Introduction to Bayesian Regression

Hanson Egbert Dmitriy Timokhin Mike Halversion

CONTENTS

1. INTRODUCTION

Here we will be exploring Bayesian regression, and how it differs from a frequentist linear regression approach. We will explain how Bayesian regression works and will provide an analysis using the approach on a sample dataset. This includes information on a conjugate prior distribution as well as how to get the posterior distribution from it. Furthermore, we will provide extensive notes on how we did our analysis and our knowledge of Bayesian regression, as well as code for it in R.

2. BODY

Bayesian regression can take on a number of prior distributions, both for σ^2 and for initializing the regression coefficients, $\vec{\beta}$. Selecting these prior distributions could be the subject of a lengthier project, but for the purposes of this exercise we will focus on one commonly used choice for the prior distributions, what is called the Normal Inverse gamma (NIG) distribution.

a. THE NIG (NORMAL INVERSE GAMMA) DISTRIBUTION

The primary advantage of using the NIG for regression is that we have a conjugate prior with an analytically solvable posterior distribution. With these assumptions for the prior come the normal assumptions regarding the variables; we still assume independence between variables, and we still assume that the relationship between the independent and dependent variables are linear. This will reduce our hard math and allow us to just follow the following equations in order to solve for the regression:

- Prior joint distribution for (β, σ^2) in two steps:
  - $\sigma^2 \sim \text{Inverse-Gamma}(a_0, b_0)$
  - $\vec{\beta} \mid \sigma^2 \sim \text{MultivariateNormal}(\vec{\mu}_\beta, \sigma^2 V_\beta)$
- Posterior joint distribution for (β, σ^2) in two steps:
  - $\sigma^2 \mid \text{data} \sim \text{Inverse-Gamma}(a_1, b_1)$
    - $a_1 = a_0 + n/2$
    - $b_1 = b_0 + (\vec{\mu}_\beta^\top V_\beta^{-1} \vec{\mu}_\beta + y^\top y - \vec{\mu}_{\beta 1}^\top V_{\beta 1}^{-1} \vec{\mu}_{\beta 1})$

- $\vec{\beta} \mid \text{data}, \sigma^2 \sim \text{MultivariateNormal}(\vec{\mu}_\beta, \sigma^2 V_\beta)$
  - $\vec{\mu}_{\beta 1} = (V_\beta^{-1} + X^\top X)^{-1}(V_\beta^{-1}\ \vec{\mu}_\beta + X^\top y)$
  - $V_{\beta 1} = (V_\beta^{-1} + X^\top X)^{-1}$

- Hyperparameters are a_0, b_0, $\vec{\mu}_\beta$, and V_β
- a_0 and b_0 indicate how certain we are in our predictions for other hyperparameters (with a_1 and b_1 being their updates)
- $\vec{\mu}_\beta$ indicates our best prior guess for $\vec{\beta}$
- V_β indicates our best prior guess for the covariance matrix for the variables in $\vec{\beta}$
- $\vec{\beta}$ is a vector with our slope coefficients for our model

With the posterior NIG distribution we are able to generate and make inferences on $\vec{\beta}$. Something a frequentist would have trouble doing at all. Let's see how we have implemented this in our R code.

b. NIG Distribution In R

```
numberOfSims=1
numberOfVariables=3
specifiedVariables=c("Intercept","MIN","PTS")
response="REB"
a0=.1
b0=.001
file="/Users/Dima/Desktop/DataSets/players_stats.csv"
```

This first section of code simply specifies our number of variables that we are including in the regression, the explanatory variables, the response variable, and the Inverse Gamma distribution hyperparameters. The number of sims is set to 1 to make the linear algebra work later on. Furthermore we read in the data.

```
# initialization
sds=rinvgamma(numberOfSims,a0,b0)
meanguess=runif(numberOfVariables)
# make sure invertible covariance
covariance0=matrix(c(2,0,0,0,5,0,0,0,5),numberOfVariables,numberOfVariables)
#covariance0=cov(data[,specifiedVariables])
betaorig=mvrnorm(numberOfSims,meanguess,sds*covariance0)
data=read_csv(file)
data$Intercept=1
```

