Lecture 4: Binomial models and prior strength

Zachary Marion

1/30/2017

As before, we need to load some packages and set some options prior to running any models:

```
library(rstan)
library(shinystan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

Exploring binomial models

Bernoulli model recap

Recall from the past few lectures the motivating question has been to estimate the probability θ of encountering water relative to land on a small model of the Earth (i.e., an inflatable globe).

- Can also interpet θ as the relative proportion of water covering the Earth (or blue covering our model).

As before, our collected data was: W W W L W, and as before we will create a Stan model that accounts for the Bernoulli nature of the data with a beta prior.

```
data {
  int<lower=0> nObs;                // Total number of obss
  int<lower=0, upper=1> obs[nObs];  // 1D array of obs
  real<lower=0> a;                  // a & b are now input as data
  real<lower=0> b;                  //  rather than hard-coded
}

parameters {
  real<lower=0, upper=1> theta;     // prob. of water
}

model {
  theta ~ beta(a,b);                // prior on theta
  for(n in 1:nObs) {
    obs[n] ~ bernoulli(theta);      // bernoulli likelihood
```

```
  }
}
```

Note that I have changed the code slightly so that `a` & `b`—the parameters defining the beta prior—are data rather than hardcoded. This makes the model more flexible down the line.

We again need to code the data as a list and then run the model using the `stan` function. Similar to last class, `a` & `b` define a flat beta prior:

```
nObs <- length(obs)
obs <- rep(c(1,0), times=c(4,1))
a <- 1.0
b <- 1.0 # giving a & b a flat prior
dat1 <- list(nObs=nObs, obs=obs, a=a, b=b)
```

```
modBern <- stan(file="ex1Bernoulli.stan", #path to .stan file
             data=dat1,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes)
```

Bernoulli model as a binomial

The Bernoulli is a special case of the binomial distribution, the likelihood function of which—for the pedantic—is:

$$p(y|N,\theta) = \frac{N!}{y!(N-y)!}\theta^y(1-\theta)^{N-y} \tag{1}$$

where $y = \sum(W)$.

We can easily streamline and recode the model using a *Binomial* likelihood for more flexibility later:

```
data {
  int<lower=0> nObs;            // Total number of observations
  int<lower=0> obs;             // scalar count of Ws
  real<lower=0> a;              // a & b are now input as data
  real<lower=0> b;              //  rather than hard-coded
}

parameters {
  real<lower=0, upper=1> theta; // prob. of water
}
```

```
model {
  theta ~ beta(a,b);             // prior on theta
  obs ~ binomial(nObs, theta);  // binomial likelihood function
}
```

We can recode the data and run the new model:

```
nObs <- length(obs)
obs <- sum(obs)
a <- 1
b <- 1
dat2 <- list(nObs = nObs, obs = obs, a = a, b = b)
```

```
modBin1 <- stan(file="ex1Binomial.stan", #path to .stan file
             data=dat2,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes)
```

The estimates from both the Bernoulli and Binomial models are identical (minus MCMC variation):

```
print(modBern)
```

```
       mean se_mean   sd  2.5%   25%   50%   75% 97.5%   n_eff Rhat
theta  0.71    0.00 0.16  0.37  0.61  0.74  0.84  0.96 2319.51    1
lp__  -4.76    0.02 0.76 -6.95 -4.96 -4.47 -4.25 -4.19 2372.47    1
```

```
print(modBin1)
```

```
       mean se_mean   sd  2.5%  25%   50%   75% 97.5%   n_eff Rhat
theta  0.71    0.00 0.16  0.36  0.6  0.73  0.83  0.95 2201.69    1
lp__  -4.72    0.01 0.72 -6.71 -4.9 -4.45 -4.25 -4.19 2555.58    1
```

Alternative parameterizations and prior strength

As mentioned in an earlier lecture, it is often convienient to specify a and b of a beta prior in terms of the concentration (κ) and mean (μ):

$$a = \mu\kappa \text{ and } b = (1-\mu)\kappa. \tag{2}$$

or—for $\kappa > 2$—mode (ω):

$$a = \omega(\kappa-2)+1 \text{ and } b = (1-\omega)(\kappa-2)+1 \tag{3}$$

We can easily do this by using the `transformed parameters` block of Stan. I will use the mode but the strategy for using the mean is the same.

We will read the mode and concentration in as `data`. The `parameters` and `model` blocks doesn't change. After specifying the transformed parameters (`a` & `b`), we write the equations as noted above.

