Applied Bayes Project

Cambrey Sullivan

11/23/2019

# The Data

The data is from the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients). It follows 30,000 lines of credit in Taiwan over a 5 month period and records which default and which don’t. There are 23 explanatory variables:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.

X2: Gender (1 = male; 2 = female).

X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others). X4: Marital status (1 = married; 2 = single; 3 = others).

X5: Age (year).

X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; …; X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.

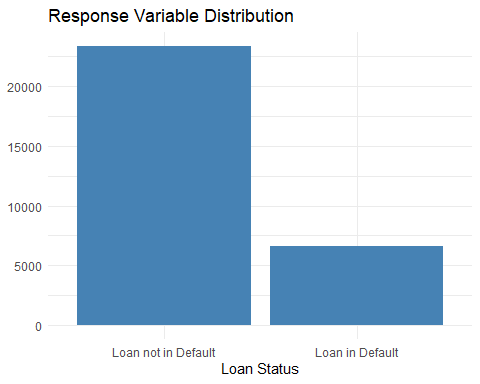
X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; …; X17 = amount of bill statement in April, 2005.

X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; …; X23 = amount paid in April, 2005.

The main objective of this project is to compare a full frequentist logistic regression results to a Bayesian Logistic regression using smaller sample size from the data.

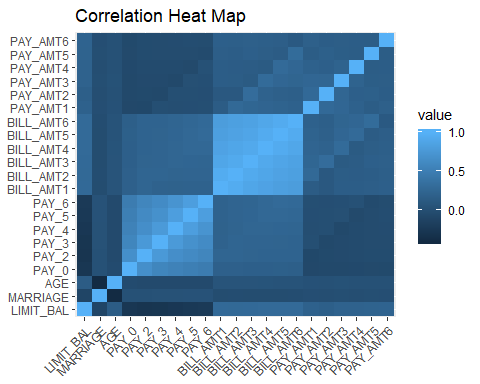
## The Response Variable

As we see in the bar chart, defaulters constitute 6,636 individuals of the total 30,000 individuals included in the data set. This accounts for 22.12% of all customers. The non-proportionality of classes means that assessing the performance of models solely based on test accuracy could be misleading. For example, a model that predicts all test observations as zero has a 78% accuracy rate. However, this does not mean that model is performing well. With this as a backdrop, we tried to assess each individual model based on its performance on a list of statistics besides accuracy.



## The Explanatory Variables

One of the common problems associated with explanatory variables is a problem of multicollinearity. The correlation heat map shows variables that are highly correlated with other variables in light blue. The variables with the highest correlation seem to be the bill amount for the month. This makes sense. Most of the time, credit lenders divide the principle amount into equal parts for payment.



We found that the bill amount variables (X\_12-X\_17) were highly correlated with one another. The following table of variables indicate their mutual correlation coefficients are greater than 0.9 in absolute value.

## row col  
## BILL\_AMT2 11 10  
## BILL\_AMT1 10 11  
## BILL\_AMT3 12 11  
## BILL\_AMT2 11 12  
## BILL\_AMT4 13 12  
## BILL\_AMT3 12 13  
## BILL\_AMT5 14 13  
## BILL\_AMT6 15 13  
## BILL\_AMT4 13 14  
## BILL\_AMT6 15 14  
## BILL\_AMT4 13 15  
## BILL\_AMT5 14 15

# Frequentist Logistic Regression

Given the explanatory variables, logistic regression assumes that each response variable follows a Bernoulli distribution with probability of

Since the link function for is a logistic distribution, finding the coefficients of the variables involve maximizing the following distribution with respect to the coefficients:

We will train the model using a data set of 70% of the 30,000 observations.

