Homework 7

STT 465, Bayesian Statistical Methods

Lowell Monis

November 14, 2025

Question 1

The file "crime.txt" contains crime rates and data on 15 predictor variables for 47 U.S. states. Download the file onto your computer, and import the data into R using:

```
crime.data <- read.table("data/crime.txt", header = TRUE)
yf <- crime.data[, 1]; Xf <- as.matrix(crime.data[, -1])
```

Then, `yf` has the crime rates and `Xf` has the 15 predictor variables. Both the crime rates and 15 predictor variables have been centered and scaled to have mean zero and variance one. A description of the 15 predictor variables is in the file "crime_variables.pdf".

(a) 30 points

Fit a linear regression model by treating the crime rates as the response variable and including the 15 predictor variables (no need to add interaction or quadratic terms). Using the invariant g-prior with parameter values $g = n = 47, a = 1, b = 1$, obtain Monte Carlo approximations to the posterior mean and 95% credible interval for each regression coefficient β_j $(j = 1, 2, \ldots, 15)$. Which variables seem strongly predictive of crime rates?

Answer

First, we set up the parameters and sample storage as defined by the question. We also load the `mvtnorm` package.

```
library(mvtnorm)

# dimensions and invariant g-prior
n <- length(yf) -> g
p <- ncol(Xf)
beta_0 <- rep(0,p) # 15x1 zero vector
Sigma_0 <- diag(p) # 15x15 identity matrix
a <- 1 -> b

S <- 5000 # MCMC samples

# storage
beta_sam <- matrix(0, nrow=S, ncol=p)
sigma2_sam <- numeric(S)
```

We consider the normal linear regression model to be:

$$Y = X\beta + \epsilon$$

where $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_n)$.

Further, Y is the $n \times 1$ response vector (crime rates), X is the $n \times p$ data matrix containing the predictors, β is the $p \times 1$ coefficient vector, and σ^2 is the error variance.

The question asks me to use the invariant g-prior with $g = n = 47, a = b = 1$. The invariant g-prior as formulated by Zellner in 1986 is a special case where $\beta_0 = 0$ and $\Sigma_0 = \kappa(X^T X)^{-1}$, with $\kappa = g\sigma^2$.

Under the invariant g-prior, the joint posterior distribution $p(\beta, \sigma^2 \mid y, X)$ can be sampled directly without a Gibbs sampler because $p(\sigma^2 \mid y, X)$ is an Inverse-Gamma distribution.

The sampling process involves two steps:

1. Sample σ^2 from the marginal posterior distribution $p(\sigma^2 \mid y, X) \sim \text{Inverse} - \text{Gamma}(\tilde{a}, \tilde{b})$ where $\tilde{a} = a + \frac{n}{2}$ and $\tilde{b} = b + \frac{1}{2}y^T y - \frac{g}{2(g+1)} y^T X (X^T X)^{-1} X^T y$.
2. Sample β from the conditional posterior distribution $p(\beta \mid y, X, \sigma^2) = \mathcal{N}(\beta_n, \Sigma_n)$ where $\beta_n = \frac{g}{g+1}(X^T X)^{-1} X^T y$ and $\Sigma_n = \frac{g}{g+1}\sigma^2 (X^T X)^{-1}$.

I can now set up the specific parameters and the linear algebra needed.

```
# constant quantities
XtX_inv <- solve(t(Xf) %*% Xf)
yTy <- sum(yf^2)
XtX_inv_Xt_y <- XtX_inv %*% t(Xf) %*% yf

# parameters
beta_n <- (g / (g + 1)) * XtX_inv_Xt_y
Sig_constant <- (g / (g + 1)) * XtX_inv
a_tilde <- a+(n/2)
b_tilde <- b + yTy / 2 -
  (g / (2 * (g + 1))) * (t(yf) %*% Xf %*% XtX_inv %*% t(Xf) %*% yf)
```

