Stat 547C Project: Bayesian Variable Selection using Gibbs with Case Studies

Ken Lau

December 15, 2013

1 Summary

We applied stepwise, Lasso, and Gibbs variable selection methods on a simulated data set and a data set based on a real-time strategy game called Starcraft 2. In our two case studies, we find that stepwise can easily overfit the model even with a BIC stopping criterion. The Lasso method can also underfit models or not detect some useful predictors. However, applying the Gibbs variable selection method, we are able to detect the true predictors explaining the response variable in case study 1. In case study 2, the globally optimized method of Gibbs allows us to pick up useful predictors that are not present in stepwise and Lasso.

2 Introduction

Stepwise regression methods have been a popular choice for many years for variable selection. However, we encounter problems such as adding variables in a greedy fashion, as well as, adding random noise due to multiple tests. In the recent years, Hastie and Tibshirani have developed shrinkage methods including lasso and ridge to allow more flexibility in the addition of variables in models. However, if we do not test each combination of variables included in the model, we are unable to find the global optimum. We can certainly try running a stepwise exhaustive search, but that would take a long time to run. Moreover, it will not give us all combinations. In order to test all 2^p combinations of predictors, we must convert the problem into Bayes. MCMC methods are usually quite fast, and with the final drawn distribution, we can interpret the predictors in our model.

3 Methods

We focus on the normal linear regression setting. So, we fit a model with the assumptions that $Y|\beta, \sigma^2 \sim N(X\beta, \sigma^2)$. For stepwise, we assume that we start out with just the intercept in our model. For each iteration, we could compute the AIC for each variable separately added to the model. We add the variable with the greatest AIC, until we reach a model with a locally optimum AIC. Shrinkage type methods include the Lasso. So, for the least squares solution of $\hat{\beta}_{LS}$, we include a regularization term on all the coefficients. In other words, the solution of the form,

$$min\{\frac{1}{n}\sum(y - \alpha - \beta^T x)^2 + \lambda \sum |\beta_j\}$$

Cross validation is used to determine λ. We are balancing the fit of a linear model and the shrinkage of the coefficients. The L_1 penalty causes some of the coefficients to shrink to 0.

For Bayesian variable selection methods, we are interested in solving the likelihood $L(\beta, \sigma) = f(Y|\beta, \sigma, X)$ [1]. We introduce an additional parameter γ, where $\gamma \in \{0, 1\}$ for whether the variable is in the model or not. This is a problem of solving $L(\beta, \sigma, \gamma|Y)$. We can revert to Gibbs sampling. If we follow the steps in George and McCulloch (1997) [3], our target of full conditionals are,

$$\pi(\beta|\sigma, \gamma, Y)$$

$$\pi(\sigma|\beta, Y)$$

$$\pi(\gamma_i|\beta, \gamma_{-i})$$

We can denote $\gamma \sim Bern(p)$. It is possible to represent a normal mixture for

$$\beta_i|\gamma_i \sim (1 - \gamma_i)N(0, \tau_i^2) + \gamma_i N(0, c_i^2\tau_i^2)$$

We have $\beta_i \sim N(0, \tau_i^2)$ when $\gamma_i = 0$, and $\beta_i \sim N(0, c_i^2\tau_i^2)$ when $\gamma_i = 1$ [2]. We would then obtain,

$$\beta|\gamma \sim N(0, D_\gamma R D_\gamma)$$

where R is a prior correlation matrix and $D_\gamma = diag(a_1\tau_1 \dots a_p\tau_p)$ [2]. Moreover, it is found that,

$$\sigma^2|\gamma \sim IG(\frac{\nu_\gamma}{2}, \frac{\nu_\gamma\lambda_\gamma}{2})$$

where IG stands for the inverse gamma. [2]. The paper develops more into picking $c_i, \tau_i, D_\gamma, R, \nu_\gamma$, and λ_γ [2]. By invoking SVSS and gibbs sampling we