##   
## Call:  
## glm(formula = default ~ ., family = binomial, data = d.train.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.1704 -0.7006 -0.5454 -0.2881 3.9893   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.653e-01 1.421e-01 -3.978 6.94e-05 \*\*\*  
## LIMIT\_BAL -6.728e-07 1.880e-07 -3.579 0.000345 \*\*\*  
## SEX -1.054e-01 3.669e-02 -2.872 0.004078 \*\*   
## EDUCATION -1.139e-01 2.515e-02 -4.528 5.95e-06 \*\*\*  
## MARRIAGE -1.806e-01 3.804e-02 -4.747 2.06e-06 \*\*\*  
## AGE 5.912e-03 2.139e-03 2.764 0.005716 \*\*   
## PAY\_0 5.694e-01 2.115e-02 26.928 < 2e-16 \*\*\*  
## PAY\_2 8.339e-02 2.405e-02 3.467 0.000526 \*\*\*  
## PAY\_3 8.109e-02 2.705e-02 2.998 0.002714 \*\*   
## PAY\_4 4.610e-02 2.992e-02 1.541 0.123354   
## PAY\_5 2.302e-02 3.233e-02 0.712 0.476350   
## PAY\_6 1.281e-02 2.636e-02 0.486 0.627027   
## BILL\_AMT1 -5.772e-06 1.360e-06 -4.243 2.21e-05 \*\*\*  
## BILL\_AMT2 2.341e-06 1.801e-06 1.300 0.193636   
## BILL\_AMT3 2.185e-06 1.628e-06 1.342 0.179738   
## BILL\_AMT4 -9.751e-07 1.667e-06 -0.585 0.558485   
## BILL\_AMT5 2.777e-06 1.765e-06 1.573 0.115680   
## BILL\_AMT6 -1.869e-06 1.380e-06 -1.354 0.175602   
## PAY\_AMT1 -1.441e-05 2.837e-06 -5.079 3.80e-07 \*\*\*  
## PAY\_AMT2 -9.022e-06 2.428e-06 -3.716 0.000203 \*\*\*  
## PAY\_AMT3 -2.333e-06 2.072e-06 -1.126 0.260224   
## PAY\_AMT4 -7.570e-06 2.475e-06 -3.059 0.002223 \*\*   
## PAY\_AMT5 -1.212e-06 1.992e-06 -0.608 0.543021   
## PAY\_AMT6 -3.122e-06 1.611e-06 -1.938 0.052598 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 22228 on 20999 degrees of freedom  
## Residual deviance: 19488 on 20976 degrees of freedom  
## AIC: 19536  
##   
## Number of Fisher Scoring iterations: 5

We can see that the logistic regression on the training data set has numerous coefficients which are not statistically significant (p-value >0.05) as seen in table 3. These exceptions are likely due to the multicollinearity present in the billing amount variables.

## Confusion Matrix

As seen in the table below, nearly a third of defaulters, 200 of the total 657 in the test data set, are missclassified as non defaulters. We are more interested in the misclassifications of defaulters than non-defaulters, since the response variable is not proportional. As such, this method of classification on the given data set may not be the best method of classification. Next we’ll look at Bayesian methods to see how they compare to Frequentist methods.

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 6823 200  
## 1 1520 457  
##   
## Accuracy : 0.8089   
## 95% CI : (0.8006, 0.817)  
## No Information Rate : 0.927   
## P-Value [Acc > NIR] : 1   
##   
## Kappa : 0.2666   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Sensitivity : 0.8178   
## Specificity : 0.6956   
## Pos Pred Value : 0.9715   
## Neg Pred Value : 0.2312   
## Prevalence : 0.9270   
## Detection Rate : 0.7581   
## Detection Prevalence : 0.7803   
## Balanced Accuracy : 0.7567   
## 'Positive' Class : 0

# Bayesian Logistic Regression

We use the same logistic model as the Frequentist approach above. However, rather than training on 70% of the data and testing on 30%, we will only use 300 observations. We did this for time constraint purposes and because we were interested in how few observations were required using Bayesian methods to obtain similar results to the Frequentist model.

For this project, we used JAGS in R to fit the Bayesian logistic model. We use MCMC to estimate the posterior distribution by drawing samples. The model description along with the likelihood and priors can be seen in the code below. We used a standard deviation value of 100 for quick convergence and in order that the priors be non-informative.

n <- length(samp.data$default)  
bin.col<-c(2:4, 24)  
norm.col<-c(1, 5:23)  
   
 logistic\_model <- "model{  
  
 # Likelihood  
  
 for(i in 1:n){  
 Y[i] ~ dbern(q[i])  
 logit(q[i]) <- beta[1] + beta[2]\*X[i,1] + beta[3]\*X[i,2] + beta[4]\*X[i,3] + beta[5]\*X[i,4] +

beta[6]\*X[i,5] + beta[7]\*X[i,6] + beta[8]\*X[i,7] + beta[9]\*X[i,8] + beta[10]\*X[i,9] +

beta[11]\*X[i,10] + beta[12]\*X[i,11] + beta[13]\*X[i,12] + beta[14]\*X[i,13] +

beta[15]\*X[i,14] + beta[16]\*X[i,15] + beta[17]\*X[i,16] + beta[18]\*X[i,17] +

beta[19]\*X[i,18] + beta[20]\*X[i,19] + beta[21]\*X[i,20] + beta[22]\*X[i,21] +

beta[23]\*X[i,22] + beta[24]\*X[i,23]  
 }  
  
 #Priors  
  
 for(j in 1:24){  
 beta[j] ~ dnorm(0,1000)  
 }  
 }"

library(rjags)

