Module 12: Linear Regression, the g-prior, and model selection

Rebecca C. Steorts

Agenda

Setup

- $X_{n \times p}$: regression features or covariates (design matrix)
- $x_{p \times 1}$: ith row vector of the regression covariates
- $y_{n \times 1}$: response variable (vector)
- $\beta_{p \times 1}$: vector of regression coefficients

Goal: Estimation of $p(y \mid x)$.

Dimensions: $y_i - \beta^T x_i = (1 \times 1) - (1 \times p)(p \times 1) = (1 \times 1)$.

Multivariate Setup

Let's assume that we have data points (x_i, y_i) available for all $i = 1, \ldots, n$.

- y is the response variable

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}_{n \times 1}$$

- x_i is the ith row of the design matrix $X_{n \times p}$.

Consider the regression coefficients

$$\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}_{p \times 1}$$

Multivariate Setup

$$y \mid X, \beta, \sigma^2 \sim MVN(X\beta, \sigma^2 I)$$
$$\beta \sim MVN(\beta_0, \Sigma_0)$$

Recall the posterior can be shown to be

$$\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X} \sim MVN(\boldsymbol{\beta}_{\boldsymbol{n}}, \Sigma_{\boldsymbol{n}})$$

where

$$\boldsymbol{\beta}_{\boldsymbol{n}} = E[\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2] = (\Sigma_{\boldsymbol{o}}^{-1} + (X^T X)^{-1}/\sigma^2)^{-1}(\Sigma_{\boldsymbol{o}}^{-1}\boldsymbol{\beta}_0 + \boldsymbol{X}^T\boldsymbol{y}/\sigma^2)$$

$$\Sigma_{\boldsymbol{n}} = \mathrm{Var}[\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2] = (\Sigma_{\boldsymbol{o}}^{-1} + (X^T X)^{-1}/\sigma^2)^{-1}$$

How do we specify $\boldsymbol{\beta}_0$ and Σ_0?

The g-prior

To do the *least amount of calculus*, we can put a *g-prior* on β

$$\beta \mid \boldsymbol{X}, \boldsymbol{z} \sim MVN(0, g\,\sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}).$$

$$\beta_n = E[\beta \ \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2] = \frac{g}{g+1}(\Sigma_o^{-1} + (X^TX)^{-1}/\sigma^2)^{-1} = \frac{g}{g+1}\hat{\beta}_{ols}$$

$$\Sigma_n = \mathrm{Var}[\beta \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2] = \frac{g}{g+1}(X^TX)^{-1}/\sigma^2)^{-1} = \frac{g}{g+1}\mathrm{Var}[\hat{\beta}_{ols}]$$

- g shrinks the coefficients and can prevent overfitting to the data
- if $g = n$, then as n increases, inference approximates that using $\hat{\beta}_{ols}$

Variance component σ^2

What about a prior on $1/\sigma^2 = \lambda$

$$y \mid X, \beta, \sigma^2 \sim MVN(X\beta, \sigma^2 I) \tag{1}$$
$$\lambda \sim \text{Gamma}(\nu_0/2, \nu_0\lambda^{-1}/2) \tag{2}$$

Then the posterior can be shown to be

$$p(\lambda \mid y, X) \sim \text{Gamma}([\nu_0 + n]/2, [\nu_0\lambda^{-1} + SSR_g]/2)$$

where SSR_g is somewhat complicated (see Hoff for details, p. 158).

Variance component σ^2

The joint distribution can be written as

$$p(\lambda^{-1}, \boldsymbol{\beta} \mid y, X) = p(\lambda^{-1} \mid y, X) \times p(\boldsymbol{\beta} \mid y, X, \lambda^{-1})$$

Goal: simulate $(\lambda^{-1}, \beta) \sim p(\lambda^{-1}, \boldsymbol{\beta} \mid y, X)$ Starting value (β_0, λ_0)
1. Simulate

$$\lambda^{-1} \sim p(\lambda^{-1} \mid y, X)$$

Gives us $(\lambda_1^{-1}, \beta_0)$ 2. Use this updated value of λ_1^{-1} to simulate

$$\beta \sim p(\boldsymbol{\beta} \mid y, X, \lambda^{-1})$$

Gives us $(\lambda_1^{-1}, \beta_1)$

Run the sampler for S iterations.