want to obtain a sequence, $\beta^{(0)}, \sigma^{(0)}, \gamma^{(0)} \ldots \beta^{(i)}, \sigma^{(i)}, \gamma^{(i)} \ldots$ [2]. From [2], we would sample β from the following,

$$\beta^{(j)} \sim N(A_{\gamma^{(j-1)}}(\sigma^{(j-1)})^{-2} X^T X \hat{\beta}_{LS}, A_{\gamma^{(j-1)}})$$

where $A_{\gamma^{(j-1)}} = ((\sigma^{(j-1)})^{-2} X^T X + D^{-1}_{\gamma^{(j-1)}} R^{-1} D^{-1}_{\gamma^{(j-1)}})^{-1}$ Next, we sample $\sigma^{(j)}$ from the following,

$$\sigma^{(j)} \sim IG\left(\frac{n + \nu_{\gamma^{(j-1)}}}{2}, \frac{|Y - X\beta^{(j)}|^2 + \nu_{\gamma^{(j-1)}} \lambda_{\gamma^{(j-1)}}}{2}\right)$$

Finally, we sample γ_i for each i from the following,

$$P(\gamma_i^{(j)} = 1 | \beta^{(j)}, \sigma^{(j)}, \gamma_{-i}^{(j)}) = \frac{a}{a+b}$$

where $a = f(\beta^{(j)} | \gamma_{-i}^{(j)}, \gamma_i^{(j)} = 1) p_i$ and $b = f(\beta^{(j)} | \gamma_{-i}^{(j)}, \gamma_i^{(j)} = 0)(1 - p_i)$ [2]. We are also required to supply a prior on the coefficients. A robust prior has been suggested in Bayarri 2008 [6]. The article has more information on the formulation of priors. An implementation of Gibbs variable selection (as described above) is in the R package BayesVarSel [10]. We will make use of the function GibbsBvs in our case studies.

4 Case Study 1

The goal of this case study is to demonstrate the usefulness of Bayesian selection methods as opposed to stepwise and shrinkage methods. A simulated data set is generated (see Appendix A). We have a response variable y which depends on x_1, x_2, x_3, x_4 predictors. We add in ten other variables with a strong relationship to y, but are not optimal because each represent only a proper subset combination of x_1, x_2, x_3, x_4. For forward stepwise with AIC stopping criterion we get x_{12}, x_{13}, x_5, x_7, x_{11}, x_4, x_1, x_2, x_3 selected variables in the respective order. We can see that variables $x_{12}, x_{13} \ldots, x_{11}$ are selected before selecting x_1, x_2, x_3, x_4 which should explain y the best. This shows how using forward stepwise could lead to an overfitted model. Lasso does a better job. However, we can see that it did not converge optimally. We used cross-validation to obtain an optimal shrinkage parameter λ. We found that all variables are included in the model. However, variables $x_5, \ldots, x_{14}$ are shrunk nearly to 0, which is a better representation of the additional variables (see plot in Appendix B). Finally, we turn to Gibbs variable selection with a robust prior. Our final model included just x_1, x_2, x_4,and x_{14}. The variable x_{14} is selected instead of x_3, because x_{14} is correlated with x_3. Therefore, we find that the Bayesian selection method globally optimizes the model. (See Appendix B).