We can now use composition sampling to generate independent Monte Carlo samples from $p(\beta, \sigma^2 \mid y, X)$.

```
set.seed(465)
for (i in 1:S) {
  sigma2_sam[i] <- 1 / rgamma(1, shape = a_tilde, rate = b_tilde)
  Sigma_n <- Sig_constant * sigma2_sam[i]
  beta_sam[i, ] <- rmvnorm(1, mean = beta_n, sigma = Sigma_n)
}
```

The Monte Carlo samples are used to approximate the posterior mean the 95% credible intervals for each regression coefficient β_j. I am also adding a flag that checks whether we are 95% confident that the coefficient is non-zero. A variable is only considered strongly predictive if its 95% credible interval does not include zero, i.e., the coefficient is non-zero by a fair chance and contributes the variable greatly to the final model. Another way to put this is that there is strong evidence that there is evidence of a difference between the two groups, when zero is not included in the credible interval.

```r
posterior_means <- colMeans(beta_sam)
credible_intervals <- t(apply(beta_sam, 2, quantile, probs = c(0.025, 0.975)))

var_names <- colnames(Xf)
results <- data.frame(
  Variable = var_names,
  Posterior_Mean = round(posterior_means, 4),
  CI_Lower = round(credible_intervals[, 1], 4),
  CI_Upper = round(credible_intervals[, 2], 4),
  CI_Excludes_Zero = (credible_intervals[, 1] > 0) | (credible_intervals[, 2] < 0)
)
results
```

```
##    Variable Posterior_Mean CI_Lower CI_Upper CI_Excludes_Zero
## 1         M         0.2774   0.0327   0.5242             TRUE
## 2        So         0.0004  -0.3414   0.3376            FALSE
## 3        Ed         0.5316   0.2069   0.8466             TRUE
## 4       Po1         1.4305   0.0079   2.8881             TRUE
## 5       Po2        -0.7600  -2.3084   0.7484            FALSE
## 6        LF        -0.0641  -0.3436   0.2132            FALSE
## 7       M.F         0.1310  -0.1531   0.4047            FALSE
## 8       Pop        -0.0674  -0.2933   0.1547            FALSE
## 9        NW         0.1085  -0.1985   0.4055            FALSE
## 10       U1        -0.2683  -0.6340   0.1020            FALSE
## 11       U2         0.3637   0.0331   0.6892             TRUE
## 12      GDP         0.2367  -0.2335   0.7009            FALSE
## 13     Ineq         0.7113   0.2814   1.1437             TRUE
## 14     Prob        -0.2791  -0.5200  -0.0324             TRUE
## 15     Time        -0.0606  -0.3002   0.1884            FALSE
```

Upon running our model, it looks like `M`, `Ed`, `Po1`, `U2`, `Ineq`, and `Prob` are strongly predictive of crime rates. This means that the percentage of males aged 14-24 in the state, the mean years of schooling in the state, the state's police funding in the year 1960, the unemployment rate of urban males aged 35-39, the state's income inequality and the probability of being imprisoned as a resident of the state are the strongest predictors of crime rates in the context of this data. All variables as stated above are positively related to the crime rate, except for the probability of imprisonment, which is negatively related to the crime rate. The more likely it is to get imprisoned in the state, the lesser the crime rate.

(b) 30 points

Let's see how well the linear regression model can predict crime rates based on the 15 predictor variables. Randomly divide the data roughly in half, into a training dataset $\{\mathbf{y}_{tr} \in \mathbb{R}^{24}, \mathbf{X}_{tr} \in \mathbb{R}^{24\times 15}\}$ and a test dataset $\{\mathbf{y}_{test} \in \mathbb{R}^{23}, \mathbf{X}_{test} \in \mathbb{R}^{23\times 15}\}$.

Answer

Using only the training dataset, compute OLS $\hat{\boldsymbol{\beta}}_{ols}$. Obtain predicted values $\hat{\mathbf{y}}_{test} = \mathbf{X}_{test} \cdot \hat{\boldsymbol{\beta}}_{ols}$ for the test data. Plot $\hat{\mathbf{y}}_{test}$ versus $\mathbf{y}_{test}$ and compute the mean squared predictive error $\frac{1}{23}\sum_{i=1}^{23}(\hat{y}_{test,i} - y_{test,i})^2$.

