Homework 5

Hong Xu, Individual

October 24, 2015

All code can be found in bitbucket.

0. Prepare the model

```
require(xtable)
```

```
## Loading required package: xtable
```

```
require(gamlr) # loads Matrix as well
```

```
## Loading required package: gamlr
## Loading required package: Matrix
```

```
data(hockey) # load the data
# Combine the covariates all together
x <- cBind(config,team,player) # cBind binds together two sparse matrices
# build 'y': home vs away, binary response
y <- goal$homegoal

nhlreg <- gamlr(x, y,
    free = 1:(ncol(config)+ncol(team)), ## free denotes unpenalized columns
    family = "binomial", standardize = FALSE,
    verb = FALSE)
```

1. Interpret AICc selected model from nhlreg lasso

```
## coefficients (grab only the players)
B <- coef(nhlreg)[colnames(player),]
B[order(-B)[1:10]] # 10 biggest
```

```
## PETER_FORSBERG  TYLER_TOFFOLI   ONDREJ_PALAT ZIGMUND_PALFFY  SIDNEY_CROSBY
##      0.7548254      0.6292577      0.6284040      0.4426997      0.4131174
##   JOE_THORNTON  PAVEL_DATSYUK  LOGAN_COUTURE      ERIC_FEHR MARTIN_GELINAS
##      0.3837632      0.3761981      0.3682103      0.3677283      0.3577613
```

The top 10 partial effects of the players: which shows how "strong" a player's presence on the field can affect the score of the homegoal. For example, if player i's coefficient is β_i, then it means: when a goal is scored and player i is on the field, odds are multiplied by e^{β_i} that his team scored.

Specifically, to translate the coefficient to Plus-Minus:

```r
## convert to expected partial plus-minus
## reference: the help document of gamlr::hockey
ng <- colSums(abs(player[,names(B)])) # total number of goals
# The individual effect on probability that a
# given goal is for vs against that player's team
p <- 1/(1+exp(-B))
# multiply ng*p - ng*(1-p) to get expected plus-minus
ppm <- ng*(2*p-1)
# organize the data together and print top 20
effect <- data.frame(b = round(B,3),
                     n_games = as.integer(ng),
                     ppm = round(ppm,3))
effect <- effect[order(-effect$ppm),]
effect_table <- xtable(effect[1:20,])
```

	b	n_games	ppm
JOE_THORNTON	0.38	1740	329.84
PAVEL_DATSYUK	0.38	1725	320.70
SIDNEY_CROSBY	0.41	1568	319.36
ALEX_OVECHKIN	0.30	1705	254.52
HENRIK_LUNDQVIST	0.17	3040	251.76
HENRIK_SEDIN	0.29	1634	237.18
MARIAN_HOSSA	0.26	1752	230.24
NICKLAS_LIDSTROM	0.21	2128	223.94
DANIEL_ALFREDSSON	0.25	1732	216.01
ANDREI_MARKOV	0.28	1549	213.38
MIIKKA_KIPRUSOFF	0.14	3076	208.80
MARIAN_GABORIK	0.32	1287	203.00
ALEXANDER_SEMIN	0.35	1148	199.86
CHRIS_PRONGER	0.26	1557	197.41
HENRIK_ZETTERBERG	0.21	1803	192.92
PETER_FORSBERG	0.76	532	191.76
JONATHAN_TOEWS	0.31	1200	182.30
TEEMU_SELANNE	0.31	1197	182.17
LUBOMIR_VISNOVSKY	0.31	1179	179.72
RYAN_GETZLAF	0.27	1341	179.29

