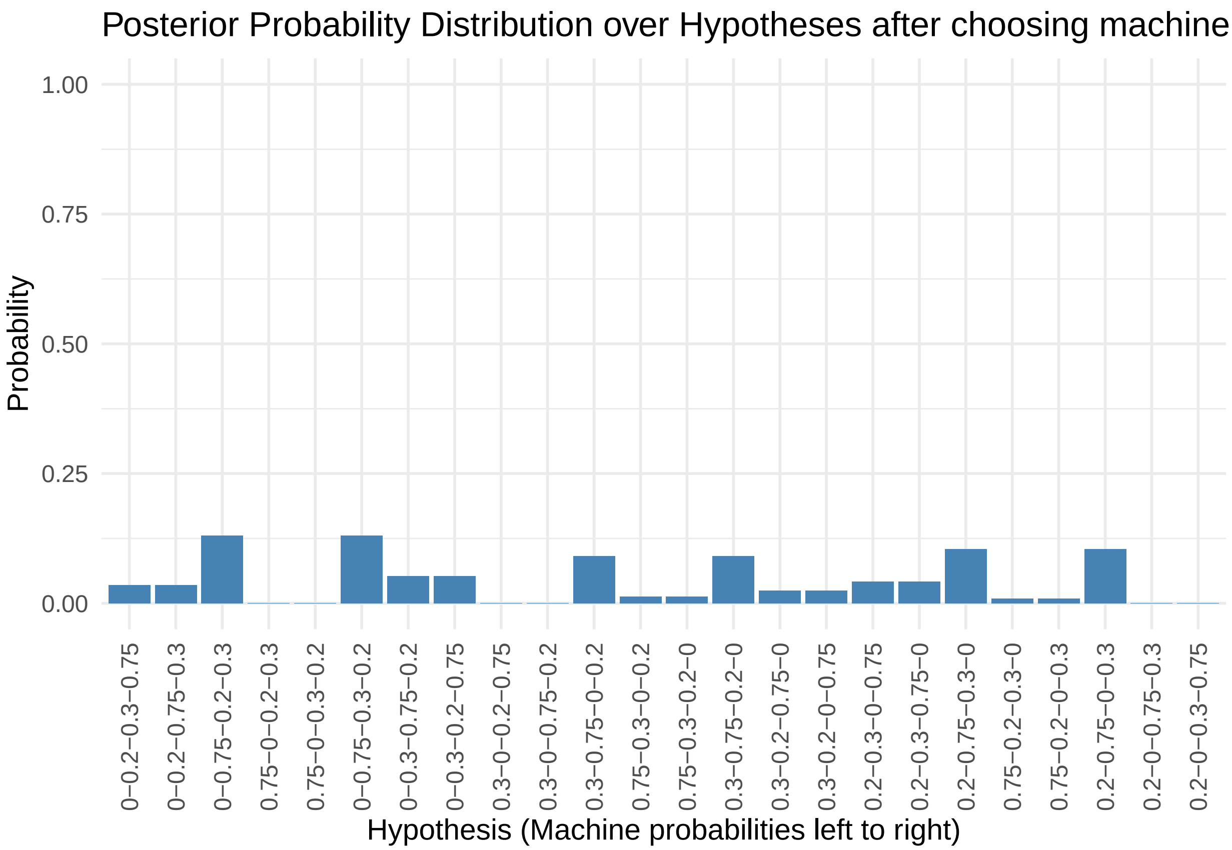
```r
post2 <- update_probs(post,2,1) # Get the new posterior with new data
entropy2 <- print(cat_entropy(post2))
```

```
## [1] 3.781135
```

```r
sapply(c(1,2,3,4),function(x) {expected_entropy(x,post2)})
```

```
## [1] 3.595719 3.613545 3.537257 3.537257
```

```r
plot_hypothesis_distribution(post2, "Posterior Probability Distribution over Hypotheses after choosing 
```



Question 8: What does this imply we should do next?

If we want to choose an intervention with maximum information gain, then we should choose the option that lowers our entropy the most.

Question 9: Is this at all counter-intuitive? What might we do differently if we are only interested in knowing which is the most reliable machine?

```r
# Calculate entropy over which machine has probability 0.75
is_75_ent <- function(probs) {
  # For each machine position (1-4), calculate probability it's the 0.75 machine
  prob_machine_is_75 <- numeric(4)

  for (machine_idx in 1:4) {
    # Sum probabilities of all hypotheses where this machine has p=0.75
    prob_machine_is_75[machine_idx] <- sum(
      probs[sapply(hyps, function(hyp) hyp[machine_idx] == 0.75)]
    )
  }

  # Return entropy over which machine is the 0.75 one
  cat_entropy(prob_machine_is_75)
}


ee_75 <- function(choice,prior) {
  pwin <- expected_outcome(choice,prior)
  if_win <- update_probs(prior,choice,1)
```

```r
  if_lose <- update_probs(prior,choice,0)
  is_75_ent(if_win)*pwin+is_75_ent(if_lose)*(1-pwin)
}

ents_75 <- sapply(c(1,2,3,4),function(x) {ee_75(x,post2)})
ents_75
```

```
## [1] 1.367541 1.263344 1.280055 1.280055
```