To evaluate the predictive performance of the linear regression model, we will use the OLS method on the training data and assess the predictions on the separate test data.

Including all variables in a linear regression model often leads to poor predictive performance, quantified by a high mean squared predictive error on the test set. This suggest the need for model selection to include variables with substantial evidence of an association with the response.

I will first prepare the data by splitting it. I set a seed for the random split at `465`. A lot of the variables will be carried forward from earlier.

```
set.seed(465)

# randomly sample 23 indices for test set
i_test <- sample(1:n,23)

y_tr <- yf[-i_test]
X_tr <- Xf[-i_test,]
y_test <- yf[i_test]
X_test <- Xf[i_test,]
```

I can now compute the OLS coefficient estimate $\hat{\boldsymbol{\beta}}_{ols}$ using only the training data, $\mathbf{X}_{tr}$ and $\mathbf{y}_{tr}$. Since the variables were already centered and scaled, I fit the model without an intercept because the response and predictors have mean zero.

The OLS estimate is calculated as $\hat{\beta}_{ols} = (\mathbf{X}_{tr}^T\mathbf{X}_{tr})^{-1}\mathbf{X}_{tr}^T\mathbf{y}_{tr}$. I can then obtain the predicted values for the test data as $\hat{\mathbf{y}}_{test} = \mathbf{X}_{test} \cdot \hat{\boldsymbol{\beta}}_{ols}$.

```
ols <- lm(y_tr~-1+X_tr)
beta_ols <- ols$coef
y_hat_test <- X_test %*% beta_ols
```

The predictive performance can now be evaluated by computing the mean squared predictive error for the test set as follows:

$$\text{MSPE} = \frac{1}{23}\sum_{i=1}^{23}(\hat{y}_{test,i} - y_{test,i})^2$$

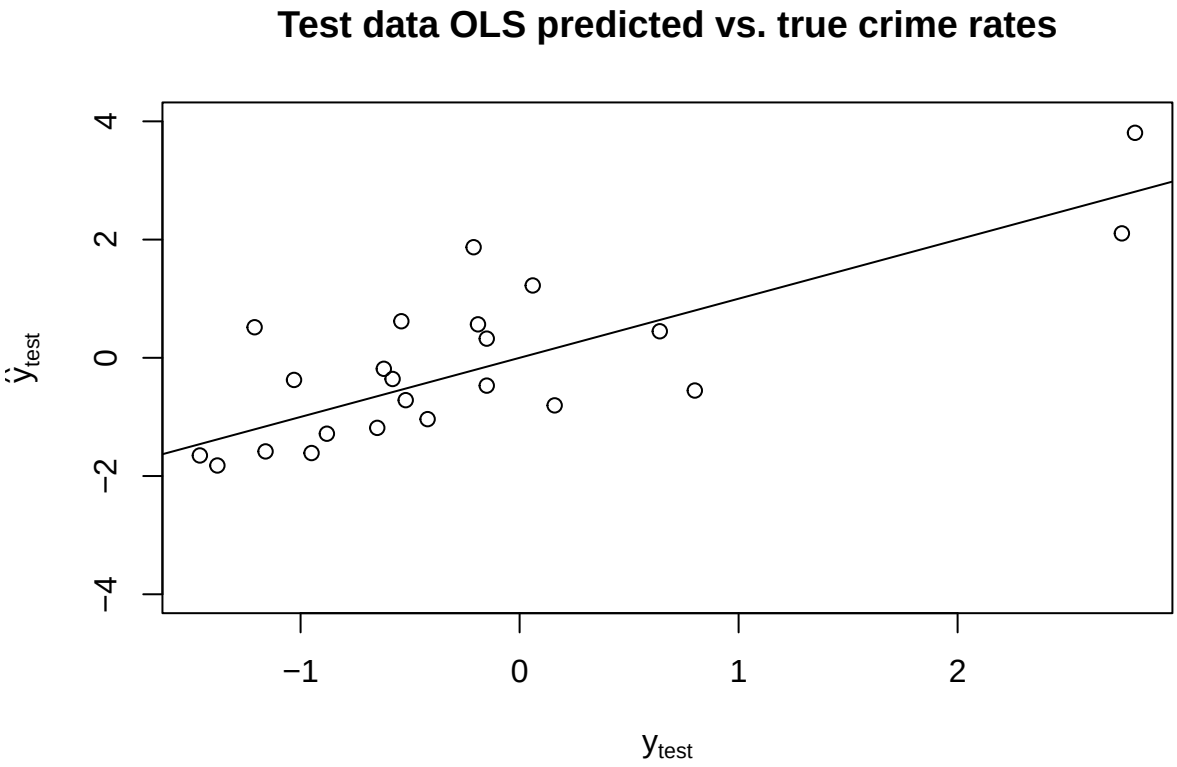
```
mspe <- mean((y_hat_test - y_test)^2)
mspe
```

```
## [1] 0.7609347
```

