Tutorial07

The Dirichlet Distribution and the Dirichlet-Categorical Distribution

The Dirichlet distribution is a distribution over probability distributions; that is, draws from a Dirichlet are a probability distribution. In the lecture on overhypotheses, the Dirichlet distribution was parameterised with two values: Dirichlet($\alpha\boldsymbol{\beta}$), where $\alpha > 0$ is a scalar and $\boldsymbol{\beta}$ is a probability distribution of size K, where all $\beta_k > 0$ and $\sum_k \beta_k = 1$.

The two parameters play different roles: $\boldsymbol{\beta}$ is the **base distribution** and α is the **concentration parameter**, governing how much draws from Dirichlet($\alpha\boldsymbol{\beta}$) will diverge from the base distribution.

This exercise is meant to strengthen your intuitions about the role of the α and $\boldsymbol{\beta}$ hyperparameters in the context of Dirichlet priors with categorical likelihoods, by examining how the hyperparameters influence the prediction of the next draw.

Dirichlet priors with categorical likelihoods have a convenient closed form for the predictive posterior, integrating over all possible draws from the Dirichlet (θ in the lectures). With the Dirichlet($\alpha\boldsymbol{\beta}$) parameterisation, the predictive posterior is:

$$p(y = k|D, \alpha, \boldsymbol{\beta}) = \frac{\alpha\beta_k + N_k}{\sum_{k'} \alpha\beta_{k'} + N_{k'}}$$

where N_k refers to the number of items of category k in D.

Exercise: Consider a dataset consisting of ten marbles, two of which are black (marbles can only be black and white; so $K = 2; N = 10; N_{\textbf{black}} = 2$). Given this dataset, calculate the probability of the next marble being black, using the following hyperparameter settings (9 different settings; preferably vary α while keeping $\boldsymbol{\beta}$ stable):

- $\boldsymbol{\beta} = [0.5, 0.5], \boldsymbol{\beta} = [0.2, 0.8], \boldsymbol{\beta} = [0.8, 0.2]$
- $\alpha = 0.01, 1, 100$

You can calculate these by hand (preferably, at least to see what the numerator and denominator look like) or use R/Matlab or any language you prefer to calculate the probability.

Questions:

- What is the effect of a small α? A large α?
- The empirical likelihood of black is 0.2. When and why do the predictive posteriors match from the empirical likelihood?
- In this example, the posterior predictive probability always matches or increases the probability of black, as compared to the empirical likelihood. What kind of hyperparameters would result in the predictive probability of black being **lower** than the empirical likelihood?

Stan Tutorial

In this part of the tutorial you'll be exploring the marbles-in-bags model from Kemp, Perfors, and Tenenbaum (2007) using a package called Stan. You will need to have the Kemp paper in front of you as well (the course website has a link to this paper).

Stan is a useful library that allows you to define a Bayesian model using the kind of notation you saw in the lecture (and will see in the code). Given a model specification and data, it calculates the posterior distribution using a sampler. Note that unlike the word learning model, which calculated a MAP solution, here we will be dealing with the full posterior distribution. Stan performs inference by building and running a sampler generated for your model.

Install Stan by running `install.packages("rstan")` in your R-console - this might take a while and will display warnings (that you can ignore). You might also need to install the graphing library `ggplot2`.

0. Getting started with Stan

Open the model file in `model.stan` and read it. The stan syntax will be new to you, but you should be able to see how the stan model specification corresponds to the model defined in the Appendix of Kemp, Perfors, and Tenenbaum (2007) (the first model, for Figure 1a).

The model specification is made up of different program blocks:

- The *data* block, where the required input variables are declared. These include:
  - dimensionalities/sizes (B, M and V);
  - the fixed (hyper)parameter ω (omega)
  - the data, in `items`
- The *parameters* block, where the model parameters are declared; these are the parameters that we will be inferring, via sampling.
- The *model* block. Here the relationships between the parameters and the data are defined, by specifying a generative story for the data.
- The *generated quantities* block. We may want to look at certain quantities that are functions of the parameters and data, but not explicitly in the model. Here we're calculating the distribution over `theta_new`, the distribution of colors in a bag from which we've seen only one marble.

Notice that there are extra higher layers of parameters in the code (which actually matches the model we saw in lecture, not the paper).

Question: What is the difference between the code and the Kemp paper's model description?

1. Homogenous bags

Exercise: Look at Figure 3a) i and ii in the Kemp paper; read the caption and make sure you understand why the distributions look the way they do. Write down your estimate/prediction of the mean of each of the distributions (for alpha, beta, a white-bag theta and a black-bag theta), so you can compare this to the model at the next step.