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

dat <- list(Y=samp.data$default,n=n,X=samp.data[ -c(24) ])  
 model <- jags.model(textConnection(logistic\_model),data = dat,n.chains=3, quiet=TRUE)  
  
   
 update(model, 10000, progress.bar="gui")  
  
 samp <- coda.samples(model,   
 variable.names=c("beta"),   
 n.iter=20000, progress.bar="gui")

## Diagnostics

### Summary Statistics

The following table is the JAGS output for the Bayesian model. It provides coefficient means, standard deviations, standard errors, and parameter quantiles.

## Iterations = 11001:31000  
## Thinning interval = 1   
## Number of chains = 3   
## Sample size per chain = 20000   
##   
## 1. Empirical mean and standard deviation for each variable,  
## plus standard error of the mean:  
##   
## Mean SD Naive SE Time-series SE  
## beta[1] -5.782e-04 3.164e-02 1.292e-04 1.674e-04  
## beta[2] -3.259e-06 1.702e-06 6.948e-09 1.640e-08  
## beta[3] -5.300e-03 3.130e-02 1.278e-04 1.722e-04  
## beta[4] 2.212e-04 3.133e-02 1.279e-04 1.833e-04  
## beta[5] -9.971e-04 3.102e-02 1.266e-04 1.709e-04  
## beta[6] -4.423e-03 7.734e-03 3.157e-05 8.343e-05  
## beta[7] 2.170e-02 3.103e-02 1.267e-04 1.649e-04  
## beta[8] 2.505e-02 3.085e-02 1.260e-04 1.610e-04  
## beta[9] 1.061e-02 3.075e-02 1.255e-04 1.625e-04  
## beta[10] 1.339e-02 3.149e-02 1.286e-04 1.686e-04  
## beta[11] 9.531e-03 3.130e-02 1.278e-04 1.646e-04  
## beta[12] 1.524e-02 3.131e-02 1.278e-04 1.655e-04  
## beta[13] -3.042e-05 2.844e-05 1.161e-07 2.204e-06  
## beta[14] -8.349e-06 4.638e-05 1.894e-07 5.464e-06  
## beta[15] 6.136e-05 4.086e-05 1.668e-07 4.457e-06  
## beta[16] -1.835e-05 3.345e-05 1.365e-07 2.936e-06  
## beta[17] -6.374e-05 4.506e-05 1.840e-07 4.698e-06  
## beta[18] 8.698e-05 4.239e-05 1.731e-07 4.160e-06  
## beta[19] 1.457e-05 2.279e-05 9.306e-08 6.618e-07  
## beta[20] -3.780e-04 1.327e-04 5.417e-07 1.977e-06  
## beta[21] 9.164e-05 5.043e-05 2.059e-07 2.290e-06  
## beta[22] -1.036e-04 7.311e-05 2.985e-07 1.065e-06  
## beta[23] -1.782e-04 7.879e-05 3.217e-07 2.182e-06  
## beta[24] -2.683e-04 1.203e-04 4.911e-07 1.415e-06  
##   
## 2. Quantiles for each variable:  
##   
## 2.5% 25% 50% 75% 97.5%  
## beta[1] -6.285e-02 -2.203e-02 -5.135e-04 2.090e-02 6.136e-02  
## beta[2] -6.739e-06 -4.382e-06 -3.211e-06 -2.081e-06 -4.796e-08  
## beta[3] -6.751e-02 -2.615e-02 -5.201e-03 1.577e-02 5.582e-02  
## beta[4] -6.106e-02 -2.094e-02 2.555e-04 2.127e-02 6.196e-02  
## beta[5] -6.193e-02 -2.190e-02 -9.808e-04 2.000e-02 5.926e-02  
## beta[6] -1.961e-02 -9.617e-03 -4.416e-03 8.026e-04 1.069e-02  
## beta[7] -3.986e-02 8.562e-04 2.174e-02 4.281e-02 8.227e-02  
## beta[8] -3.528e-02 4.237e-03 2.522e-02 4.597e-02 8.496e-02  
## beta[9] -4.955e-02 -1.022e-02 1.046e-02 3.140e-02 7.124e-02  
## beta[10] -4.761e-02 -7.865e-03 1.341e-02 3.466e-02 7.518e-02  
## beta[11] -5.187e-02 -1.157e-02 9.746e-03 3.059e-02 7.059e-02  
## beta[12] -4.574e-02 -5.819e-03 1.514e-02 3.626e-02 7.656e-02  
## beta[13] -8.861e-05 -4.920e-05 -2.902e-05 -1.148e-05 2.304e-05  
## beta[14] -1.054e-04 -3.949e-05 -6.563e-06 2.487e-05 7.551e-05  
## beta[15] -1.674e-05 3.286e-05 6.011e-05 8.914e-05 1.414e-04  
## beta[16] -8.656e-05 -4.029e-05 -1.725e-05 4.659e-06 4.382e-05  
## beta[17] -1.484e-04 -9.624e-05 -6.225e-05 -3.084e-05 1.905e-05  
## beta[18] 8.584e-06 5.656e-05 8.609e-05 1.177e-04 1.701e-04  
## beta[19] -2.798e-05 -7.275e-07 1.363e-05 2.893e-05 6.192e-05  
## beta[20] -6.615e-04 -4.649e-04 -3.696e-04 -2.824e-04 -1.439e-04  
## beta[21] -1.166e-05 5.924e-05 9.278e-05 1.253e-04 1.881e-04  
## beta[22] -2.715e-04 -1.476e-04 -9.210e-05 -4.865e-05 4.980e-06  
## beta[23] -3.583e-04 -2.252e-04 -1.683e-04 -1.222e-04 -5.037e-05  
## beta[24] -5.279e-04 -3.443e-04 -2.571e-04 -1.808e-04 -6.792e-05

