Bayesian inference

Statistical Inference

Brian Caffo, Roger Peng, Jeff Leek
Johns Hopkins Bloomberg School of Public Health

Bayesian analysis

- Bayesian statistics posits a prior on the parameter of interest

prior encapsulates our prior knowledge or assumptions

- All inferences are then performed on the distribution of the parameter given the data, called the posterior
- In general,

dieses alpha bedeutet 'proportional'

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

- Therefore (as we saw in diagnostic testing) the likelihood is the factor by which our prior beliefs are updated to produce conclusions in the light of the data

Prior specification

- The beta distribution is the default prior for parameters between 0 and 1.
- The beta density depends on two parameters α and β

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\, p^{\alpha-1}\,(1-p)^{\beta-1} \quad \text{for } \; 0 \leq p \leq 1$$

- The mean of the beta density is $\alpha/(\alpha+\beta)$
- The variance of the beta density is

$$\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

- The uniform density is the special case where $\alpha=\beta=1$

```r
## Exploring the beta density
library(manipulate)
pvals <- seq(0.01, 0.99, length = 1000)
manipulate(
    plot(pvals, dbeta(pvals, alpha, beta), type = "l", lwd = 3, frame = FALSE),
    alpha = slider(0.01, 10, initial = 1, step = .5),
    beta = slider(0.01, 10, initial = 1, step = .5)
    )
```

Posterior

- Suppose that we chose values of α and β so that the beta prior is indicative of our degree of belief regarding p in the absence of data
- Then using the rule that

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

and throwing out anything that doesn't depend on p, we have that

$$\begin{aligned} \text{Posterior} &\propto \overbrace{p^x (1-p)^{n-x}}^{\texttt{likelihood part}} \times \overbrace{p^{\alpha-1}(1-p)^{\beta-1}}^{\texttt{prior part}} \\ &= p^{x+\alpha-1}(1-p)^{n-x+\beta-1} \end{aligned}$$

- This density is just another beta density with parameters $\tilde{\alpha} = x + \alpha$ and $\tilde{\beta} = n - x + \beta$

```
alpha: a priori num of successes     alpha~: a posteriori num of successes
beta:  a priori num of failures      beta~:  a posteriori num of failures
```

Posterior mean

```
prior mean, without the data:

E_prior[p] = alpha/(alpha+beta)
```

$$
\begin{aligned}
E[p \mid X] &= \frac{\tilde{\alpha}}{\tilde{\alpha} + \tilde{\beta}} \\
&= \frac{x + \alpha}{x + \alpha + n - x + \beta} \\
&= \frac{x + \alpha}{n + \alpha + \beta} \\
&= \frac{x}{n} \times \frac{n}{n + \alpha + \beta} + \frac{\alpha}{\alpha + \beta} \times \frac{\alpha + \beta}{n + \alpha + \beta} \\
&= \text{MLE} \times \pi + \text{Prior Mean} \times (1 - \pi)
\end{aligned}
$$

```
pi is just 'some proportion'

Gewichtung des MLE (entspr. Daten) vs Prior.
Wenn zB n sehr gross ist, wird der erste Term fast 1
und der andere fast 0. Und umgekehrt.
```

Thoughts

- The posterior mean is a mixture of the MLE ($\hat{p}$) and the prior mean
- π goes to 1 as n gets large; for large n the data swamps the prior
- For small n, the prior mean dominates
- Generalizes how science should ideally work; as data becomes increasingly available, prior beliefs should matter less and less
- With a prior that is degenerate at a value, no amount of data can overcome the prior

Example

- Suppose that in a random sample of an at-risk population 13 of 20 subjects had hypertension. Estimate the prevalence of hypertension in this population.
- $x = 13$ and $n = 20$
- Consider a uniform prior, $\alpha = \beta = 1$
- The posterior is proportional to (see formula above)

$$p^{x+\alpha-1}(1-p)^{n-x+\beta-1} = p^x(1-p)^{n-x}$$

  That is, for the uniform prior, the posterior is the likelihood

- Consider the instance where $\alpha = \beta = 2$ (recall this prior is humped around the point $.5$) the posterior is

$$p^{x+\alpha-1}(1-p)^{n-x+\beta-1} = p^{x+1}(1-p)^{n-x+1}$$

- The "Jeffrey's prior" which has some theoretical benefits puts $\alpha = \beta = .5$

```r
pvals <- seq(0.01, 0.99, length = 1000)
x <- 13; n <- 20
myPlot <- function(alpha, beta){
    plot(0 : 1, 0 : 1, type = "n", xlab = "p", ylab = "", frame = FALSE)
    lines(pvals, dbeta(pvals, alpha, beta) / max(dbeta(pvals, alpha, beta)),
            lwd = 3, col = "darkred")
    lines(pvals, dbinom(x,n,pvals) / dbinom(x,n,x/n), lwd = 3, col = "darkblue")
    lines(pvals, dbeta(pvals, alpha+x, beta+(n-x)) / max(dbeta(pvals, alpha+x, beta+(n-x))),
        lwd = 3, col = "darkgreen")
    title("red=prior,green=posterior,blue=likelihood")
}
manipulate(
    myPlot(alpha, beta),
    alpha = slider(0.01, 10, initial = 1, step = .5),
    beta = slider(0.01, 10, initial = 1, step = .5)
    )
```

Credible intervals

- A Bayesian credible interval is the Bayesian analog of a confidence interval
- A 95% credible interval, $[a, b]$ would satisfy

$$P(p \in [a,b] \mid x) = .95$$

- The best credible intervals chop off the posterior with a horizontal line in the same way we did for likelihoods
- These are called highest posterior density (HPD) intervals

Getting HPD intervals for this example

- Install the \texttt{binom} package, then the command

```
library(binom)
binom.bayes(13, 20, type = "highest")
```

```
  method  x  n shape1 shape2   mean  lower  upper  sig
1  bayes 13 20   13.5    7.5 0.6429 0.4423 0.8361 0.05
```

gives the HPD interval.

- The default credible level is 95% and the default prior is the Jeffrey's prior.

```r
pvals <- seq(0.01, 0.99, length = 1000)
x <- 13; n <- 20
myPlot2 <- function(alpha, beta, cl){
    plot(pvals, dbeta(pvals, alpha+x, beta+(n-x)), type = "l", lwd = 3,
    xlab = "p", ylab = "", frame = FALSE)
    out <- binom.bayes(x, n, type = "highest",
        prior.shape1 = alpha,
        prior.shape2 = beta,
        conf.level = cl)
    p1 <- out$lower; p2 <- out$upper
    lines(c(p1, p1, p2, p2), c(0, dbeta(c(p1, p2), alpha+x, beta+(n-x)), 0),
        type = "l", lwd = 3, col = "darkred")
}
manipulate(
    myPlot2(alpha, beta, cl),
    alpha = slider(0.01, 10, initial = 1, step = .5),
    beta = slider(0.01, 10, initial = 1, step = .5),
    cl = slider(0.01, 0.99, initial = 0.95, step = .01)
    )
```