Chapter 11.8 Informative Prior

Jim Albert and Monika Hu

Chapter 11 Simple Linear Regression

Priors

- One challenge in a Bayesian analysis is the construction of a prior that reflects beliefs about the parameters
- Thinking about prior beliefs can be difficult since the intercept β_0 does not have a meaningful interpretation
- One way to help: standardization

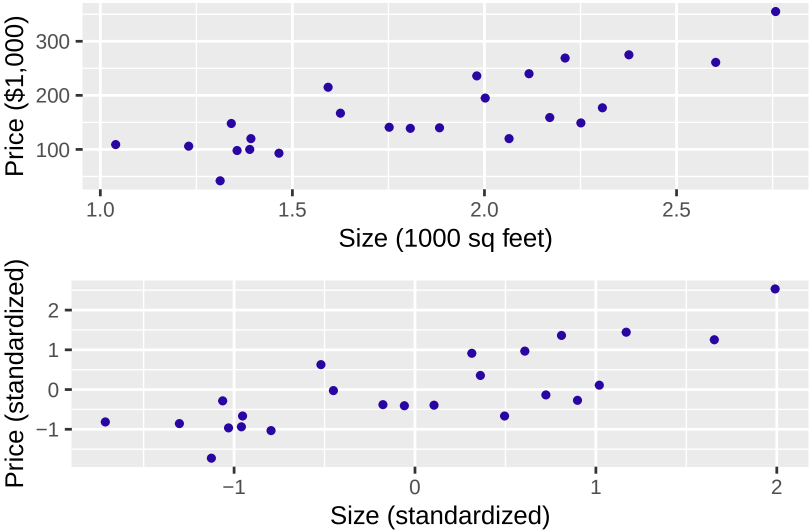
Standardization

- Standardization is the process of putting different variables on similar scales

$$y_i^* = \frac{y_i - \bar{y}}{s_y}, \ x_i^* = \frac{x_i - \bar{x}}{s_x} \tag{1}$$

```
PriceAreaData$price_standardized <-
  scale(PriceAreaData$price)
PriceAreaData$size_standardized <-
  scale(PriceAreaData$newsize)
```

Standardization cont'd



Standardization cont'd

- A standardized value represents the number of standard deviations that the value falls above or below the mean
- The ranges of the standardized scores for the x^* and y^* are similar: both sets of standardized scores fall between -2 and 2
- The association pattern of the two graphs agree which indicates that the standardization procedure has no impact on the relationship of house size with the sale price

Standardization cont'd

- Standardization of the variables provides more meaningful interpretations of the regression parameters β_0 and β_1

$$
\begin{aligned}
Y_i^* \mid \mu_i^*, \sigma &\overset{ind}{\sim} \mathrm{Normal}(\mu_i^*, \sigma), && (2)\\
\mu_i^* &= \beta_0 + \beta_1 x_i^* && (3)
\end{aligned}
$$

- The intercept parameter β_0 now is the expected standardized sale price for a house where $x_i^* = 0$ corresponding to a house of average size
- The slope β_1 gives the change in the expected standardized sale price μ_i^* when the standardized size x_i^* increases by 1 unit, or when the size variable increases by one standard deviation

Standardization cont'd

- In addition, when the variables are standardized, the slope β_1 can be shown equal to the correlation between x_i and y_i
  - a positive value β_1 indicates a positive linear relationship between the two variables
  - the absolute value of β_1 indicates the strength of the relationship

Prior distributions

- As before, assume that the three parameters β_0, β_1 and σ are independent

$$\pi(\beta_0, \beta_1, \sigma) = \pi(\beta_0)\pi(\beta_1)\pi(\sigma)$$

- The task of assigning a joint prior simplifies to the task of assigning priors separately to each of the three parameters
- We will describe the process one by one

Prior on the intercept β_0

- After standardization, the intercept β_0 represents the expected standardized sale price given a house of average size (i.e. $x_i^* = 0$)
- If we believe a house of average size will also have an average price, then a reasonable guess of β_0 is zero
- We can give a normal prior for β_0 with mean $\mu_0 = 0$ and standard deviation s_0:

$$\beta_0 \sim \mathrm{Normal}(0, s_0)$$

- The standard deviation s_0 in the normal prior reflects how confident we believe in the guess of $\beta_0 = 0$

Prior on the slope β_1

- After standardization, the slope β_1 represents the correlation between the house size and the sale price
- We represent our belief about the location of β_1 by means of a normal prior.

$$\beta_1 \sim \mathrm{Normal}(\mu_1, s_1)$$

- μ_1 represents our guess of the correlation
- s_1 represents the sureness of this guess