Back to the oxygen uptake example

$$y \mid X, \beta, \sigma^2 \sim MVN(X\beta, \lambda^{-1} I) \quad (3)$$

$$\beta \mid \lambda MVN(0, g(X^T X)^{-1}) \quad (4)$$

$$\lambda^{-1} \sim \text{Gamma}(\nu_0/2, \nu_0 \lambda^{-1}/2) \quad (5)$$

We will use the g-prior, where

1. $g = n$
2. $\nu_0 = 1$
3. $\sigma_0^2 = \lambda^{-1} = \hat{\sigma}_{ols} = 8.54$

Application to diabetes (Exercise 9.2, part a)

As described in Exercise 7.6, suppose we have data on health-related variables of a population of 532 women.

In this exercise we will be modeling the conditional distribution of glucose level (glu) as a linear combination of the other variables, excluding the variable diabetes.

Regression model on the g-prior

Fit a regression model using the g-prior with $g = n$, $\nu_0 = 2$ and $\sigma_0^2 = 1$. Obtain posterior confidence intervals for all of the parameters.

Regression model on the g-prior

Section 9.2.2 (Hoff) shows that under the g prior, $p(\sigma^2 \mid \boldsymbol{y}, \boldsymbol{X})$ and $p(\boldsymbol{\beta} \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2)$ are inverse gamma and multivariate normal distributions respectively.

Regression model on the g-prior

Therefore samples from the joint posterior $p(\sigma^2, \beta \mid \boldsymbol{y}, \boldsymbol{X}, \sigma^2)$ can be made with a Monte Carlo approximation.

We first center and scale all the variables so that there is no need to include an intercept in the model.

Regression model on the g-prior

```r
library(knitr)
rm(list=ls())
azd_data = read.table("azdiabetes.dat", header = TRUE)
head(azd_data)
```

```
##   npreg glu bp skin  bmi   ped age diabetes
## 1     5  86 68   28 30.2 0.364  24       No
## 2     7 195 70   33 25.1 0.163  55      Yes
## 3     5  77 82   41 35.8 0.156  35       No
## 4     0 165 76   43 47.9 0.259  26       No
## 5     0 107 60   25 26.4 0.133  23       No
## 6     5  97 76   27 35.6 0.378  52      Yes
```

Regression model on the g-prior

```r
y = azd_data$glu
X = as.matrix(azd_data[,c(-2,-8)])
head(X)
```

```
##      npreg bp skin  bmi   ped age
## [1,]     5 68   28 30.2 0.364  24
## [2,]     7 70   33 25.1 0.163  55
## [3,]     5 82   41 35.8 0.156  35
## [4,]     0 76   43 47.9 0.259  26
## [5,]     0 60   25 26.4 0.133  23
## [6,]     5 76   27 35.6 0.378  52
```

Standardization

```r
# standardize data to have mean 0 and variance 1
ys = scale(y)
Xs = scale(X)
n = dim(Xs)[1]
p = dim(Xs)[2]
```

Hyper-parameters

hyper-parameters

```
g = n
nu0 = 2
s20 = 1
```

Intermediate Matrices

```
# intermediate matrices
Hg = (g/(g+1)) * Xs %*% solve(t(Xs) %*% Xs) %*% t(Xs)
SSRg = t(ys) %*% ( diag(1,nrow=n) - Hg ) %*% ys
```

Monte carlo

```
# number of posterior samples
S = 1000

# generate posteriors
s2 = 1/rgamma(S, (nu0+n)/2, (nu0*s20 + SSRg)/2)
Vb = g * solve(t(Xs) %*% Xs)/(g+1)
Eb = Vb %*% t(Xs) %*% ys
E = matrix(rnorm(S*p, 0, sqrt(s2)),S,p)
beta_s = t( t(E %*% chol(Vb)) + c(Eb))

# transform coefficients to the original scale
sd_X = apply(X,2,sd)
Beta_a = sweep(beta_s,2,sd_X,FUN = "/")
```

The 95% posterior confidence intervals

```
# 95% credible interval
Beta_CIa = apply(Beta_a, 2, quantile, c(0.025, 0.975))
kable(data.frame(Beta_CIa))
```

	npreg	bp	skin	bmi	ped
2.5%	-0.0515254	-0.0005462	-0.0033335	0.0059186	0.1047847
97.5%	0.0091583	0.0133459	0.0159199	0.0352423	0.5617721

Model selection

- Often we have a large number of covariates.
- Using all of them induces poor statistical performance.
- How can we reduce the covariates and have good inference and prediction?
- Common method: Backwards and stepwise regression (slow).