Additionally, we calculated the odds for each variable and reported them below.

## LIMIT\_BAL SEX EDUCATION MARRIAGE AGE PAY\_0 PAY\_2 PAY\_3   
## 0.9999967 0.9952009 1.0007124 0.9994841 0.9956163 1.0224289 1.0258516 1.0111447

## PAY\_4 PAY\_5 PAY\_6 BILL\_AMT1 BILL\_AMT2 BILL\_AMT3 BILL\_AMT4 BILL\_AMT5   
## 1.0139827 1.0100714 1.0158497 0.9999696 0.9999917 1.0000614 0.9999816 0.9999363

## BILL\_AMT6 PAY\_AMT1 PAY\_AMT2 PAY\_AMT3 PAY\_AMT4 PAY\_AMT5 PAY\_AMT6   
## 1.0000870 1.0000146 0.9996220 1.0000916 0.9998964 0.9998218 0.9997318

## LIMIT\_BAL SEX EDUCATION MARRIAGE AGE PAY\_0   
## 1.701900e-06 3.114412e-02 3.137066e-02 3.100468e-02 7.699679e-03 3.171668e-02

## PAY\_2 PAY\_3 PAY\_4 PAY\_5 PAY\_6 BILL\_AMT1   
## 3.164525e-02 3.111040e-02 3.194549e-02 3.160951e-02 3.181997e-02 2.843696e-05

## BILL\_AMT2 BILL\_AMT3 BILL\_AMT4 BILL\_AMT5 BILL\_AMT6 PAY\_AMT1   
## 4.638203e-05 4.085933e-05 3.344651e-05 4.505801e-05 4.239828e-05 2.279415e-05

## PAY\_AMT2 PAY\_AMT3 PAY\_AMT4 PAY\_AMT5 PAY\_AMT6   
## 1.326410e-04 5.043021e-05 7.309907e-05 7.877204e-05 1.202536e-04

### Effective Sample Size

We also calculated the effective sample sizes of the parameters to look at the correlation between the chains. These are reported below.