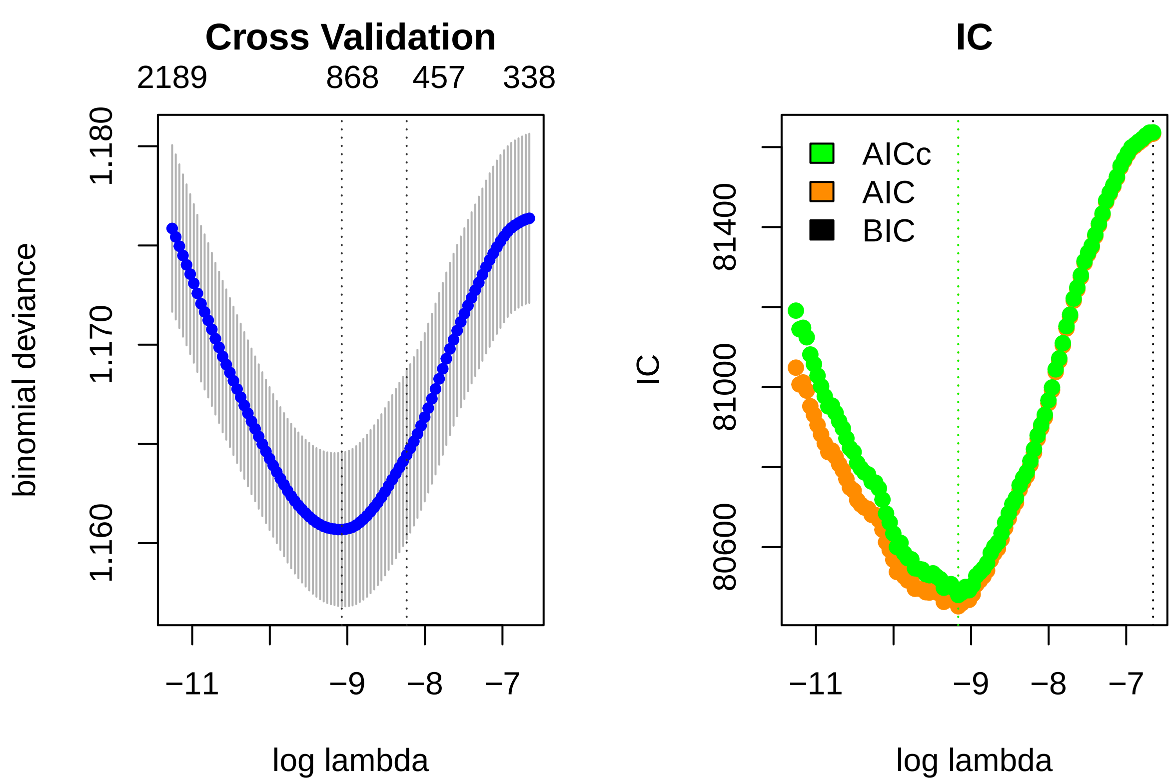
This gives an overall rank of player by combining the "personal effect"" and the number of goals they "witness" into traditional Plus-Minus. Interestingly, for some players, such as HENRIK LUNDQVIST, who is not very high in "personal effect" (only 0.17), but since he has been on the ice for numerous goals (3040), he still ranks very high.

2. Standardization

All the predictors in this example use the the same scale of $-1, 0, 1$, and it nicely represents the concept of home/away (i.e. 1 for home, -1 for away). If we standardize, we will lose the interpretability of the resulting coefficients, and will result in a different, unreasonable model as well. For example, if the number of games Player A plays is different from Player B's, then the value that stands for "home team player on the ice" will be different among them, which does not make sense at all (they should always be equal).