Model selection

Suppose that we believe some of the regression coefficients are 0.

Come up with a prior distribution that reflects the probability of this occuring.

Consider

$$y_i = z_1 b_1 x_{i,1} + \dots z_p b_p x_{i,p},$$

where b_p is a real number and z_j indicate which regression coefficients are nonzero.

Note: $\beta_j = b_j \times z_j$.

Bayesian model selection

Bayesian model selection works by obtaining a posterior distribution for $\boldsymbol{z}$.

Assume a prior $p(\boldsymbol{z})$.

Then

$$p(\boldsymbol{z} \mid \boldsymbol{Y}, \boldsymbol{X}) = \frac{p(\boldsymbol{z})p(\boldsymbol{Y} \mid \boldsymbol{X}, z)}{\sum_z p(\boldsymbol{z})p(\boldsymbol{Y} \mid \boldsymbol{X}, z)}$$

Bayesian model selection

Suppose we want to compare two models z_a and z_b . Consider

$$\text{odds}(z_a, z_b \mid \boldsymbol{Y}, \boldsymbol{X}) = \frac{p(z_a \mid \boldsymbol{Y}, \boldsymbol{X})}{p(z_b \mid \boldsymbol{Y}, \boldsymbol{X})} = \frac{p(z_a)}{p(z_b)} \times \frac{p(\boldsymbol{Y} \mid \boldsymbol{X}, z_a)}{p(\boldsymbol{Y} \mid \boldsymbol{X}, z_b)}$$

This is posterior odds $=$ prior odds $\times$ “Bayes factor"

“Bayes factor“: how much the data favor model z_a over model z_b

To obtain a posterior distribution over models, we must compute $p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z})$ for *each* model under consideration.

Bayesian model selection

We must compute

$$p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z}) = \int\int p(\boldsymbol{Y}, \beta, \sigma^2, \mid \boldsymbol{X}, \boldsymbol{z}) \tag{6}$$

$$\int\int p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z}) p(\beta \mid \boldsymbol{X}, \boldsymbol{z}) p(\sigma^2). \tag{7}$$

To do the *least amount of calculus*, we can put a *g-prior* on β

$$\beta \mid \boldsymbol{X}, \boldsymbol{z} \sim MVN(0, g\, \sigma^2 (\boldsymbol{X}^T \boldsymbol{X})^{-1}).$$

Back to the g-prior

Given the g-prior

$$\beta \mid \boldsymbol{X}, \boldsymbol{z} \sim MVN(0, g\, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}),$$

$p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z})$ can be worked out in closed form (details p. 165).

Go through the details on your own.

Back to the g-prior

This results in being able to compute

$$\frac{p(\boldsymbol{Y} \mid \boldsymbol{X}, z_a)}{p(\boldsymbol{Y} \mid \boldsymbol{X}, z_b)} = (1+n)^{(p_{z_b} - p_{z_a})/2} \times \left(\frac{s^2_{z_a}}{s^2_{z_b}} \right)^{1/2} \tag{8}$$

$$\times \left(\frac{s^2_{z_b} + SSR^{z_b}_g}{s^2_{z_b} + SSR^{z_a}_g} \right)^{(n+1)/2} \tag{9}$$

We have a ratio of the marginal probabilities, giving us a balance between model complexity and model fit.

Suppose p_{z_b} is large compared to p_{z_a}.

This causes a penalization of model z_b

Note that a large value of $SSR^{z_b}_g$ compared to $SSR^{z_a}_g$ will penalize model z_a.

Bayesian Model Averaging

Suppose that we are content with a estiamte of β from which we can make predictions.

We may also want a list of relatively high probablitiy models.

We can use a Markov chain to search through the space of models for values of z with high posterior probability.

Bayesian model averaging

Suppose p is large. Then 2^p models to consider.

Instead let's use a Gibbs sampler to search through the space of models for values where $\boldsymbol{z}$ has a high posterior probability.

Generate a new value of $\boldsymbol{z}$ via

$$p(z_j \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{z}_{-j}).$$

The full conditional that $z_j = 1$ can be written as $o_j/(o_j + 1)$.

$$\begin{aligned} o_j &= \frac{p(z_j = 1 \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{z}_{-j})}{p(z_j = 0 \mid \boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{z}_{-j})} && (10) \\ &= \frac{p(z_j = 1)p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z}_{-j}, z_j = 1)}{p(z_j = 0)p(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{z}_{-j}, z_j = 0)} && (11) \end{aligned}$$

Bayesian model averaging

Note: we may also want to obtain posterior samples of $\boldsymbol{\beta}$ and σ^2.