The mean squared predictive error is 76.093467 percent, which is really high.

I will now visualize the model and compare the predictions against the actual test values by plotting them against each other.

```
plot(y_test, y_hat_test,
     xlab = expression(y[test]),
     ylab = expression(hat(y)[test]),,
     ylim = c(-4,4),
     main = "Test data OLS predicted vs. true crime rates")
abline(0, 1)
```



From the high value of MSPE, and high variance in the test predictions with a majority of the points being far from the ideal range where $\hat{\mathbf{y}}_{test} = \mathbf{y}_{test}$, or the predicted test values are at least close to the actual test values, I can conclude that this is bad model that is probably overfitted as evident from the high test error. These diagnostics provide a quantitative measure of how well the full model with all 15 predictors generalizes to unseen data. It looks like model selection might be necessary to improve prediction performance.

Now obtain the posterior mean $\hat{\boldsymbol{\beta}}_{Bayes} = E(\boldsymbol{\beta}|\mathbf{y}_{tr}, \mathbf{X}_{tr})$ using the g-prior with $g = 24, a = 1, b = 1$ and only the training dataset (run similar procedures as in (1a)). Obtain predicted values $\hat{\mathbf{y}}_{test} = \mathbf{X}_{test}\hat{\boldsymbol{\beta}}_{Bayes}$ for the test data. Plot $\hat{\mathbf{y}}_{test}$ versus $\mathbf{y}_{test}$ and compute the mean squared predictive error $\frac{1}{23}\sum_{i=1}^{23}(\hat{y}_{test,i} - y_{test,i})^2$. Compare the results to those of the full model.

I will first re-establish the training and test sets using the same random seed to ensure the split is identical to the one used for the full OLS model. Thus, I will retain these variables from above. I will now set the invariant g-prior parameters. I will retain `a` and `b` from (1a) too.

```
g <- nrow(X_tr)
```

The posterior mean for the invariant g-prior is given by the closed-form solution:

$$\hat{\boldsymbol{\beta}}_{Bayes} = E(\boldsymbol{\beta}|\mathbf{y}_{tr}, \mathbf{X}_{tr}) = \frac{g}{g+1}(\mathbf{X}_{tr}^T\mathbf{X}_{tr})^{-1}\mathbf{X}_{tr}^T\mathbf{y}_{tr}$$