5 Case Study 2

Starcraft 2 is a 1 vs 1 real-time strategy game. Each player builds up an economy by collecting minerals and gas with workers, and use the resources to train fighting units. The objective is to destroy the enemies' headquarters [11]. Each player is ranked in one of eight leagues (bronze, silver, gold, platinum, diamond, master, grandmaster, and professional). We are interested in features which best explains which league a player is in. Thompson has developed a list of useful features that are representative of the skill involved in a particular game. A few of the features include, hours played per week, action per minute, number of hotkeys used per time-stamp, etc. For more information on the full data set see Thompson, 2013 [11]. In his paper, he worked on random forests and variable importance to rank the usefulness of each feature. We will work on variable selection, and apply stepwise, lasso, and Gibbs variable selection on this data set. Since the response variable is ordinal, we will make an assumption that each value is equidistant. So, we could treat the response as numerical. The results from applying variable selection on the data set is in Table 1. All the code and results can be found in Appendix C. Note that Thompson implemented a variable importance method, and we considered the top seven variables. We could see that stepwise over fitted the model. For example, it included complex units and unique units made. From experience, the amount of complex and unique units could vary between all skill levels. Many players in silver and gold leagues could equally build as many of the complex units in a game as a master league player. The Lasso method appeared to have thrown away some useful predictors such as, workers made. Workers made is the number of workers trained per timestamp. In Starcraft 2 a lot of success is in building up a strong economy, and most star players achieve that by the ability to continuously train workers while attacking their opponents. For the Gibbs variable selection method, it included a few useful predictors such as minimap attacks and workers made. It also included predictors such as unique hotkeys and actions in PAC, which are useful predictors.

variable	stepwise	lasso	Gibbs	Thompson
ActionLatency	1	1	0	1
APM	1	0	0	1
AssignToHotkeys	1	1	1	1
MinimapAttacks	1	1	1	0
GapBetweenPACs	1	0	0	1
WorkersMade	1	0	1	1
NumberOfPACs	1	1	0	1
HoursPerWeek	1	0	0	0
SelectByHotkeys	1	0	0	1
UniqueHotkeys	1	0	1	0
TotalMapExplored	1	0	1	0
Age	1	0	1	0
ActionsInPAC	1	0	1	0
ComplexUnitsMade	1	0	0	0
UniqueUnitsMade	1	0	0	0
MinimapRightClicks	0	0	1	0
HoursPerWeek	0	0	0	0
TotalHours	0	0	0	0
ComplexAbilitiesUsed	0	0	0	0

Table 1: Variables Selected for each Method. A value of 1 means the variable is included, and 0 means the variable is not included

6 Conclusion

We have demonstrated a Bayesian variable selection method, which allows us to search all combinations of predictors to find a globally optimal model. In the first case study, we showed how the forward stepwise method can overfit models by adding other predictors early in the process. Whereas Gibbs variable selection checks all combinations of predictors in the model. In the second study, the stepwise selection method overfitted the model and Lasso was unable to detect many useful features. Gibbs variable selection is able to find a useful set of predictors which are not detected by stepwise and Lasso.

References

[1] Atilla Yardimci, Aydin Erar, *Bayesian Variable Selection in Linear Regression and a Comparison*, 2002.

[2] Edward I. George, Robert E. McCulloch, *Variable Selection Via Gibbs Sampling*, 1993.

[3] Edward I. George, Robert E. McCulloch, *Approaches for Bayesian Variable Selection*, 1997.

[4] Sanford Weisburg, *Variable Selection and Regularization*, 2012.

[5] Trevor Hastie, Robert Tibshirani, Jerome Friedman, *Elements of Statistical Learning*, 2008

[6] M.J. Bayarri, J.O. Berger, A. Forte, G. Garcia-Donato, *Criteria for Bayesian Model Choice with Application to Variable Selection*, 2008.

[7] William Welch, *STAT 306 Finiding Relationships in Data: An Introduction to Building Statistical Models*, 2010

[8] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, url=http://www.R-project.org/, 2013.

[9] Jerome Friedman, Trevor Hastie, Robert Tibshirani, *Regularization Paths for Generalized Linear Models via Coordinate Descent*, Journal of Statistical Software, url=http://www.jstatsoft.org/v33/i01/, 2010.

[10] Gonzalo Garcia-Donato and Anabel Forte, *BayesVarSel: Bayesian Variable Selection in Linear Models*, R package version 1.3, url=http://CRAN.R-project.org/package=BayesVarSel, 2013.

[11] Thompson JJ, Blair MR, Chen L, Henry AJ, *Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning*, PLoS ONE 8(9): e75129, doi:10.1371/journal.pone.0075129, 2013.