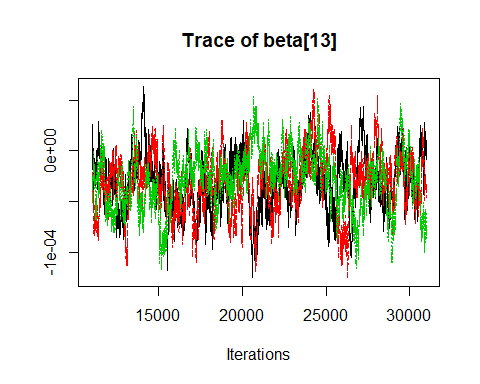
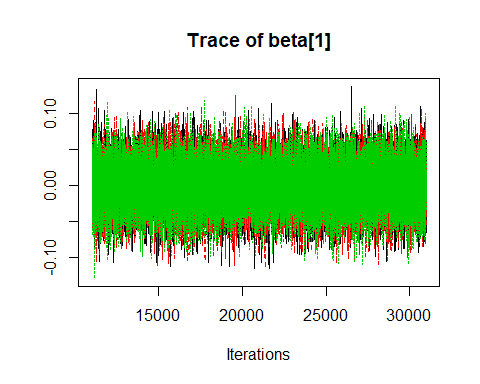
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6]   
## 35766.77110 10772.59370 33063.19454 29265.13207 32942.35530 8594.70882   
## beta[7] beta[8] beta[9] beta[10] beta[11] beta[12]   
## 35477.20355 36778.34149 35886.44443 34917.27847 36192.83726 35895.96871   
## beta[13] beta[14] beta[15] beta[16] beta[17] beta[18]   
## 167.79738 72.00857 102.14010 143.17001 91.84546 104.26328   
## beta[19] beta[20] beta[21] beta[22] beta[23] beta[24]   
## 1248.13526 4531.38326 551.14075 5634.89405 1318.68947 7287.93868

Importantly the highly correlated variables have a low ESS and many of the other variables approach the sample size used in the Frequentist approach. Since some of the estimates are quite low (<100), we would prefer to run more iterations or remove/transform these variables, as this may be indicative of poor mixing.

In order to look at the mixing of the MCMC, we next looked at trace plots for each of the parameters.

### Trace Plots

Most of the trace plots looked like the graph on the bottom left and indicated good mixing. However, beta[13]-beta[18] looked more like the plot on the bottom right suggesting that they didn’t mix as well. In the interest of space, we only showed 2 plots here.



### Autocorrelation Plots

We also looked at autocorrelation plots, which we won’t post here in the interest of brevity. There was high autocorrelation for the variables with high correlation with other variables. This is not concerning for the MCMC, and we used a burn in period anyway.

### Gelman Plots

Indeed, the Gelman plots didn’t suggest non-convergence. Again, in the interest of space, the plots won’t be shown here, but they are in the R code provided.

### Geweke Diagnostic

The Geweke diagnostic tests the hypothesis that the parameter means of the first 10% of the chain and the first 50% of the chain are independent. The z-scores for our MCMC are in the table below. Almost all of the absolute values, are within the absolute value threshold of 2.

## [[1]]  
##   
## Fraction in 1st window = 0.1  
## Fraction in 2nd window = 0.5   
##   
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]   
## 0.91233 -0.17934 -0.72388 0.35392 -0.66958 -0.98647 0.35196 -0.35647   
## beta[9] beta[10] beta[11] beta[12] beta[13] beta[14] beta[15] beta[16]   
## 0.19987 -1.85896 0.76745 -1.18356 -0.63948 0.42003 -0.30480 0.38106   
## beta[17] beta[18] beta[19] beta[20] beta[21] beta[22] beta[23] beta[24]   
## -0.07367 -0.10040 -2.14984 1.09781 -0.04846 2.39507 0.11812 0.29284   
##   
##   
## [[2]]  
##   
## Fraction in 1st window = 0.1  
## Fraction in 2nd window = 0.5   
##   
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]   
## 0.03300 -1.74630 0.05622 -1.18352 -1.37166 3.15625 -2.38243 -1.67122   
## beta[9] beta[10] beta[11] beta[12] beta[13] beta[14] beta[15] beta[16]   
## -1.41076 -0.95696 -1.14012 2.46398 -1.54327 0.93535 -0.09552 -0.15187   
## beta[17] beta[18] beta[19] beta[20] beta[21] beta[22] beta[23] beta[24]   
## -1.12456 1.52495 -0.40381 1.14526 0.76659 -0.68154 -1.92099 -0.12424   
##   
##   
## [[3]]  
##   
## Fraction in 1st window = 0.1  
## Fraction in 2nd window = 0.5   
##   
## beta[1] beta[2] beta[3] beta[4] beta[5] beta[6] beta[7] beta[8]   
## -1.55214 1.60810 0.35788 -3.81091 -0.58935 0.47596 -0.71928 1.24168   
## beta[9] beta[10] beta[11] beta[12] beta[13] beta[14] beta[15] beta[16]   
## -0.07367 1.13213 0.99177 1.88537 -1.52665 0.88411 0.16713 -1.82951   
## beta[17] beta[18] beta[19] beta[20] beta[21] beta[22] beta[23] beta[24]   
## -0.93514 1.07340 -1.76829 0.03904 2.76381 0.43846 -2.50845 -1.43074

# Results

The coefficient estimates are very different. Some of the signs are even opposite! However, it does look like all of the Frequentist Estimates are between in the 2.5 and 97.5 percentiles provided in the ‘Summary Statistics’ portion of this report.

