Talking out our posteriors, or how to do Bayesian regression

April 10, 2018

Overview of the Learning Session

- Introduction to Bayesianism
- Simple computational Bayes
- Bayesian regression

Discussions to have

- Forgeting event spaces for random variables.
- Adult form of the Bayes Rule.
- Bayesian learning, vs. Bayesian statistics.
- The theory and practice of Markov Chain Monte Carlos.
- More intuitive ways to turn Bayes's rule to a posterior.

Want to show how to run regressions with non-parametric parameters.

Theoretical setup

Example: Let

$$y_i = \sum_i \beta_k x_{ik} + \epsilon_i$$

If we assume i.i.d., $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, so

- $p(x|\beta, \sigma) = \mathcal{N}(y_i | \sum_i \beta_k x_i k, \sigma_i^2)$
- $p(\beta) \propto 1$, $p(\log \sigma) \propto 1$
- $p(\beta, \sigma | x)$ is the regression result.

Simple example

Suppose we have data x_i and our data-generating model is

$$x_i \sim \mathcal{N}(\beta, \sigma)$$

and we don't impose that β has a parametric form.
Start with non-informative priors:

$$\begin{aligned} p(\beta) &\propto 1 \\ p(\log \sigma) &\propto 1 \end{aligned}$$

Using our math:

$$\begin{aligned} p(\beta, \sigma | x) &\propto p(\beta) p(\sigma) p(x | \beta, \sigma) \\ &\propto \sigma^{-n-2} e^{-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2} \\ &\propto \sigma^{-n-2} e^{-\frac{1}{2\sigma^2} \left[(n-1) \mathrm{Var}(x) + n(\bar{x} - \mu)^2\right]} \end{aligned}$$

Does that tell us anything?

Simple example

Suppose we have data x_i and our data-generating model is

$$x_i \sim \mathcal{N}(\beta, \sigma)$$

and we don't impose that β has a parametric form.
Start with non-informative priors:

$$p(\beta) \propto 1$$
$$p(\log\sigma) \propto 1$$

Using our math:

$$\begin{aligned} p(\beta, \sigma | x) &\propto p(\beta) p(\sigma) p(x | \beta, \sigma) \\ &\propto \sigma^{-n-2} e^{-\frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2} \\ &\propto \sigma^{-n-2} e^{-\frac{1}{2\sigma^2} \left[(n-1)\mathrm{Var}(x) + n(\bar{x} - \mu)^2\right]} \end{aligned}$$

Does that tell us anything? **Yes**: A relative measure of the probability of any given combination of μ and σ.

How do we calculate the actual distributions of β and σ?

- There is no closed-form solution.
- (This is not the case for every kind of parameter, but analytical solutions are rare cases.)
- The probability of a given value of β depends on σ.
- Various ways to numerically approximate.

```
## Bayesian estimate
library(rstan)

## A simple model in Stan
stan.code <- "
data {
  int<lower=0> N;
  vector[N] x;
}
parameters {
  real mu;
  real<lower=0> sigma;
}
model {
  x ~ normal(mu, sigma);
}"

## Fit the model to each set of data
fit <- stan(model_code=stan.code, data=list(N=N, x=xx))
fit1 <- stan(fit=fit, data=list(N=2*N, x=xx1))
fit2b <- stan(fit=fit, data=list(N=3*N, x=c(xx, xx1)))

la <- extract(fit)
la1 <- extract(fit1)
la2b <- extract(fit2b)
```

```r
## Plot a comparison vs. individual studies
ggplot() +
    geom_density(data=data.frame(x=la$mu), aes(x, colour="Study 1", linetype="Numeric")) +
    geom_density(data=data.frame(x=la1$mu), aes(x, colour="Study 2", linetype="Numeric")) +
    geom_line(data=data.frame(x=xgrid, y=gaus0*.96), aes(x, y, colour="Study 1", linetype="Approximate")) +
    geom_line(data=data.frame(x=xgrid, y=gaus1*1.05), aes(x, y, colour="Study 2", linetype="Approximate")) +
    theme_minimal() +
    scale_linetype_discrete(name="") + scale_colour_discrete(name="") + xlab(NULL)

## Plot a comparison vs. pooled
dens0 <- density(la$mu, from=beta.range[1], to=beta.range[2])
dens1 <- density(la1$mu, from=beta.range[1], to=beta.range[2])

ggplot() +
    geom_density(data=data.frame(x=la2b$mu), aes(x, colour="Pooled", linetype="Numeric")) +
    geom_line(data=data.frame(x=dens0$x, y=7.2*dens0$y * dens1$y), aes(x, y, colour="Updated", linetype="Numeric")
) +
    theme_minimal() +
    scale_linetype_discrete(name="") + scale_colour_discrete(name="") + xlab(NULL)
```