Appendix A

Here is the R code for simulating the data for case study 1.

```
#number of observations
n=350
#generate important predictors
x1 = rnorm(n=n, mean=2, sd=1)
x2 = rnorm(n=n, mean=3, sd=1)
x3 = rnorm(n=n, mean=1, sd=1)
x4 = rnorm(n=n, mean=.5, sd=1)
```

```
#response variable
y = x1*3.2 + x2*1.8 + x3 + x4*2.5 + rnorm(n=n, sd=.5)

##add good predictors but not optimal#
x5 = x1 + x2 + rnorm(n=n, sd=.5)
x6 = x1 + x3 + rnorm(n=n, sd=.5)
x7 = x1 + x4 + rnorm(n=n, sd=.5)
x8 = x2 + x3 + rnorm(n=n, sd=.5)
x9 = x2 + x4 + rnorm(n=n, sd=.5)
x10 = x3 + x4 + rnorm(n=n, sd=.5)
x11 = x1 + x2 + x3 + rnorm(n=n, sd=.5)
x12 = x1 + x2 + x4 + rnorm(n=n, sd=.5)
x13 = x1 + x3 + x4 + rnorm(n=n, sd=.5)
x14 = x2 + x3 + x4 + rnorm(n=n, sd=.5)
##construct covariate matrix
x = data.frame(y=y, x1=x1, x2=x2, x3=x3, x4=x4, x5=x5, x6=x6, x7=x7,
      x8=x8, x9=x9, x10=x10, x11=x11, x12=x12, x13=x13, x14=x14)

#save simulated data
#write.table(x, "x.txt", row.names=F, col.names=T, sep=",")
#read in simulated data
x = read.table("x.txt", header=T, sep=",")
#test = read.table("test.txt", header=T, sep=",")
```