The Bayesian standard deviations are smaller than the Frequentist standard error about half of the time. We are not surprised, as we used a MUCH smaller sample to generate the Bayesian estimates.

## Bayesian Means Frequentist Estimates Bayesian SD Frequentist SE  
## beta[1] -5.781996e-04 -5.652501e-01 3.164190e-02 1.420786e-01  
## beta[2] -3.258587e-06 -6.727624e-07 1.701906e-06 1.879662e-07  
## beta[3] -5.300391e-03 -1.053688e-01 3.129971e-02 3.668772e-02  
## beta[4] 2.212426e-04 -1.138833e-01 3.133106e-02 2.514977e-02  
## beta[5] -9.970565e-04 -1.805992e-01 3.101726e-02 3.804477e-02  
## beta[6] -4.423244e-03 5.912421e-03 7.734235e-03 2.139337e-03  
## beta[7] 2.169987e-02 5.694004e-01 3.102698e-02 2.114502e-02  
## beta[8] 2.504733e-02 8.338974e-02 3.085177e-02 2.405289e-02  
## beta[9] 1.061016e-02 8.109331e-02 3.074880e-02 2.704516e-02  
## beta[10] 1.338985e-02 4.609830e-02 3.149352e-02 2.991748e-02  
## beta[11] 9.531367e-03 2.302288e-02 3.129653e-02 3.232709e-02  
## beta[12] 1.523520e-02 1.280656e-02 3.130863e-02 2.635552e-02  
## beta[13] -3.042355e-05 -5.771643e-06 2.843788e-05 1.360414e-06  
## beta[14] -8.349407e-06 2.341543e-06 4.638262e-05 1.801327e-06  
## beta[15] 6.136440e-05 2.184798e-06 4.085678e-05 1.628549e-06  
## beta[16] -1.835404e-05 -9.750664e-07 3.344719e-05 1.666516e-06  
## beta[17] -6.374096e-05 2.777036e-06 4.506100e-05 1.765251e-06  
## beta[18] 8.698184e-05 -1.868823e-06 4.239446e-05 1.379796e-06  
## beta[19] 1.457034e-05 -1.440787e-05 2.279375e-05 2.837002e-06  
## beta[20] -3.780375e-04 -9.021902e-06 1.326944e-04 2.428018e-06  
## beta[21] 9.163907e-05 -2.333126e-06 5.042581e-05 2.072302e-06  
## beta[22] -1.035600e-04 -7.570143e-06 7.310883e-05 2.474918e-06  
## beta[23] -1.782456e-04 -1.211568e-06 7.878820e-05 1.991888e-06  
## beta[24] -2.682620e-04 -3.121634e-06 1.202898e-04 1.610581e-06

# Conclusion

Clearly the Bayesian and Frequentist methods did not yield the same results. This is likely for many reasons. First, we used different sample sizes, and observations, for modelling. The Bayesian would be more similar to the Frequentist model, if we used either 30,000 or 300 for both methods. Again, part of the purpose of this project was to see if a significantly smaller sample size would produce the same results as a Frequentist regression.

Furthermore, the structure of the data likely contributed to the difference. In order to see an actual comparison, we did not classify the categorical variables into ‘factors’. We noticed that running the frequentist regression with the categorical data classified as ‘factors’ lead to different estimates for all the parameters. Additionally, we could have reinterpreted all the time series variables (ex: the number of months in the period an observation was behind in payments rather than how many months an observation is late for each month). This likely would have dealt with the multicollinearity issues.

Finally, the choices we made on the Bayesian model certainly affected the difference between the results. In future studies we could choose more informative priors, run more iterations, and add interaction terms.