This next section of code, creates the prior distributions we will need for this analysis. σ^2 is generated as expected from our choice of a_0 and b_0. Since we have no prior information regarding what the means of the variables are, we initialize them to a sample from the uniform between 0 and 1. If information was provided, we could initialize them to a more suitable guess. The covariance is set so the betas vary, but they do not vary together, although this is easily changeable depending on the data in question. This is mainly done so the covariance matrix is easily invertible, which is necessary for calculating the posteriors.

```
# posteriors
theta1=solve(solve(covariance0)+t(as.matrix(data[specifiedVariables]))%*%as.matrix(data[,specifiedVariables]))%*%
  (solve(covariance0)%*%betaorig+
     t(as.matrix(data[,specifiedVariables]))%*%as.matrix(data[,response]))
covariance1=solve((solve(covariance0)+t(as.matrix(data[specifiedVariables]))
                   %*%as.matrix(data[,specifiedVariables])))
a1=a0+.5*nrow(data)
b1=b0+.5*(t(betaorig)%*%solve(covariance0)%*%betaorig+as.matrix(t(data[,response]))%*%as.matrix(data[,response])-
            t(theta1)%*%solve(covariance1)%*%theta1)
```

Here we calculate the posterior distribution using the equations which were solved analytically[1, Elster]. This is done to achieve the multivariate normal and inverse gamma distributions from which we will be sampling from in order to complete our analysis. Notice that the covariance has its inverse taken several times, which is the reason we set the covariance guess to be something simple.

```
# metrics
predictions=t(theta1)%*%t(as.matrix(data[,specifiedVariables]))
MSE=(1/nrow(data))*sum((predictions-data[,response])^2)
R2=1-sum((predictions-data[,response])^2)/(sum((data[,response]-mean(as.matrix(data[,response])))^2))

# comparison to actual linear regression
fit<-lm(as.formula(paste(response,"~",paste(specifiedVariables[specifiedVariables!="Intercept"],collapse = "+"),sep="")),data=data)
summary(fit)$r.squared;R2
mean(fit$residuals^2);MSE
fit$coefficients;theta1
```

Here we make predictions using our model, and then calculate common metrics to compare bayesian regression to a frequentist approach. We compare R^2 , MSE, and the coefficients of both the analyses. The output of these results is shown later.

```
# confidence intervals
sims1=10000
sds1=rinvgamma(sims1,a1,b1)
mean(sds1);sort(sds1)[.025*sims1];sort(sds1)[.975*sims1]
Intercept = rep(0, 10000)
MIN = rep(0, 10000)
PTS = rep(0, 10000)
betafinal = data_frame(Intercept, MIN, PTS)
for(j in 1:length(sds1)){
  out = mvrnorm(1,theta1,sds1[j]*covariance1)
  betafinal$Intercept[j] = out[1]
  betafinal$MIN[j] = out[2]
  betafinal$PTS[j] = out[3]
}

for (name in 1:length(colnames(betafinal))){
  message(colnames(betafinal)[name])
  message("Mean: ",round(mean(betafinal[[name]]), 4))
  message("SD: ", round(sd(betafinal[[name]]), 4))
  message("95% Credible Interval: ", "(",round(sort(betafinal[[name]])[.025*sims1], 4),
          ", ",round(sort(betafinal[[name]])[.975*sims1], 4), ")" )

}
```

This portion of code gets 10,000 randomly generated posterior σ^2 values and coefficients from the multivariate normal distribution using those values to create mean, sd, and credible intervals for each of the variables we are looking at. In this case it is the intercept, average minutes played, and average points scored.

c. NBA (National Basketball Association) Regression

Dataset:

NBA Player stats from 2014-2015 season

https://www.kaggle.com/drgilermo/nba-players-stats-20142015/data

i. Univariate Analysis

In this univariate analysis we will be focusing on predicting rebounds per minute from a given players height. We will believe that height is a very strong predictor of rebounds and thus our regression may prove very powerful.

```
covariance0=matrix(c(5,1,1,5),numberOfVariables)
```

