Advanced Statistical Inference
Bayesian inference for a coin tossing experiment

1 Aims

- To become familiar with statistical operations in matlab.
- To do some simple Bayesian things.

2 Coin tossing

2.1 Computing posteriors

The beta probability density function is defined as:

$$p(r|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} r^{\alpha-1}(1-r)^{\beta-1}$$

where $0 \leq r \leq 1$ and $\alpha, \beta > 0$.

Download `plotbeta.m` from the course webpage. This functions takes α and β as arguments and plots the corresponding beta pdf. You can also pass a figure number as a third argument to plot onto that particular figure. For example, try the following two snippets of code:

```
plotbeta(1,2);
plotbeta(2,2);
```

and

```
f = plotbeta(1,2);
hold on
plotbeta(2,2,f);
```

1. r is the probability of some hypothetical coin landing heads. The beta distribution is a suitable prior distribution for r. Choose 2 different pairs of values α and β and plot the corresponding densities. For both cases, describe (to one of the demonstrators) the possible beliefs they encode about the coin. Note, there is no 'right' answer here, just think about how likely different values of r are in your densities and how this reflects on the coin.

2. `coin1.m` and `coin2.m` are two 'coins' that you can download from the project webpage. In either case, calling the function with an argument N, tosses the coin N times and returns the number of heads. Pick either one and, using the equations we derived in the class, along with any choice for α and β, plot the posterior density for r after $N = 1$, $N = 5$, $N = 10$, $N = 100$ tosses. Use `plotbeta` to do the plotting. You can put them all on one plot or all on separate plots. Explain what you see to one of the demonstrators.

3. Repeat the previous exercise with a different set of prior parameter values (α and β). What do you notice?

2.2 Making predictions

Download `fairground.m`. This is a coin game – you pay £1 to play. The stall owner tosses a coin ten times. If there are 6 or fewer heads, you win £2. The function returns 1 if you win and 0 if you lose.

4. The probability that you win could be approximated by playing the game lots of times and computing the proportion of times you do win! By embedding `fairground.m` in a loop, approximate the probability that you will win in this way.

5. In the lectures, we derived an expression for the probability of getting y_{new} heads in N tosses having observed y_N heads in N tosses by taking an expectation with respect to the posterior. Choose α and β as you please (as long as you can justify your choice – do you know anything about this coin?). Use `coin2.m` to generate y_N for $N = 10$. Compute the posterior parameters. By using them, the expression from the lectures and the following identity, compute the probability that you will win given the coin tosses you have observed (i.e., you just need to add together a series of the expressions you derived in the lecture). Note that this quantity depends on the prior and the particular y_N you get from `coin2.m` and therefore could be quite different from the quantity computed in the previous question.

$$P(y_{\mathsf{new}} \leq 6|y_N,\alpha,\beta) = \sum_{a=0}^{6} P(y_{\mathsf{new}} = a|y_N,\alpha,\beta)$$

or

$$P(y_{\mathsf{new}} \leq 6|y_N,\alpha,\beta) = 1 - \sum_{a=7}^{10} P(y_{\mathsf{new}} = a|y_N,\alpha,\beta)$$

Note that you can use the function `nchoosek` to evaluate $\begin{pmatrix} N \\ y_{\mathsf{new}} \end{pmatrix}$.

6. Vary your prior and observe the effect it has on this probability.