# R Markdown Code

---

title: "Project"

author: "Cambrey Sullivan"

date: "11/23/2019"

output:

word\_document: default

pdf\_document: default

html\_document:

df\_print: paged

---

```{r setup, include=FALSE}

knitr::opts\_chunk$set(echo = FALSE)

```

```{r, echo=FALSE}

#install.packages("readxl")

#install.packages("e1071")

#install.packages("class")

#install.packages('glmnet')

#install.packages("ggplot2")

#install.packages("knitr")

library(ggplot2)

library(reshape2)

library(glmnet)

library(caret)

library(ROCR)

```

```{r, echo=FALSE}

library(readxl)

data=read\_xls("CreditDefault.xls")

names(data)[names(data) == "default payment next month"] = "default"

row.names(data) <- data$ID

data[1] <- NULL

d.data=data

d.data$SEX<-as.factor(d.data$SEX)

d.data$EDUCATION<-as.factor(d.data$EDUCATION)

d.data$MARRIAGE<-as.factor(d.data$MARRIAGE)

d.data$default<-as.factor(d.data$default)

```

```{r data\_split}

set.seed(123)

sample.ind = sample(seq(1,dim(data)[1], 1),floor(0.01\*dim(data)[1]), replace = F)

tr.ind = sample(seq(1,dim(data)[1], 1),floor(0.7\*dim(data)[1]), replace = F)

samp.data = data[sample.ind,]

d.train.data=data[tr.ind,]

d.test.data=data[-tr.ind,]

```

# The Data

The data is from the [UCI Machine Learning Repository](https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients). It follows 30,000 lines of credit over a 5 month period and records which default and which don't. There are 23 explanatory variables:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.

X2: Gender (1 = male; 2 = female).

X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).

X4: Marital status (1 = married; 2 = single; 3 = others).

X5: Age (year).

X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows:

X6 = the repayment status in September, 2005;

X7 = the repayment status in August, 2005;

...;

X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.

X12-X17: Amount of bill statement (NT dollar).

X12 = amount of bill statement in September, 2005;

X13 = amount of bill statement in August, 2005;

...;

X17 = amount of bill statement in April, 2005.

X18-X23: Amount of previous payment (NT dollar).

X18 = amount paid in September, 2005;

X19 = amount paid in August, 2005;

...;

X23 = amount paid in April, 2005.

The main objective of this project is to compare frequentest logistic regression results with a a Bayesian Logistic regression using smaller sample size from the data.

```{r}

ggplot(data=d.data, aes(x=default))+

geom\_bar(fill="steelblue")+

labs(title="Response Variable Distribution", x="Loan Status", y=NULL)+

scale\_x\_discrete(breaks=c("0", "1"), labels=c("Loan not in Default", "Loan in Default"))+

theme\_minimal()

```

As we see above, defaulters constitute 6,636 individuals of the total 30,000 individuals included in the data set. This accounts for 22.12% of all customers. The non-proportionality of classes means that assessing the performance of models solely based on test accuracy could be misleading. For example, a model that predicts all test observations as zero has a 78% accuracy rate. However, this does not mean that model is performing well. With this as a backdrop, we tried to assess each individual model based on its performance on a list of statistics besides accuracy.

## Multicollinearity

One of the common problems associated with explanatory variables is a problem of multicollinearity.

```{r}

cormat=cor(data[, c(1,4:23)])

melted\_cormat <- melt(cormat)

ggplot(data = melted\_cormat, aes(x=Var1, y=Var2, fill=value)) +

geom\_tile()+

theme(

axis.title.x = element\_blank(),

axis.title.y = element\_blank(),axis.text.x = element\_text(angle = 45, vjust = 1, hjust = 1))+

ggtitle("Correlation Heat Map")

```

We found that the bill amount variables (X\_12-X\_17) were highly correlated with one another. The following correlation matrix on those variables indicate, their mutual correlation coefficients are greater than 0.9 in absolute value.

```{r}

which((cor(data[, c(1,4:23)])>=0.9 & cor(data[, c(1,4:23)]) < 1), arr.ind=TRUE)

```

# Frequentist Logistic Regression

```{r}

library(glmnet)

library(MASS)

```

Given the explanatory variables, logistic regression assumes that each response variable follows a Bernoulli distribution with probability of $p(y\_i=1|X)$

$y\_i |X \sim Bernoulli(p(y\_i=1|X))$

Since the link function for $p(y\_i=1???X)$ is a logistic distribution, finding the coefficients of the variables involve maximizing the following distribution with respect to the coefficients:

$$

\begin{align}

L(y\_i\mid \mathbf{X}) \sim \prod\_{i=1}^{n} (\frac{1}{1+e^{-\mathbf{XB}}})^ {\sum\_{i=1}^{n}y\_i}(\frac{e^{-\mathbf{XB}}}{1+e^{-\mathbf{XB}}})^ {n-\sum\_{i=1}^{n}y\_i}