Using the conditional distributions from Section 9.2, we can sample from these directly.

The Gibbs sampling scheme requires using Section 9.2 and 9.3 (covered in lab).

Bayesian model averaging

$$\begin{tikzcd}
\boldsymbol{z}^{(s)} \arrow[r] \arrow[d] & \sigma^{2(s)} \arrow[r] & \boldsymbol{\beta}^{(s)} \\
\boldsymbol{z}^{(s+1)} \arrow[r] & \sigma^{2(s+1)} \arrow[r] & \boldsymbol{\beta}^{(s+1)}
\end{tikzcd}$$

Figure 1: Start with $\boldsymbol{z}^{(s)}$. Then in random order update z_j from its full conditional.

Bayesian model averaging

Generate

$$\{\boldsymbol{z}^{(s+1)}, \sigma^{2(s+1)}, \beta^{(s+1)}\} :$$

1. Set $\boldsymbol{z} = \boldsymbol{z}^{(s)}$
2. For $j \in \{1, \ldots, p\}$ in random order, replace z_j with a sample from
$$p(z_j \mid \boldsymbol{z}_{-j}, \boldsymbol{Y}, \boldsymbol{X})$$
3. Set $\boldsymbol{z}^{(s+1)} = \boldsymbol{z}^{(s)}$
4. Sample $\sigma^{2(s)} \sim p(\sigma^2 \mid \boldsymbol{z}^{(s+1)}, \boldsymbol{Y}, \boldsymbol{X})$
5. Sample $\beta^{(s+1)} \sim p(\beta \mid \boldsymbol{z}^{(s+1)}, \sigma^{2(s+1)}, \boldsymbol{Y}, \boldsymbol{X})$

Back to diabetes data (Exercise 9.2, b)

Let's perform Bayesian model averaging (as described in Section 9.3)

Obtain $P(\beta_j \neq | \ y)$ as well as posterior confidence intervals for all of the parameters. Compare our results to that in part (a.)

Back to diabetes data (Exercise 9.2, b)

The following function `lpy.X` calculates the log of $p(\boldsymbol{y} \mid \boldsymbol{X})$, which we will use in implementing the Gibbs sampler for Bayesian model averaging.

```
## a function to compute the marginal probability
lpy.X = function(y, X, g=length(y), nu0=1, s20=try(summary(
  n = dim(X)[1]
  p = dim(X)[2]
  if (p==0) { Hg = 0; s20 = mean(y^2)}
  if (p>0){ Hg = (g/(g+1)) * X %*% solve(t(X) %*% X) %*% t(
  SSRg = t(y) %*% ( diag(1, nrow=n) - Hg ) %*% y -
    .5*( n*log(pi) + p*log(1+g) + (nu0+n)*log(nu0*s20 + SSR
    lgamma( (nu0+n)/2 ) - lgamma(nu0/2)
}
```

Back to diabetes data (Exercise 9.2, b)

Let $\boldsymbol{z}$ be the random binary vector of variable indicators. Generating samples of $p(\boldsymbol{z}, \sigma^2, \beta)$ from the joint posterior distribution is achieved with the following steps:

1. For $j \in \{1, \ldots, p\}$ in random order, draw z_j from $p(z_j \mid \boldsymbol{z}_{-j}, \boldsymbol{y}, \boldsymbol{X})$.
2. Sample $\sigma^2 \sim p(\sigma^2 \mid \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{X})$.
3. Sample $\beta \sim p(\beta \mid \boldsymbol{z}, \sigma^2, \boldsymbol{y}, \boldsymbol{X})$.

Gibbs sampler

```
## MCMC setup
g = n
nu0 = 1 # unit information prior
z = rep(1, p)
lpy.c = lpy.X(ys, Xs[,z==1,drop=FALSE])
S = 10000
Z = matrix(NA, S, p)
# Sigma2 = numeric(S)
B = matrix(0, S, p)

## Gibbs sampler
for(s in 1:S){
  # if(s %% 100 ==0) {print(s)}
  # sample z
  for (j in sample(1:p)){
    zp = z
    zp[j] = 1 - zp[j]
    lpy.p = lpy.X(ys,Xs[, zp==1, drop=FALSE])
```