This covariance was selected after a variety of other, simpler ones were tried. The others generated predictions that were far off from what we would anticipate, including negative R squared values. This matrix, as we will see later, generates realistic results.

```
> summary(fit)$r.squared;R2
[1] 0.4189254
[1] 0.4123019
> mean(fit$residuals^2);MSE
[1] 0.00524804
[1] 0.005307861
> fit$coefficients;theta1
 (Intercept)        Height
-1.214123646  0.007045995
                     REBM
Intercept -1.039027173
Height     0.006160899
```

With this results we can see we roughly saw the same Mean squared error and R squared value, although slightly worse. The coefficients are slightly different, likely due to our selection of the priors.

```
> mean(sds1);sort(sds1)[.025*sims1];sort(sds1)[.975*sims1]
[1] 0.005684212
[1] 0.004965365
[1] 0.006500182
```

We can see our σ^2 is small, indicating that our model has very little variance in it.

```
> confint(fit, level = .95)
                  2.5 %       97.5 %
(Intercept) -1.37142323 -1.056824064
Height       0.00625008  0.007841909
```

These are the 95% confidence intervals we calculated for the frequentist approach. We are 95% confident that a player with a height of zero will on average have between -1.3714 and -1.0568 rebounds per minute. This negative interval isn't meaningful as nobody in the NBA has a height of zero inches.. Furthermore we are 95% confident that each one inch increase of height is associated with an increase of rebounds per minute of between 0.00625 and 0.00784.

```
Intercept
Mean: -1.0394
SD: 0.0741
95% Credible Interval: (-1.1808, -0.8919)
Height
Mean: 0.0062
SD: 4e-04
95% Credible Interval: (0.0054, 0.0069)
```

On average for our many samples of the Intercept and Height, the intercept coefficient was -1.0394, and the height coefficient is .0062. However, there is a 95% chance that the true intercept of $\vec{\beta}$ is between -1.18 and -.89. Furthermore there is a 95% chance that the true coefficient of Height is between 0.0054 and 0.0069. Since both intervals don't contain zero, we conclude that both the Intercept and Height coefficients are significant in predicting rebounds per minute. The widths of the two sets of intervals (frequentist and Bayesian) are roughly the same, indicating our Bayesian regression is as confident as our frequentist regression.

Since the coefficients are roughly the same, we can say the assumptions about the residuals are satisfied as long as they are for the standard frequentist linear regression.

ii. Multivariate Analysis

This NBA dataset contains many statistics on players, but we will be focusing on REB (number of rebounds in a season), MIN (number of minutes played in a season), and PTS (number of points scored in a season). We believe that using PTS and MIN we can predict REB fairly accurately for a player.

```
> summary(fit)$r.squared;R2
[1] 0.5851916
[1] 0.5851916
> mean(fit$residuals^2);MSE
[1] 15257.73
[1] 15257.73
> fit$coefficients;theta1
(Intercept)         MIN         PTS
-1.54210599  0.18900411 -0.02129187
                 REB
Intercept -1.53434850
MIN        0.18899708
PTS       -0.02128575
```

Here we see a comparison of the R^2, MSE, and the coefficients of the two different approaches. The first value outputted is the Frequentist approach and the second value outputted is the Bayesian approach. From inspection it is clear that the two do not differ by much with the R^2 and the MSE being identical and the coefficients being almost identical.

```
> mean(sds1);sort(sds1)[.025*sims1];sort(sds1)[.975*sims1]
[1] 15310.96
[1] 13474.05
[1] 17341.17
```

This output represents the Mean, and the 95% credible interval for the posterior σ^2 values.

```
> confint(fit, level = .95)
                   2.5 %     97.5 %
(Intercept) -21.79334180 18.7091298
MIN           0.15473119  0.2232770
PTS          -0.08792155  0.0453378
```