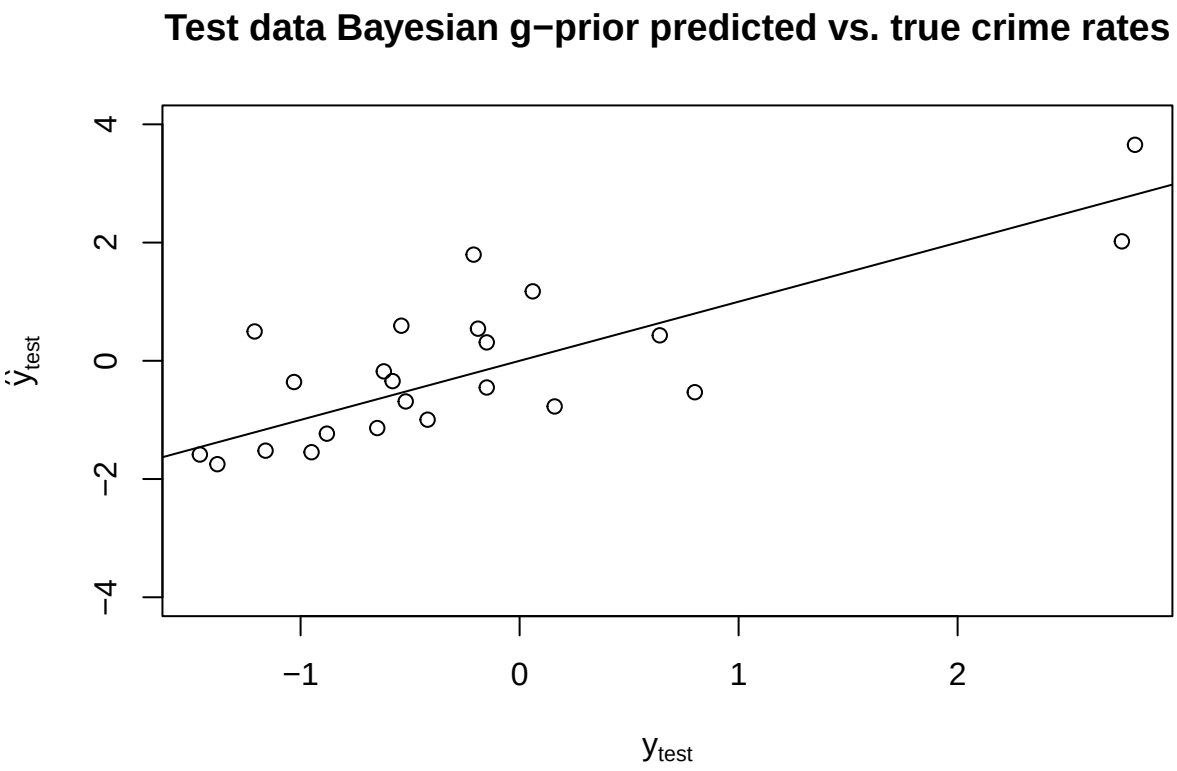
The formula is a shrinkage estimator that pulls the OLS estimate $(\mathbf{X}_{tr}^T\mathbf{X}_{tr})^{-1}\mathbf{X}_{tr}^T\mathbf{y}_{tr}$ towards zero by a factor of $\frac{g}{g+1}$.

```r
beta_Bayes <- (g / (g + 1)) * solve(t(X_tr) %*% X_tr) %*% t(X_tr) %*% y_tr
y_hat_test_Bayes <- X_test %*% beta_Bayes
mspe_Bayes <- mean((y_hat_test_Bayes - y_test)^2)
```

Using the posterior mean $\hat{\boldsymbol{\beta}}_{Bayes}$ obtained from the training data, the predicted values for the test data have been computed, giving me a MSPE of 70.8519766 percent.

I will now visualize the predicted crime rates versus the true crime rates for the test set like before.

```r
plot(y_test, y_hat_test_Bayes,
     xlab = expression(y[test]),
     ylab = expression(hat(y)[test]),
     ylim = c(-4,4),
     main = "Test data Bayesian g-prior predicted vs. true crime rates")
abline(0, 1)
```



The error is slightly lower for the Bayesian g-prior model compared to the OLS model. The g-prior acts as a form of regularization by shrinking the OLS coefficients towards zero (since $\hat{\boldsymbol{\beta}}_{Bayes} = \frac{g}{g+1}\hat{\boldsymbol{\beta}}_{ols}$), which effectively reduces the model's complexity and guards against overfitting the training data. A lower MSPE on the test set indicates better generalization to unseen data. In terms of the plot, the points move closer to the ideal line where predicted and true values are the same, but there is no significant change, which is consistent with the still high MSPE value. In fact, it is difficult to notice a change between the two plots. So while this model is better, it is still not a great model for accurate predictions.