```
data {
  int<lower=0> nObs;    // Total number of observations
  int<lower=0> obs;     // obs as scalar
  real<lower=0, upper=1> omega;  // mode as input data
  real<lower=2> kappa;  // concentration
}

...

transformed parameters {
  real<lower=0> a;
  real<lower=0> b;

  a = omega * (kappa - 2) + 1;
  b = (1 - omega) * (kappa - 2) + 1;
}
```

Suppose we think there is a 50/50 probability of water. We can see the effect of varying κ, our confidence in this assumption b y running two models:

- Completely uninformed: $\omega = 0.5$, $\kappa = 2$
- Low confidence: $\omega = 0.5$, $\kappa = 4$
- High confidence: $\omega = 0.5$, $\kappa = 12$

To determine the influence of the prior, One option (for simple models), to exclude the likelihood and look at the posterior distribution of our parameters. We can do this by commenting out the last part of our model, and comparing the parameter estimates with and without the "data."

```
...

model {
  theta ~ beta(a,b);            // prior on theta
 // obs ~ binomial(nObs, theta);  // Exclude likelihood
}
```

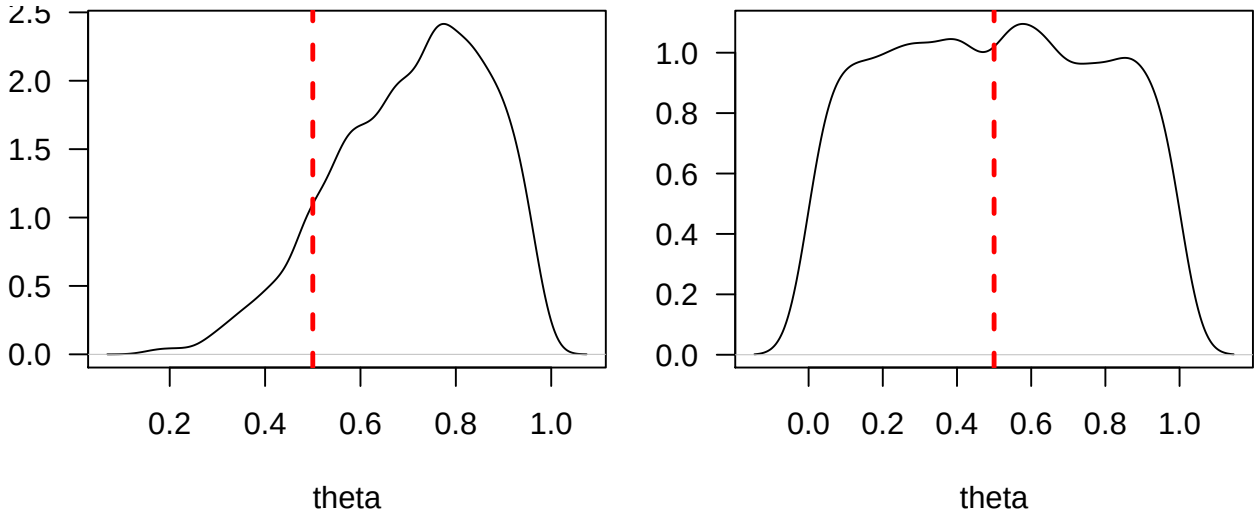
Completely uninformed: $\omega = 0.5$, $\kappa = 2$:

```
## Essentially Beta(1,1)
omega <- 0.5
kappa <- 2
dat11 <- list(nObs = nObs, obs = obs, omega = omega, kappa = kappa)
```