These are the 95% confidence intervals we calculated for the frequentist approach. We are 95% confident that with zero average minutes played and zero average points scored that on average the number of rebounds for that season is between -21.79 and 18.7. Furthermore we are 95% confident that each one unit increase in average minutes played is associated with an increase in rebounds for the season by between 0.15473 and 0.2232. Finally we are 95% confident that each one unit increase of average points per game is associated with between a decrease of 0.0879 and an increase of 0.045 in rebounds for the season.

```
Intercept
Mean: -1.5867
SD: 10.2414
95% Credible Interval: (-21.6528, 18.4974)
MIN
Mean: 0.1889
SD: 0.0174
95% Credible Interval: (0.1551, 0.2228)
PTS
Mean: -0.021
SD: 0.0337
95% Credible Interval: (-0.0873, 0.0446)
```

This output shows means, standard deviations, and 95% credible intervals for each coefficient in our slope coefficient vector $\vec{\beta}$. To interpret the output for Intercept, on average our generated $\vec{\beta}$'s have an intercept value of -1.59, with a standard deviation of 10.24 meaning that these generated intercepts are not very consistent around -1.59. However, there is a 95% chance that the true intercept of $\vec{\beta}$ is between -21.65 and 18.48. Similarly, there is a 95% chance that the coefficient for minutes is between .1551 and .2228, and there is a 95% chance that the coefficient for points is between -.0873 and .0446. Given these interval, we can say that points is not a significant predictor, while minutes played is significant. The 95% credible interval is actually smaller in width than the 95% confidence interval, indicating that our Bayesian regression is slightly more confident than the frequentist interval.

Since the coefficients are roughly the same, we can say the assumptions about the residuals are satisfied as long as they are for the standard frequentist linear regression.

METHOD NOTES:

One of the weaknesses observed with the method is its reliance on selection of the covariance matrix and the inverse gamma distribution parameters. These variables are the most 'arbitrary' in their selection of the prior, but can have a great deal of influence, most notably on the credible intervals width, although on the coefficients themselves for certain selections. The choice of the initial guess seems to be less sensitive, although it does have some implications for the future coefficients as well. Setting the guess of the means to the uniform from other ranges typically lead to the same coefficients as our usual starting guess of 0 to 1, but in certain runs can lead to much worse predictions.

Our prediction errors were not fantastic, primarily because we were focused on illustrating the method rather than performing model selection procedures. For slightly better results, we could have standardized the predictor and response variables, which is known to be useful for Bayesian regression. Given our results, we can likely say that we would end up with similar results to linear regression if a method such as forward stepwise or backwards stepwise was attempted.

3.CONCLUSION

Bayesian regression, just as the vast majority of our Bayesian approaches, delivered the same or comparable methods to the frequentist approach, linear regression. So we end up with similar advantages for this new method as we did with our other Bayesian approaches, namely that our results are easier to

interpret and easier to explain to non-statisticians. In addition to that advantage, the ability to parametrize with various distributions initially, although not shown here, is something not as easily provided by the frequentist approach. A use case where Bayesian regression would do outright better is where the statistician has strong prior beliefs about what the relationship should be between the variables, and the sample size for the data is small. In this case, a strong prior belief is an effective counter to over relying on the small amount of data, and thus, can prevent from overfitting. This strong prior belief may come from previous research, or from subject matter knowledge. No matter what the reason, Bayesian regression allows for the statistician to build off this knowledge in a way that frequentist regression never could. Overall, Bayesian regression is likely better for performing regression on small amounts of data, and roughly even for regression on large datasets, making it a better all-around tool than the frequentist approach.