We provide you the code to generate the marble bags in `data.R`. First we'll be running the model with the data from the single color per bag setting and look at the output that's produced.

```
data.single <- generate_data.onecolor(num_items=200, num_bags=10, num_values=2,
                                      omega_param=1)
table(data.single$item)
```

```
##
##   1   2
## 100 100
```

Now we run the model specified in `model.stan` by providing it the data set. This will take a while, as Stan runs 4 chains of MCMC for 2000 iterations.

```
#fitsingle <- stan(file = 'model.stan', data = data.single, verbose = FALSE,
#            iter = 2000, chains = 4 , control=list(adapt_delta=0.95))
```

The first thing to check is whether the model is working as expected; whether the inference procedure has converged to a good estimate of the posterior distribution. Since we ran 4 chains of MCMC, we can check how consistent these chains are - one common diagnostic is called the Gelman-Rubin diagnostic, or $\hat{R}$. Print out the full summary: `summary(fitsingle)` and check that the `Rhat` values are smaller than 1.1.

```
#s <- summary(fitsingle)
#s$summary #reports the merged chains
```

In addition to the `Rhat` values the output also gives you the mean posterior estimates and distributions over the parameters in the model and the generated quantities.

Questions:

- Do these means correspond to what you predicted from the paper?
- What about the shape/spread of the distribution? You can look at the percentiles, and/or use `stan_hist(fit,pars=c("alpha", "beta"), bins=50)` to examine the sampled values, which correspond to the posterior. (I.e. once it's converged, the sampler draws from the posterior, so you would expect many samples from high-probability areas and few from low-probability areas).
- Check `theta_new`, the prediction about the distribution of a new bag of which you've seen only one marble. What does it say about the probable color of the next marble?

2. Heterogeneous bags

Exercise: Do the same (go through all steps and questions from (1)) for the mixed bags scenario in Figure 3a(iii). Note that this means inputting different data – use `generate_data.mixed_proportion` in `data.R`.

```
#data.mixed <- Your code here
```

```
#fitmixed <- stan(file = 'model.stan', data = , verbose = FALSE,
#            iter = 2000, chains = 4)
```

Question: Why is the shape/spread of the distribution of theta_new so different between Fig 3a(ii) and 3a(iii)?

This concludes this weeks' tutorial. We will return to explore Kemp, Perfors, and Tenenbaum (2007) in next week's tutorial. The next sections contain the exercises for next week's tutorial – if you have time you can already have a look.

3. A model without overhypotheses

Exercise: Edit the model declaration in `model.stan`, or create a copy of that file so that you get the non-hierarchical, single level model in Figure 3b. Run this truncated model with both the single-color-bags and the mixed-color-bags settings. Compare `theta_new` under both data settings - how does this differ from the full model?

Hints:

- Variable assignments (+ simultaneous type declarations) can be made in the `model` block, but not the `parameters` block.
- Commenting out is done with `//` (comment out, don't delete, so you can restore things later!)
- For beta, you will get a syntax error if you try to declare it as a simplex (i.e. as a vector that has to sum to 1); use `vector` instead.
- `rep_vector(real, int)` is a function that creates a vector (size int) that consists of `[int]` copies of `[real]`.
- make sure your lines end with `;`

Once you are done restore the model to the original (full, hierarchical) structure.

4. Hyperparameter settings

Exercise: Play with changing the value of the hyperparameter (omega). Try three different values, fairly far apart — try different orders of magnitude. To do this, you will have to specify the `omega_param` when you call `generate_data`.

Questions:

- Does this change have an effect on the final posterior, for example on theta_new?
- Does it matter which data (mixed or homogeneous) you use?
- Does any change that you see make sense, given what you know about the distributions?

Hints:

- Draws from Exponentials with smaller parametrisations (i.e, smaller values of omega) will tend to be larger, since the mean/expected values is $E[X] = 1/\omega$ (see https://en.wikipedia.org/wiki/Exponential_distribution)
- For alpha, this end up cancelling out (you're drawing from exp(exp(omega)), with expected value $\frac{1}{1/\omega} = \omega$ so smaller omega correspond to (expected) smaller alphas;
- For beta: smaller values of omega will lead to larger values of mu, and $\beta \sim Dirichlet(\mu)$; remember that here μ is playing the role of 'concentration parameter' (so same as alpha on the layer below), while the 'base distribution' is uniform over all categories.
- The concentration parameter governs how concentrated draws from the Dirichlet will be to the base distribution. Small values will lead to greater spread, while large values lead to most draws staying close to the base distribution.

Note: The sampler may start giving you warnings about divergences; these are important indicators that the hyperparameter values you've given aren't ideal, since it makes it harder for the sampler to find the correct posterior distribution. However, you can ignore them for now, as long as the Rhat values look OK.

5. Learning at different levels

a) Inspect Figure 6 in Kemp, Perfors, and Tenenbaum (2007), which shows how the model can learn more confident, i.e., tighter, predictions at different levels, depending on the data it's exposed to. Again, first, look at the figure and read the figure caption; make sure you understand the reason for the shapes of the posteriors.

b) Now recreate the figures with your model, by giving it the different datasets (have a look at `data.R` for the generating functions).

c) What effect will the tighter distributions over alpha, beta in scenario b(iii) have on theta? Does this pattern of thetas in b(iii) depend on the values of alpha, beta, or just the certainty? (I.e. if the peaks of the posterior alpha and beta were elsewhere, but the shape was the same, would the pattern of theta look the same?)

Hint: You can use `stan_hist(fit,pars=c("alpha", "beta"), bins=50)` to look at the sample distribution of specific parameters.

References

Kemp, Charles, Amy Perfors, and Joshua Tenenbaum. 2007. "Learning Overhypotheses with Hierarchical Bayesian Models." *Developmental Science* 10. Wiley Online Library.