```
## Beta(1,1) parameterized by mode
modBin11 <- stan(file="binomialMode.stan", #path to .stan file
             data=dat11,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```
## No likelihood Beta(1,1)
modNoLik11 <- stan(file="noLikBinomial.stan", #path to .stan file
             data=dat11,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```
theta11 <- as.matrix(modBin11, par = "theta")
thetaNoLik11 <- as.matrix(modNoLik11, par = "theta")
par(mfrow = c(1, 2))
par(mar = c(4, 3, 0.1, 0.5))
plot(density(theta11), xlab = "theta", las = 1, main = "", ylab = "Density")
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
plot(density(thetaNoLik11), xlab = "theta", las = 1, main = "", ylab = "")
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
```



Vaguely informed: $\omega = 0.5$, $\kappa = 4$**:**

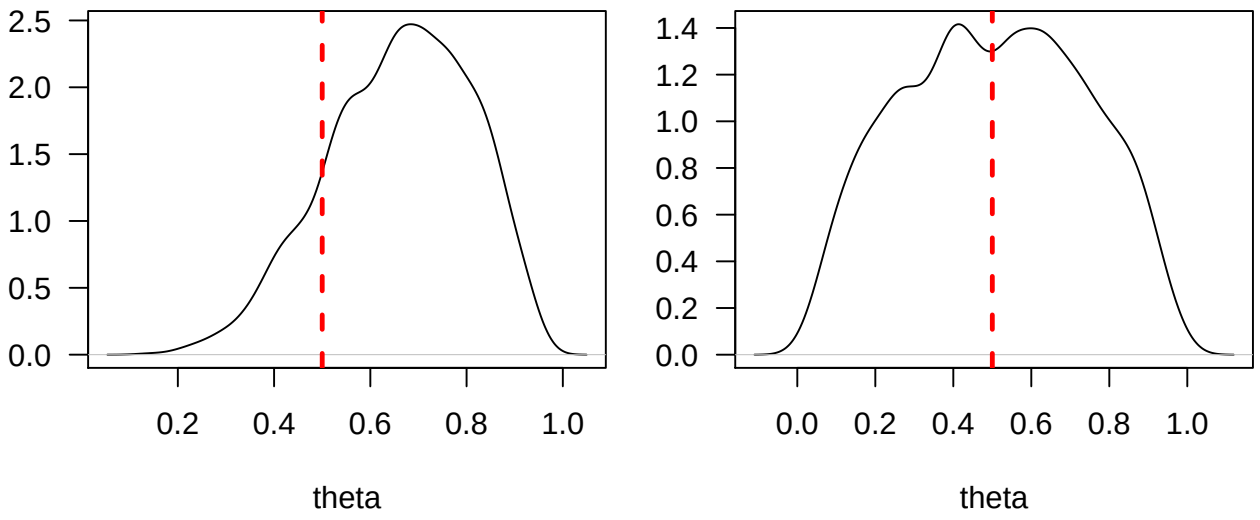
```
## Essentially Beta(2,2)
omega <- 0.5
kappa <- 4
dat22 <- list(nObs = nObs, obs = obs, omega = omega, kappa = kappa)
```

```
## Beta(2,2) parameterized by mode
modBin22 <- stan(file="binomialMode.stan", #path to .stan file
             data=dat22,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```
## No likelihood Beta(1,1)
modNoLik22 <- stan(file="noLikBinomial.stan", #path to .stan file
             data=dat22,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```
theta22 <- as.matrix(modBin22, par = "theta")
thetaNoLik22 <- as.matrix(modNoLik22, par = "theta")
par(mfrow = c(1, 2))
par(mar = c(4, 3, 0.1, 0.5))
plot(density(theta22), xlab = "theta", las = 1, main = "", ylab = "Density")
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
plot(density(thetaNoLik22), xlab = "theta", las = 1, main = "", ylab = "")
```

```r
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
```



Strongly informed: $\omega = 0.5$, $\kappa = 20$:

```r
## Essentially Beta(10,10)
omega <- 0.5
kappa <- 20
dat1010 <- list(nObs = nObs, obs = obs, omega = omega, kappa = kappa)
```

```r
## Beta(2,2) parameterized by mode
modBin1010 <- stan(file="binomialMode.stan", #path to .stan file
             data=dat1010,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```r
## No likelihood Beta(1,1)
modNoLik1010 <- stan(file="noLikBinomial.stan", #path to .stan file
             data=dat1010,
             iter=2000, # number of MCMC iterations
             chains=4,  # number of independent MCMC chains
             seed=3,    # set the seed so run is repeatable
             verbose=FALSE)  # turn off annoying warnings for notes
```

```r
theta1010 <- as.matrix(modBin1010, par = "theta")
thetaNoLik1010 <- as.matrix(modNoLik1010, par = "theta")
par(mfrow = c(1, 2))
```

```r
par(mar = c(4, 3, 0.1, 0.5))
plot(density(theta1010), xlab = "theta", las = 1, main = "", ylab = "Density")
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
plot(density(thetaNoLik1010), xlab = "theta", las = 1, main = "", ylab = "")
abline(v = omega, lty = 2, lwd = 2.5, col = "red")
```