\end{align}

$$

```{r}

logit=glm(default ~., data=d.train.data, family = binomial)

summary(logit)

```

We can see that the logistic regression on the training data set has numerous coefficients which are not statistically significant (p-value >0.05) as seen in table 3. These exceptions are likely due to the multicollinearity present in the billing amount variables.

```{r}

p = predict(logit, d.test.data[, 1:24], type = "response")

pred.logistics= ifelse(p > 0.5, 1, 0)

```

```{r}

library(caret)

```

## Confusion Matrix

As seen in the table below, nearly a third of defaulters, 200 of the total 657, are missclassified as non defaulters. As such, this method of classification on the given data set may not be the best method of classification. We are more interested in the misclassifications of defaulters than non-defaulters, since response variable is not proportional.

```{r}

confusionMatrix(as.factor(d.test.data$default), as.factor(pred.logistics))

```

# Bayesian Logistic Regression

For this project, we used JAGS in R to fit the Bayesian logistic model. The model description along with the likelihood and priors can be seen in the code below.

```{r, echo=TRUE}

n <- length(samp.data$default)

bin.col<-c(2:4, 24)

norm.col<-c(1, 5:23)

logistic\_model <- "model{

# Likelihood

for(i in 1:n){

Y[i] ~ dbern(q[i])

logit(q[i]) <- beta[1] + beta[2]\*X[i,1] + beta[3]\*X[i,2] + beta[4]\*X[i,3] + beta[5]\*X[i,4] + beta[6]\*X[i,5] +

beta[7]\*X[i,6] + beta[8]\*X[i,7] + beta[9]\*X[i,8] + beta[10]\*X[i,9] + beta[11]\*X[i,10] +

beta[12]\*X[i,11] + beta[13]\*X[i,12] + beta[14]\*X[i,13] + beta[15]\*X[i,14] + beta[16]\*X[i,15] +

beta[17]\*X[i,16] + beta[18]\*X[i,17] + beta[19]\*X[i,18] + beta[20]\*X[i,19] + beta[21]\*X[i,20] +

beta[22]\*X[i,21] + beta[23]\*X[i,22] + beta[24]\*X[i,23]

}

#Priors

for(j in 1:24){

beta[j] ~ dnorm(0,1000)

}

}"

```

```{r, echo=TRUE}

library(rjags)

dat <- list(Y=samp.data$default,n=n,X=samp.data[ -c(24) ])

model <- jags.model(textConnection(logistic\_model),data = dat,n.chains=3, quiet=TRUE)

update(model, 10000, progress.bar="gui")

samp <- coda.samples(model,

variable.names=c("beta"),

n.iter=20000, progress.bar="gui")

```

## Diagnostics

### Trace Plots

```{r}

traceplot(samp)

```

### Autocorrelation Plots

We see autocorrelation for the variables with high correlation with other variables. This is not concerning for the MCMC. We used a burn in period anyway.

```{r}

# autocorr.plot(samp)[beta[13]]

```

### Gelman Plots

```{r}

# gelman.plot(samp)

```

Indeed, the Gelman plots don't suggest non-convergence.

### Effective Sample Size

```{r}

effectiveSize(samp)

```

### Geweke Diagnostic

```{r}

geweke.diag(samp)

```

## Bayesian Output

The following table is the output for the Bayesian model.

```{r}

summary(samp)

```

```{r}

beta1 <- samp[[1]] # samples from chain 1

beta2 <- samp[[2]]

beta3 <- samp[[3]]

beta <- rbind(beta1,beta2,beta3)

dim(beta1)

```

```{r}

odds <- exp(beta)

colnames(odds)<-c("Int",colnames(samp.data[ -c(24) ]))

odds <- odds[,-1]

apply(odds,2,mean)

```

```{r}

apply(odds,2,sd)

```

```{r}

for(j in 1:23){

hist(odds[,j],breaks=50,main=colnames(samp.data[ -c(24) ])[j],xlab="Odds")

}

```

# Results

```{r}

coefb = summary(samp)$statistics[,1]

sdb = summary(samp)$statistics[,2]

coef2 = summary(logit)$coefficients[,1]

sd2 = summary(logit)$coefficients[,2]

compare = as.data.frame(cbind(coefb, coef2, sdb, sd2))

names(compare) = c("Bayesian Means", "Frequntist Estimates", "Bayesian SD", "Frequntist SE")

print(compare)

```