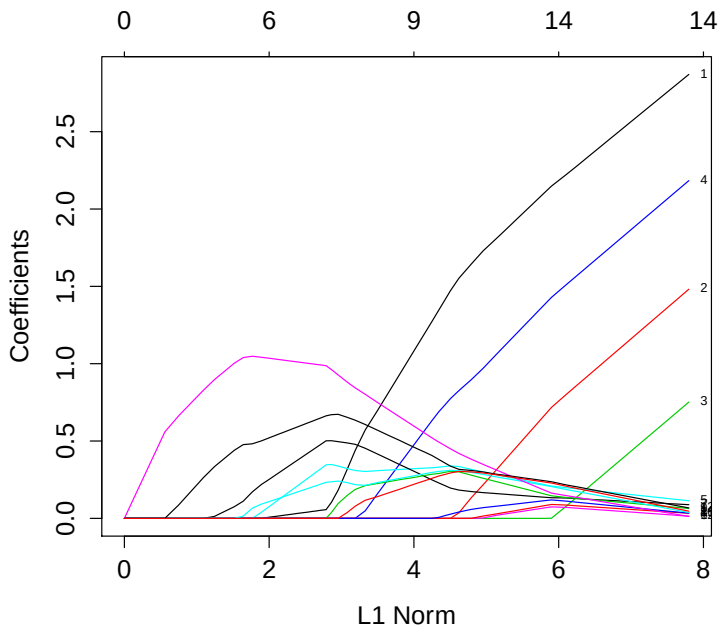
Appendix B

Here is the code and results for the analysis of Case Study 1.

```
#forward stepwise
#call forward stepwise
out_step = step(lm(y~1, data=x), list(upper=lm(y~., data=x)),
            direction="forward")
#extract out selected variables
coef_step = names(out_step$coefficients)[-1]
coef_step
#[1] "x12" "x13" "x5"  "x7"  "x11" "x4"  "x1"  "x2"  "x3"

#lasso
#extract predictor X matrix
X = data.matrix(x[,-match("y",names(x))])
#extract response variable
```

```
Y = data.matrix(x[,match("y",names(x)),drop=F])
#call cross validation in lasso to find optimal lambda
cv_lasso = cv.glmnet(x=X, y=Y, alpha=1)
out_lasso = cv_lasso$glmnet.fit
#extract selected variables and respective coefficients
coef_lasso = out_lasso$beta[,which.min(cv_lasso$cvm)]
coef_lasso
#         x1          x2          x3          x4          x5          x6
#2.87020970 1.48168827 0.75196938 2.18357433 0.11383759 0.01381382
#x7          x8          x9         x10         x11         x12
#0.08840575 0.03285751 0.06363527 0.03374477 0.03813501 0.01257526
#x13         x14
#0.06746695 0.04644490
```




```
#gibbs
#call gibbs variable selection
out_gibbs = GibbsBvs(y~., data=x, prior.betas="Robust",
              prior.models="Constant", n.burnin=300, n.iter=20000)
#outputs a table with information on selected variables
coef_gibbs = out_gibbs$HPMbin
coef_gibbs = cbind(coef_gibbs, names(x)[-1])
names(coef_gibbs) = c("bin.mod", "coef")
```

```
#extract selected variables
coef_gibbs = as.character(coef_gibbs$coef[coef_gibbs$bin.mod==1])
coef_gibbs
#[1] "x1"  "x2"  "x4"  "x14"
```

Appendix C

Here is the code and results for the analysis of Case Study 2.

```
starcraft = read.table("SkillCraft1_Dataset.csv", sep=",", header=T,
              stringsAsFactors=T, na.strings="?")
starcraft = starcraft[,grep("^((?!gameid).)",names(starcraft),
              ignore.case=T,perl=T)]
starcraft = starcraft[complete.cases(starcraft),]

##forward stepwise
out_step = step(lm(LeagueIndex~1, data=starcraft),
            list(upper=lm(LeagueIndex~., data=starcraft)),
            direction="forward", trace=0)
coef_step = names(out_step$coefficients)[-1]
coef_step
#[1] "ActionLatency"    "APM"              "AssignToHotkeys"
#[4] "MinimapAttacks"   "GapBetweenPACs"   "WorkersMade"
#[7] "NumberOfPACs"     "HoursPerWeek"     "SelectByHotkeys"
#[10] "UniqueHotkeys"    "TotalMapExplored" "Age"
#[13] "ActionsInPAC"     "ComplexUnitsMade" "UniqueUnitsMade"

#lasso
X = data.matrix(starcraft[,-match("LeagueIndex",names(starcraft))])
Y = data.matrix(starcraft[,match("LeagueIndex",names(starcraft))])
cv_lasso = cv.glmnet(x=X, y=Y, alpha=1)
out_lasso = cv_lasso$glmnet.fit
coef_lasso = out_lasso$beta[,which.min(cv_lasso$cvm)]
coef_lasso
#     Age         HoursPerWeek           TotalHours
#0.000000000          0.000000000          0.000000000
#APM      SelectByHotkeys      AssignToHotkeys
#0.005411143          0.000000000        572.435551051
#UniqueHotkeys       MinimapAttacks   MinimapRightClicks
#0.000000000        117.481302610          0.000000000
```

```
#NumberOfPACs       GapBetweenPACs        ActionLatency
#62.543716813         -0.004358252         -0.021668480
#ActionsInPAC     TotalMapExplored          WorkersMade
#0.000000000          0.000000000          0.000000000
#UniqueUnitsMade     ComplexUnitsMade ComplexAbilitiesUsed
#0.000000000          0.000000000          0.000000000

#gibbs
out_gibbs = GibbsBvs(LeagueIndex~., data=starcraft,
              prior.betas="Robust", prior.models="Constant",
              n.burnin=300, n.iter=20000)
coef_gibbs = out_gibbs$HPMbin
coef_gibbs = cbind(coef_gibbs, names(starcraft)[-1])
names(coef_gibbs) = c("bin.mod", "coef")
coef_gibbs = as.character(coef_gibbs$coef[coef_gibbs$bin.mod==1])
coef_gibbs
#[1] "Age"                "AssignToHotkeys"    "UniqueHotkeys"
#[4] "MinimapAttacks"     "MinimapRightClicks" "ActionsInPAC"
#[7] "TotalMapExplored"   "WorkersMade"
```