Full Bayesian regression

Suppose we have an OLS-style model:

$$y|\beta, \sigma, X \sim \mathcal{N}(X\beta, \sigma^2 I)$$

As used above, the (conditional) posterior distribution of β for a known σ has an analytical form:

$$\beta|\sigma, y \sim \mathcal{N}(\hat{\beta}, V_\beta \sigma^2)$$

where

$$\hat{\beta} = (X^\intercal X)^{-1} X^\intercal y$$
$$V_\beta = (X^\intercal X)^{-1}$$

The (marginal) posterior distribution of σ is also recognizable:

$$\sigma^2|y \sim \text{Inv-}\chi^2(n-k, s^2)$$

where

$$s^2 = \frac{1}{n-k}(y - X\hat{\beta})^\intercal (y - X\hat{\beta})$$

Algorithm for drawing from posterior distribution

1. Calculate the model in OLS. Let $\hat{\beta}$ be the OLS coefficients.
2. Use the model residuals to calculate s^2.
3. Use the model covariance matrix ($= s^2(X^\intercal X)^{-1}$) calcalculate V_β.
4. Draw a random value of σ^2 from the scaled inverse-χ^2 distribution.
5. Draw a random value of β from a multivariate normal, using that value of σ.

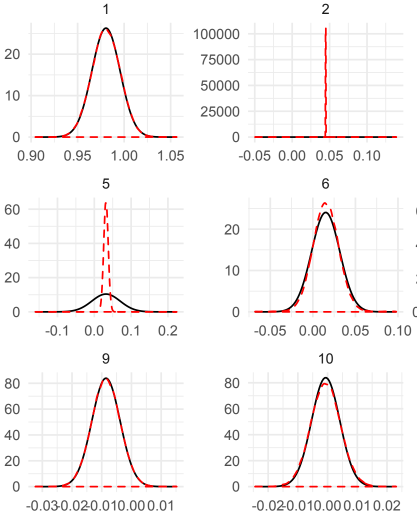
Properties:

- The OLS result will be the mode of the Bayesian posterior ($=$ MLE)
- The standard error is the second derivative of the Bayesian posterior

```r
## Load the data
data(treering)

## Setup the model
library(splines)

year <- -6000:1979
sin2 <- sin(2*pi*year / 2)
cos2 <- cos(2*pi*year / 2)
sin7 <- sin(2*pi*year / 7)
cos7 <- cos(2*pi*year / 7)

## Calculate the OLS model
mod <- lm(treering ~ ns(year, df=5) + sin2 + cos2 + sin7 + cos7)
summary(mod)

## Setup the
library(MASS)
library(geoR)

## Calculate the Bayesian model
nn <- length(treering)
kk <- 10
betahat <- mod$coeff
s2 <- sum(mod$resid^2) / (nn - kk)
V.beta <- vcov(mod) / s2
sigmas <- rinvchisq(100000, nn - kk, s2)
betas <- t(sapply(sigmas, function(sigma) mvrnorm(1, betahat, V.beta * sigma)))
```

```r
## Construct a comparison set of Gaussians
df <- data.frame(kk=c(), xx=c(), pp=c())
for (kk in 1:10) {
    xx <- seq(betahat[kk] - 5 * sqrt(vcov(mod)[kk, kk]), betahat[kk] + 5 * sqrt(vcov(mod)[kk, kk]),
              length.out=100)
    pp <- dnorm(xx, betahat[kk], sqrt(vcov(mod)[kk, kk]))
    df <- rbind(df, data.frame(kk, xx, pp))
}

## Construct a dataset of observations for density calculation
df.bayes <- data.frame(kk=c(), xx=c())
for (kk in 1:10) {
    ## Drop values beyond limits of OLS
    limitlo <- betahat[kk] - 5 * sqrt(vcov(mod)[kk, kk])
    limithi <- betahat[kk] + 5 * sqrt(vcov(mod)[kk, kk])
    valid <- betas[, kk] > limitlo & betas[, kk] < limithi
    df.bayes <- rbind(df.bayes, data.frame(kk, xx=betas[valid, kk]))
}

## Graph it!
library(ggplot2)
ggplot(df, aes(xx)) +
    facet_wrap(~ kk, scales="free", nrow=4, ncol=4) +
    geom_line(aes(y=pp)) +
    geom_density(data=df.bayes, col=2, linetype=2) +
    theme_minimal() + xlab(NULL) + ylab(NULL)
```

Comparison


1
2
3
4
5
6
7
8
9
10
0.90 0.95 1.00 1.05
-0.05 0.00 0.05 0.10
-0.10 -0.05 0.00 0.05 0.10
-0.1 0.0 0.1 0.2
-2e+10 0e+00 2e+10 4e+10
-0.01 0.00 0.01 0.02
-0.03 -0.02 -0.01 0.00 0.01
-0.02 -0.01 0.00 0.01 0.02
100000 75000 50000 25000 0
6e-11 4e-11 2e-11 0e+00