3. Compare model seletion methods

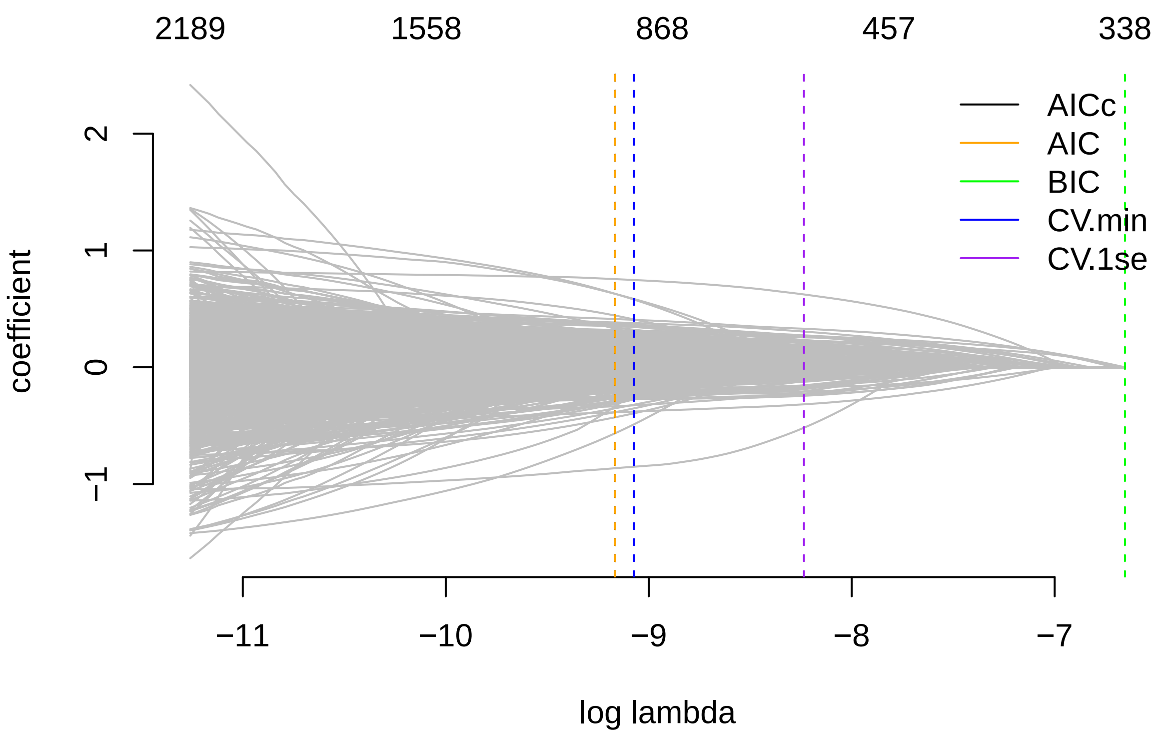
```r
## Cross Validation
nhlreg.cv <- cv.gamlr(x, y,
    free = 1:(ncol(config)+ncol(team)), ## free denotes unpenalized columns
    family = "binomial", standardize = FALSE,
    verb = FALSE)

## plot CV results and the various IC
ll <- log(nhlreg$lambda) ## the sequence of lambdas
old_par <- par(mfrow=c(1,2))
plot(nhlreg.cv, main = "Cross Validation")
plot(ll, AIC(nhlreg),
     xlab="log lambda",
     ylab="IC",
     main = "IC",
     pch=19, col="darkorange")
abline(v=ll[which.min(AIC(nhlreg))], col="darkorange", lty = 3)
abline(v=ll[which.min(BIC(nhlreg))], col="black", lty = 3)
abline(v=ll[which.min(AICc(nhlreg))], col="green", lty = 3)
points(ll, BIC(nhlreg), pch=19, col="black")
points(ll, AICc(nhlreg), pch=19, col="green")
legend("topleft", bty="n",
    fill=c("green","darkorange","black"),legend=c("AICc","AIC","BIC"))
```



```r
par(old_par)
```

AIC curve and AICc curve look very similar. AICc curve looks like CV curve.
In practice, BIC works more like the 1se CV rule (It is out of range on y axis in this plot).
But with big n, it chooses too simple models (it underfits).

```r
## all metrics, together in a path plot.
plot(nhlreg, col="grey")
abline(v=ll[which.min(AICc(nhlreg))], col="black", lty=2)
abline(v=ll[which.min(AIC(nhlreg))], col="orange", lty=2)
abline(v=ll[which.min(BIC(nhlreg))], col="green", lty=2)
abline(v=log(nhlreg.cv$lambda.min), col="blue", lty=2)
abline(v=log(nhlreg.cv$lambda.1se), col="purple", lty=2)
legend("topright", bty="n", lwd=1,
    col=c("black","orange","green","blue","purple"),
    legend=c("AICc","AIC","BIC","CV.min","CV.1se"))
```

