STA242 Proposal for final project

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

April 24, 2015

1 What we are trying to do

We would like to write a R package for the *Hypercube estimators* [1] introduced by Rudolf Beran.

1.1 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 estimator $\hat{\eta}$ so that the risk $\mathrm{E}|\hat{\eta} - \eta|^2$ is minimized. Here, we consider two types of estimators:

1. Submodel least squares estimators:

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

2. Penalized least squares estimator

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

where

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

Now, define the hypercube estimator of η to be

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

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

It can be shown that $\hat{\eta}_{\text{sub}}$ and $\hat{\eta}_{\text{pls}}$ can be obtained from $\hat{\eta}_{\text{H}}(V)$ with V carefully chosen. In this sense, hypercube estimator extend the above frameworks.

Moreover, restricting V into some suitable subclasses, we can compute the V shown that the risk $\mathrm{E}|\hat{\eta}_{\text{H}}(V) - \eta|^2$ is minimized in those subclasses. Thus, we obtain estimators which is much better than the least square estimator.

Another advantage of hypercube estimator $\hat{\eta}_{\text{H}}(V)$ is that it is more numerically stable than the original form of $\hat{\eta}_{\text{sub}}$ and $\hat{\eta}_{\text{pls}}$.

1.2 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.

1.3 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 .

1.4 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.

2 For each of the following, enumerate specifics about which you might use

2.1 Software

- We may call `C/C++` library in order to get faster computation and processing on large data.
- We may also use `RStudio` because it has convenient features for building package.

References

[1] Rudolf Beran. Hypercube estimators: Penalized least squares, submodel selection, and numerical stability. *Computational Statistics & Data Analysis*, 71:654–666, 2014.