STA242 Final Project Proposal

Chi Po Choi (UCD ID:912494157), Amy Kim (UCD ID:912492829)

April 25, 2015

We would like to write a R package for the *Hypercube estimators* [**?**] introduced by Rudolf Beran. We have noticed there is no package available to utilize the hypercube esimating method, so it would be nicer if there is such a package, then people can get estimators by more convenient way.

1 Description of Hypercube estimators

Given a data set (y, X), we fit the model:

$$y = X\beta + \epsilon.$$

Let $\eta = \mathrm{E}(y) = X\beta$. We would like to find an hypercube estimator $\hat{\eta}$,

$$\hat{\eta}_{\mathrm{H}}(V) = XV(VX'XV + I_p - V^2)^{-1}VX'y$$

where V is a symmetric matrix with all eigenvalues $\in [0, 1]$

We can compute the V that the risk $\mathrm{E}|\hat{\eta}_{\mathrm{H}}(V) - \eta|^2$ is minimized by restricting V into some suitable subclasses. Hypercube estimator are better than the least square estimator since the least square estimator does not minimize the risk.

Beside, the hypercube estimator $\hat{\eta}_{\mathrm{H}}(V)$ is a generalization of penalized least squares estimator and submodel least square estimator. Also, the computation of $(\hat{\eta}_{\mathrm{H}}(V))^{-1}$ is numerically stable. It is proved in Beran's article[**?**].

1. Penalized least squares estimator

   $$\hat{\eta}_{\mathrm{pls}} = X\hat{\beta}_{\mathrm{pls}}$$

   where

   $$\hat{\beta}_{\mathrm{pls}} = \arg\min_{\beta}[|y - X\beta|^2 + \beta' W\beta] = (X'X + W)^{-1}X'y$$

2. Submodel least squares estimators:

   $$\hat{\eta}_{\mathrm{sub}} = X_0 X_0^{+} y \qquad \text{where } R(X_0) \subset R(X) \text{ and } {}^{+} \text{ denotes MoorePenrose pseudoinverse}$$

2 Description of the R package

We would like to write a `R` package[**?**] for the hypercube estimators. The package will contain functions which facilitate data analysis using hypercube estimators. We will also define corresponding classes and methods. We want classes and methods to be compatible with the class `lm` in `R`.

For example, we will implement the function `helm()` (stands for HyperCube Estimator Linear Model), `sublm()` (Submodel least square estimation) and `plm()` (Penalized least square estimation). Those functions return variables of the class `helm` with methods `summary`, `predict`, `coef`, `residuals`, etc.

3 Our plan on the R package

- We implement functions for hypercube estimators.
- We define classes and methods for hypercube estimators.
- We include example data sets.
- We write full documentation on the classes and functions in the package.
- We maintain a git repository for the package.
- The package satisfies the CRAN repository policy and will be eventually available in the CRAN repository .

4 Expected difficulties

- We want our package to be accessible to those people are not familiar with the details of the hypercube estimators. We need to design an "easy interface" which make the functions `sublm` and `plm` just work. It may be difficult to design a robust way to provide suitable V for different user-input data.
- Besides the "easy interface", we want more feasible interface which allows advanced users do some small tweaks. We want to make those advanced users feel that using our package is easier than writing their own codes. To find the balance between "easy" and "hackable" may be difficult.
- As mentioned in the description of Hypercube estimators, hypercube estimators are numerical stable [**?**]. We want our package do have this advantage. It may be difficult, because it may require some advanced knowledges in numerical computations.
- Our team does not have previous experience in building softwares. We will need some time to learn.

5 Software

- We use `RStudio` because it has convenient features for building package.

6 Appendix: Some example of code

```
#PLS linear operator
####################
Dif <- D2
XX <- t(X) %*% X
A.pls <- function(lam){
X %*% ginv(XX + lam*t(Dif)%*%Dif) %*% t(X)
}

#Minimum relevant eigenvalue
############################
elmin <- function(lam){
B <- XX + lam*t(Dif)%*%Dif
el <- eigen(B,symmetric=T,only.values=T)$values
min(el)
}

#PLS estimator function (Dmhat = D%*%mhat)
##########################################
```

```
fit.pls <- function(lam){
etahat <- A.pls(lam) %*% y
mhat <- ginv(XX + lam*t(Dif)%*%Dif) %*% t(X) %*% y
list(mhat=mhat,etahat=etahat)
}

#Estimated risk function
########################
rhat <- function(A){
rh <- normsq(y - A%*%y) + (2*sum(diag(A)) - n)*ss
rh/q
}

risk.pls <- function(lam){
rhat(A.pls(lam))
}


#LS estimator and its risk
#########################
fit.ls <- fit.pls(0)
etahat.ls <- fit.ls$etahat
ss <- normsq(y - etahat.ls)/(n-q)
risk.ls <- risk.pls(0)
```

References