MIS 381N Stochastic Control and Optimization: Project 3

Anurag Agarwal, Siqi Chen, Nicholas Scheifler, Aishwarya Srikanth

March 21, 2018

Direct Selection - MIQP Problem

To use MIQP for Lasso regression, where we want to choose at most k variables out of p variables, we need to choose $\beta_i, z_i, i = 1, 2, ..., 64$, where z_i is a binary indicator variable to indicate whether β_i is zero. We want to minimize

$$\frac{1}{2}\|y - X\beta\|_2^2$$

subject to the constraints:

$$\begin{cases} \sum_{i=1}^{64} z_i \leq k \\ -Mz_i \leq \beta_i \leq Mz_i, \quad i = 1, ..., p \\ z_i \text{ is binary}, \quad i = 1, ..., p \end{cases}$$

Hence, we first need to represent the objective function in the form (given the format of 'Gurobi' API)

$$x^T Q x + c^T x + alpha$$

and the constraints in the form

$$Ax = b$$

x is the vector of variables we're solving for, then

$$x = \begin{bmatrix} \beta_1 \\ ... \\ \beta_{64} \\ z_1 \\ ... \\ z_{64} \end{bmatrix}$$

Now, expanding the objective function and contraints into matrix form, we can see that

$$Q = \frac{1}{2}X^T X, \quad c = X^T y, \quad A = \begin{bmatrix} 1 & ... & 1 & 0 & ... & 0 \\ 1 & ... & 0 & M & ... & 0 \\ & & & ... & & \\ 0 & ... & 1 & 0 & ... & M \\ 1 & ... & 0 & -M & ... & 0 \\ & & & ... & & \\ 0 & ... & 1 & 0 & ... & -M \end{bmatrix}, \quad b = \begin{bmatrix} k \\ 0 \\ ... \\ 0 \\ 0 \\ ... \\ 0 \end{bmatrix}$$

Then, using the package Gurobi, we could sovle for $\beta_i, \quad i = 1, ..., 64$ in R (see complete code in attached R file)

```
> library(gurobi)
> k = 8
> M = 0.5
> success = FALSE
> while (success == FALSE){
+   X.bin = matrix(0,nrow = n, ncol = p)
+   X.adj = cbind(X, X.bin)
+   model = list()
+   model$obj = -t(X.adj) %*% y
+   model$Q = (t(X.adj) %*% X.adj)*0.5
+   model$A = rbind(c(rep(0,p),rep(1,p)),
+                   cbind(diag(p),diag(p)*M),
+                   cbind(diag(p),diag(p)*(-M)))
+   model$rhs = c(k,rep(0,2*p))
+   model$modelsense = "min"
+   model$sense = c('<=', rep('>=',p), rep('<=',p))
+   model$vtype = c(rep('C',p), rep('B',p))
+   result = gurobi(model)
+   miqp.beta = result$x[1:64]
+   if (max(miqp.beta) == M){
+     M = M * 2
+   }
+   else{
+     success = TRUE
+   }
+ }
```

Lasso with glmnet

Then we fit Lasso regression directly with the package glmnet in R, with λ ranging from 0.01 to 100

```
> library(glmnet)
> grid = 10^seq(-2,2,length = 100)
> lasso.mod = glmnet(X, y, alpha = 1, lambda = grid, intercept = FALSE)
> cv.lasso = cv.glmnet(X, y, alpha = 1, lambda = grid, intercept = FALSE)
> bestlam.lasso = cv.lasso$lambda.1se
> lasso.beta = predict(lasso.mod, type = 'coefficients', s = bestlam.lasso)
```

Comparing Results: MIQP vs Lasso

The last step is to compare the prediction error between β_i we obtained from MIQP vs Lasso. The prediction error is defined as

$$\frac{\left\|X\widetilde{\beta} - X\beta^0\right\|_2^2}{\left\|X\beta^0\right\|_2^2}$$

Using the estimate for β_i from previous parts we can calculate that the prediction errors are

Method	Prediction Error
MIQP	0.004456
Lasso	0.018433

Note: we used 1se as the critiria for best lambda in Lasso (glmnet)