Computational Cognitive Science 2026/27

Week 6

Frank Keller Chris Lucas

Tutorial 4: Bayesian Estimation

Nisbett and Wilson: Are people biased?

In lecture 6, we discussed the study of Nisbett and Wilson (1977) in which people chose between four identical consumer products arranged from left to right. We used this example to introduce the multinomial likelihood and the Dirichlet prior. We will use the same data here, first for Bayesian parameter estimation and then for model comparison.

Options A–D are identical products arranged from left to right. The observations are the number of participants who chose each item.

Choice	A	B	C	D
Observations	6	9	16	21

```
choice_data <- c(A=6, B=9, C=16, D=21)
N <- sum(choice_data)
```

Bayesian estimation

Suppose we use a symmetric Dirichlet prior with $\alpha_i = 2$:

$$P(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \text{Dirichlet}(\boldsymbol{\theta}|\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = (2, 2, 2, 2)$$

Recall that the Dirichlet prior is conjugate to the multinomial likelihood. The posterior is therefore also Dirichlet, with the observed counts added to the prior parameters.

Question 1 What are the parameters of the posterior Dirichlet distribution for these data?

Write R code that creates the prior parameter vector $\boldsymbol{\alpha} = (2, 2, 2, 2)$ and computes the posterior parameters by adding the observed counts.

```
# Your code here
```

Question 2 Compute the MLEs of the four choice probabilities and the posterior predictive probabilities.

For a Dirichlet–multinomial model, the posterior predictive probability of choice i is

$$P(x_{N+1} = i \mid \mathbf{y}, \boldsymbol{\alpha}) = \frac{\alpha_i + y_i}{\sum_j (\alpha_j + y_j)}.$$

Write R code that computes both sets of probabilities and displays them in a two-row matrix or data frame. Then briefly explain how they differ and why.

```
# Your code here
```

Question 3 The posterior gives us more than a point estimate. Estimate the posterior probability that option D is chosen more than 25% of the time,

$$P(\theta_D > .25 \mid \mathbf{y}),$$

by drawing samples directly from the posterior.

The function below samples from a Dirichlet distribution by drawing independent Gamma random variables and normalizing them. You may use this function as given.

```
rdirichlet <- function(n, alpha) {
  z <- matrix(rgamma(n * length(alpha),
                     shape=rep(alpha, each=n)),
              nrow=n, byrow=FALSE)
  z / rowSums(z)
}
```

Now write R code that:

1. draws 100,000 samples from the posterior;
2. estimates $P(\theta_D > .25 \mid \mathbf{y})$;
3. computes the posterior mean and a 95% credible interval for θ_D.

```
# Your code here
```

Model comparison

We now compare two models:

- **M1 (uniform):** all four choices have probability 0.25.
- **M2 (multinomial):** the four choice probabilities are estimated from the data.

M1 is a special case of M2. M2 is more flexible, so it cannot fit the training data worse than M1. But, as discussed in lecture 8, better fit to the observed data does not necessarily imply better prediction.

Question 4 Compare the models on held-out data.

To make the idea concrete, expand the counts into individual observations and use 75% of them for training and 25% for testing.

```
choices <- rep(names(choice_data), choice_data)

set.seed(2026)
ix <- sample(seq_along(choices))
n_train <- floor(.75 * length(choices))

train <- choices[ix[1:n_train]]
test  <- choices[ix[(n_train+1):length(choices)]]

train_counts <- table(factor(train, levels=names(choice_data)))
test_counts  <- table(factor(test,  levels=names(choice_data)))

train_counts
```

```
##
##  A  B  C  D
##  4  6 10 19
```

```
test_counts
```

```
##
## A B C D
## 2 3 6 2
```

M1 assigns probability .25 to each choice. M2 should be fitted from the training counts.

Use the helper function below to compute predictive log likelihood:

```
loglik_test <- function(test_counts, p) {
  sum(test_counts * log(p))
}
```

Write R code that:

1. creates the probability vector for M1;
2. estimates the M2 probabilities from the training data;
3. computes the held-out predictive log likelihood for both models;
4. reports which model predicts the held-out data better.

```
# Your code here
```

Why do we not need to add an explicit parameter-count penalty when the test data are genuinely unseen?

References

Nisbett, Richard E., and Timothy D. Wilson. 1977. "Telling More Than We Can Know: Verbal Reports on Mental Processes." *Psychological Review* 84 (3): 231–59. https://doi.org/10.1037/0033-295X.84.3.231.