ADVANCED BAYESIAN MODELING

Bread and Peace Example: Posterior Inference

Recall Bread and Peace model data for U.S. presidential elections (1952 to 2004)

$$
\begin{aligned}
y_i &= \text{incumbent two-party vote \% in election } i \\
x_i &= \text{weighted-average per capita real income \% growth (previous term)}
\end{aligned}
$$

Consider simple linear regression

$$
y_i \mid \theta, X \;\sim\; \text{indep. N}(\beta_1 + \beta_2 x_i,\, \sigma^2) \qquad\qquad i = 1, \ldots, 14
$$

```
> bp <- read.table("breadandpeace.txt", header=TRUE)

> bp
   Election IncumbentPct IncomeGrowth
1      1952         44.6          2.4
2      1956         57.8          2.9
3      1960         49.9          0.8
4      1964         61.3          4.2
5      1968         49.6          3.0
6      1972         61.8          3.6
7      1976         49.0          1.1
8      1980         44.7         -0.4
9      1984         59.2          3.9
10     1988         53.9          2.3
11     1992         46.5          0.4
12     1996         54.7          1.0
13     2000         50.3          2.4
14     2004         51.2          1.9
```

Classical linear regression in R is performed with function `lm`:

```
> mod <- lm(IncumbentPct ~ IncomeGrowth, data=bp)
```

The first argument is a *formula*, which defines y and X.

```
> ( X <- model.matrix(mod) )
   (Intercept) IncomeGrowth
1            1          2.4
2            1          2.9
3            1          0.8
4            1          4.2
5            1          3.0
6            1          3.6
7            1          1.1
8            1         -0.4
9            1          3.9
10           1          2.3
11           1          0.4
12           1          1.0
13           1          2.4
14           1          1.9
attr(,"assign")
[1] 0 1
```

Classical inference is available using `summary` and other R functions, but we will just extract what we need: $\hat{\beta}$, s^2, V_β, and $n-k$.

```
> betahat <- coef(mod)

> smod <- summary(mod)

> s.2 <- smod$sigma^2

> Vbeta <- smod$cov.unscaled

> n.minus.k <- df.residual(mod)
```

Posterior Simulation

We simulate directly from the posterior, with the help of a function for simulating multivariate normal variates, from package `MASS`:

```
> library(MASS)  # provides mvrnorm

> Nsim <- 1000

> post.sigma.2.sim <- n.minus.k * s.2 / rchisq(Nsim, n.minus.k)

> post.beta.sim <- matrix(NA, Nsim, length(betahat))
> for(s in 1:Nsim)
+   post.beta.sim[s,] <- mvrnorm(1, betahat, post.sigma.2.sim[s] * Vbeta)
```

Can approximate posterior intervals and probabilities as usual:

```
> quantile(post.sigma.2.sim, c(0.025,0.975))  # posterior interval
    2.5%    97.5%
 8.29754 42.93682

> apply(post.beta.sim, 2, quantile, c(0.025,0.975))  # posterior intervals
          [,1]     [,2]
2.5%  41.56126 1.520744
97.5% 49.74621 4.897986

> confint(mod)  # exact classical intervals, for comparison
                2.5 %    97.5 %
(Intercept)  41.450056 50.178157
IncomeGrowth  1.403473  4.908561

> mean(post.beta.sim[,2] > 0)  # posterior prob. slope is positive
[1] 0.999
```

A note on R function `apply`:

```
apply(mat, 2, fun, ...)
```

applies function `fun` to each <u>column</u> of matrix `mat`, returning the results as a vector or matrix (if needed).

(`...` allows additional arguments to `fun`.)

```
apply(mat, 1, fun, ...)
```

does the same for each <u>row</u> of `mat` instead.

Prediction

We wish to predict the incumbent (Republican) vote % for the 2008 election.

It was known (prior to the election) that the real income growth in the previous term was 0.75%.

We start by simulating from the posterior predictive distribution:

```
> post.pred.y.sim <- rnorm(Nsim, post.beta.sim[,1] + post.beta.sim[,2] * 0.75,
+                            sqrt(post.sigma.2.sim))
```

Approximate the posterior predictive interval as usual, and compare with the classical prediction interval:

```
> quantile(post.pred.y.sim, c(0.025,0.975))  # posterior predictive interval
    2.5%    97.5%
38.89235 56.79167

> predict(mod, data.frame(IncomeGrowth=0.75), interval="prediction")
       fit      lwr      upr
1 48.18112 38.86593 57.49631
>                                             # exact 95% classical interval
```

For reference, the actual Republican two-party vote share for 2008 was 46.3%.

Can also approximate the posterior predictive probability of a 2008 incumbent (Republican) victory:

```
> mean(post.pred.y.sim > 50)  # post. pred. prob. of incumbent victory
[1] 0.342
```