4.Appendices

Full R Code:

```
library(invgamma)
library(MASS)
library(tidyverse)
# set up
numberOfSims=1
numberOfVariables=3
specifiedVariables=c("Intercept","MIN","PTS")
response="REB"
a0=.1
b0=.001
file="/Users/Dima/Desktop/DataSets/players_stats.csv"
# initialization
sds=rinvgamma(numberOfSims,a0,b0)
meanguess=runif(numberOfVariables)
# make sure invertible covariance
covariance0=matrix(c(2,0,0,0,5,0,0,0,5),numberOfVariables,numberOfVariables)
betaorig=mvrnorm(numberOfSims,meanguess,sds*covariance0)
data=read_csv(file)
data$Intercept=1
# posteriors
theta1=solve(solve(covariance0)+t(as.matrix(data[specifiedVariables]))%*%as.matrix(data[,speci
fiedVariables]))%*%
  (solve(covariance0)%*%betaorig+
     t(as.matrix(data[,specifiedVariables]))%*%as.matrix(data[,response]))
covariance1=solve((solve(covariance0)+t(as.matrix(data[specifiedVariables]))
                   %*%as.matrix(data[,specifiedVariables])))
a1=a0+.5*nrow(data)
b1=b0+.5*(t(betaorig)%*%solve(covariance0)%*%betaorig+as.matrix(t(data[,response]))%*%as.matri
x(data[,response])-
            t(theta1)%*%solve(covariance1)%*%theta1
# metrics
predictions=t(theta1)%*%t(as.matrix(data[,specifiedVariables]))
MSE=(1/nrow(data))*sum((predictions-data[,response])^2)
R2=1-sum((predictions-data[,response])^2)/(sum((data[,response]-mean(as.matrix(data[,response]
)))^2)
# comparison to actual linear regression
fit<-lm(as.formula(paste(response,"~",paste(specifiedVariables[specifiedVariables!="Intercept"
],collapse = "+"),sep="")),data=data)
summary(fit)$r.squared;R2
mean(fit$residuals^2);MSE
fit$coefficients;theta1
# confidence intervals
sims1=10000
sds1=rinvgamma(sims1,a1,b1)
```

```
mean(sds1);sort(sds1)[.025*sims1];sort(sds1)[.975*sims1]
Intercept = rep(0, 10000)
MIN = rep(0, 10000)
PTS = rep(0, 10000)
betafinal = data_frame(Intercept, MIN, PTS)
for(j in 1:length(sds1)){
  out = mvrnorm(1,theta1,sds1[j]*covariance1)
  betafinal$Intercept[j] = out[1]
  betafinal$MIN[j] = out[2]
  betafinal$PTS[j] = out[3]
}
for (name in 1:length(colnames(betafinal))){
  message(colnames(betafinal)[name])
  message("Mean: ",round(mean(betafinal[[name]]), 4))
  message("SD: ", round(sd(betafinal[[name]]), 4))
  message("95% Credible Interval: ", "(",round(sort(betafinal[[name]])[.025*sims1], 4),
          ", ",round(sort(betafinal[[name]])[.975*sims1], 4), ")" )

}
```

TIME LOG:

Date:	Member:	Time(minutes):	Activity:
10/27/2017	Entire team	120	Topic/Proposal
10/28/207	Entire team	180	Finalize topic and begin researching a dataset and reading about conjugate priors for bayesian regression.
12/1/2017 - 12/2/2017	Entire team	300	Researching bayesian regression.
12/3/2017	Entire team	240	Begin implementing code for analysis
12/4/2017	Dmitriy T, Hanson E	120	Add simulation credible intervals to code and debug .
12/4/2017	Michael halverson	80	Debug beta1 calculations and remove unnecessary code.
12/5/2017	Michael H	30	Add posterior Sd's to credible intervals.
12/5/2017	Michael H	30	Found suitable starting guess for alpha/beta, fixed covariance.
12/5/2017	Dmitriy T, Hanson E	60	Work on write up and formatting of report.
12/6/2017	Michael H	60	Check over report and provide feedback to other members write up.

REFERENCES:

[1] C. Elster, K. Klauenberg, M. Walzel, G. W¨ubbeler, P. Harris, M. Cox, C. Matthews, I. Smith, L. Wright, A. Allard, N. Fischer, S. Cowen, S. Ellison, P. Wilson, F. Pennecchi, G. Kok, A. van der Veen, L. Pendrill, A Guide to Bayesian Inference for Regression Problems, Deliverable of EMRP project NEW04 "Novel mathematical and statistical approaches to uncertainty evaluation", 2015.

[2] Reich, Brian. "Bayesian Linear Regression." NC State University, NC State University, 2017, www4.stat.ncsu.edu/~reich/ABA/notes/BLR.pdf.