Prior on σ

- As before, assign a weakly informative prior for the sampling error standard deviation σ

$$1/\sigma^2 \sim \mathrm{Gamma}(1, 1)$$

Summary of priors

$$
\begin{aligned}
\pi(\beta_0, \beta_1, \sigma) &= \pi(\beta_0)\pi(\beta_1)\pi(\sigma) \\
\beta_0 &\sim \mathrm{Normal}(0, 1) \\
\beta_1 &\sim \mathrm{Normal}(0.7, 0.15) \\
1/\sigma^2 &\sim \mathrm{Gamma}(1, 1)
\end{aligned}
$$

Posterior analysis

- Use JAGS
- Make sure to work with standardized data

```
PriceAreaData$price_standardized <-
  scale(PriceAreaData$price)
PriceAreaData$size_standardized <-
  scale(PriceAreaData$newsize)
```

JAGS step 1: describe the model by a script

- Same `modelString` as before

```
modelString <-"
model {
## sampling
for (i in 1:N){
y[i] ~ dnorm(beta0 + beta1*x[i], invsigma2)
}
## priors
beta0 ~ dnorm(mu0, g0)
beta1 ~ dnorm(mu1, g1)
invsigma2 ~ dgamma(a, b)
sigma <- sqrt(pow(invsigma2, -1))
}"
```

JAGS step 2: define the data and prior parameters

```
y <- as.vector(PriceAreaData$price_standardized)
x <- as.vector(PriceAreaData$size_standardized)
N <- length(y)
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 1,
                 "mu1" = 0.7, "g1" = 44.4,
                 "a" = 1, "b" = 1)
```

JAGS step 3: generate samples from the posterior distribution

```
posterior2 <- run.jags(modelString,
                       n.chains = 1,
                       data = the_data,
                       monitor = c("beta0",
                                   "beta1", "sigma"),
                       adapt = 1000,
                       burnin = 5000,
                       sample = 5000)
```

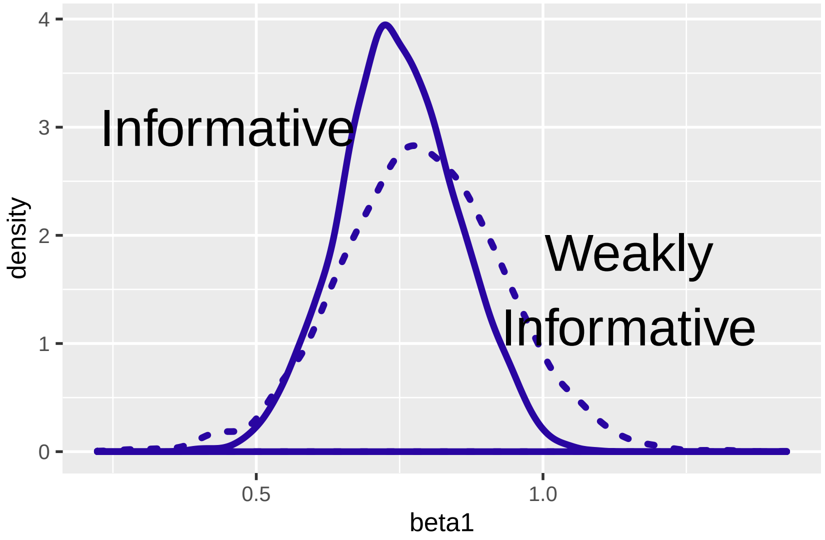
Comparing posteriors for two priors

- To understand the influence of the informative prior, we can contrast this posterior distribution with a posterior using a weakly informative prior

```
the_data <- list("y" = y, "x" = x, "N" = N,
                 "mu0" = 0, "g0" = 0.0001,
                 "mu1" = 0.7, "g1" = 0.0001,
                 "a" = 1, "b" = 1)
```

```
posterior3 <- run.jags(modelString,
                       n.chains = 1,
                       data = the_data,
                       monitor = c("beta0",
                                   "beta1", "sigma"),
                       adapt = 1000,
                       burnin = 5000,
                       sample = 5000)
```

Comparing posteriors for two priors cont'd



Comparing posteriors for two priors cont'd

- The "informative prior" posterior has less spread than the "weakly informative prior" posterior
- The "informative prior" posterior shifts the "weakly informative prior" posterior towards the prior belief that the slope is close to the value 0.7

Comparing posteriors for two priors cont'd

```
print(posterior2, digits = 3)
      Lower95   Median Upper95     Mean    SD Mode   MC
beta0  -0.267 0.000358   0.276 0.000372 0.138   -- 0.00
beta1   0.551    0.751   0.959    0.749 0.104   -- 0.00
sigma   0.498     0.67   0.878    0.682 0.102   -- 0.00

print(posterior3, digits = 3)
      Lower95   Median Upper95     Mean    SD Mode   MC
beta0  -0.273 0.000362   0.281 0.000421 0.141   -- 0.00
beta1   0.501    0.794    1.08    0.792 0.146   -- 0.00
sigma   0.502    0.677   0.894    0.688 0.105   -- 0.00
```