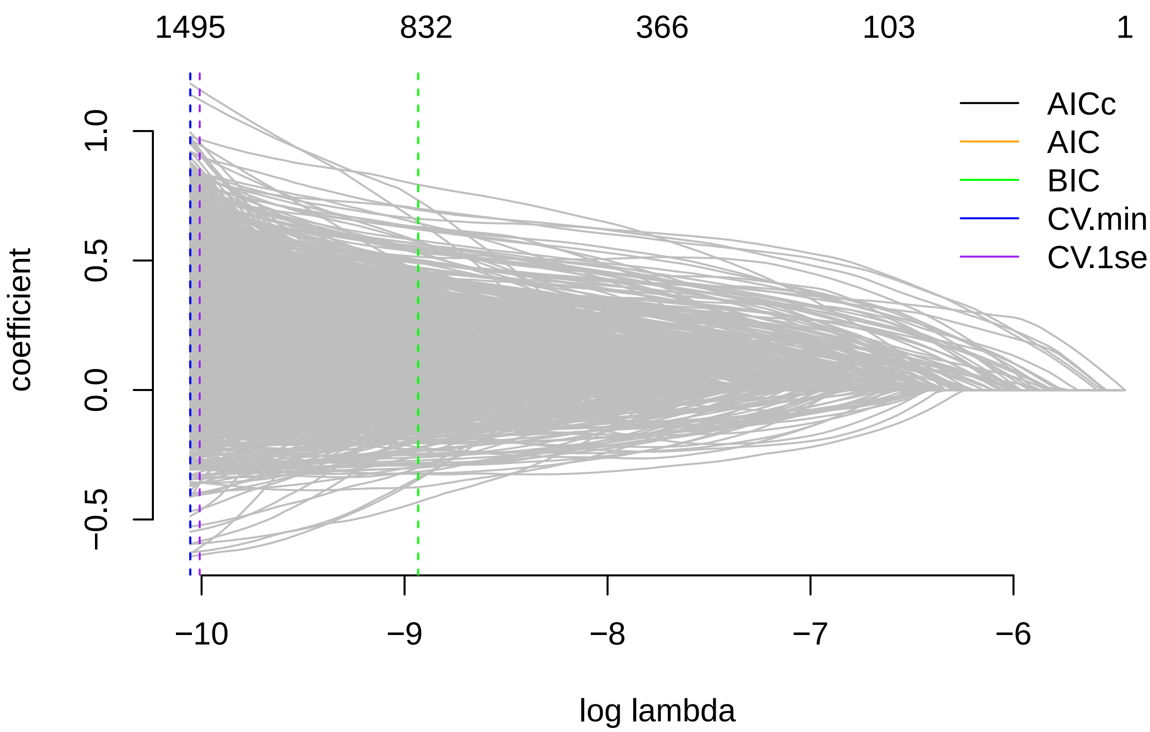


Again, AICc and CV min choose similar complexity for the model ($log(\lambda)$ around -9). BIC favors larger lambda, thus result in a too simple model in this case.

4. Confounding information

```r
# If we ingore the info and fit a player only model
x <- player
```

```r
# build 'y': home vs away, binary response
y <- goal$homegoal
nhlreg_player <- gamlr(x, y,
                        family = "binomial",
                      standardize = FALSE)
nhlreg_player_cv <- cv.gamlr(x, y,
                            family = "binomial",
                            standardize = FALSE)

plot(nhlreg_player, col="grey")
abline(v=ll[which.min(AICc(nhlreg_player))], col="black", lty=2)
abline(v=ll[which.min(AIC(nhlreg_player))], col="orange", lty=2)
abline(v=ll[which.min(BIC(nhlreg_player))], col="green", lty=2)
abline(v=log(nhlreg_player_cv$lambda.min), col="blue", lty=2)
abline(v=log(nhlreg_player_cv$lambda.1se), col="purple", lty=2)
legend("topright", bty="n", lwd=1,
    col=c("black","orange","green","blue","purple"),
    legend=c("AICc","AIC","BIC","CV.min","CV.1se"))
```



If we fit a player-only model, LASSO fails in variable selection in both IC and CV (they just choose the full model!), except for BIC, where it favors a simpler model.
However we might gain some interpretability in this case since every player's effect can be directly mapped to "home game goal" probability regardless of the levels of "confounding variables".