STA242 Report: HyperCube

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

2015-06-10

- Bitbucket SSH: `git@bitbucket.org:taeyen/sta242_15_project.git`

Summary what we have done

We built a package, **HyperCube**, as an implementation of the Hypercube Estimator introduced by (Beran 2014). Hypercube estimator is a richer class of regularized estimators of linear model, which extends penalized least squares estimators with quadratic penalties. Hypercube estimator can be applied in various cases for fitting linear models. The R package **HyperCube** let R users conduct statistical analysis with Hypercube Estimator efficently and conveniently.

- We implemented the Hypercube Estimator with S3 classes and methods. We built a "formula" interface for the Hypercube Estimator, which is similar to other linear model fitting functions in R.
- Along with the classes and methods, we wrote helps files with Roxygen2. We also wrote vignettes with Rmarkdown. With the help files and vignettes, users can self-learn how to use the package.
- We also included some data sets used in (Beran 2014) so that users can try out the examples easily.
- We wrote some tests with testthat to ensure the package works properly.

Some Examples

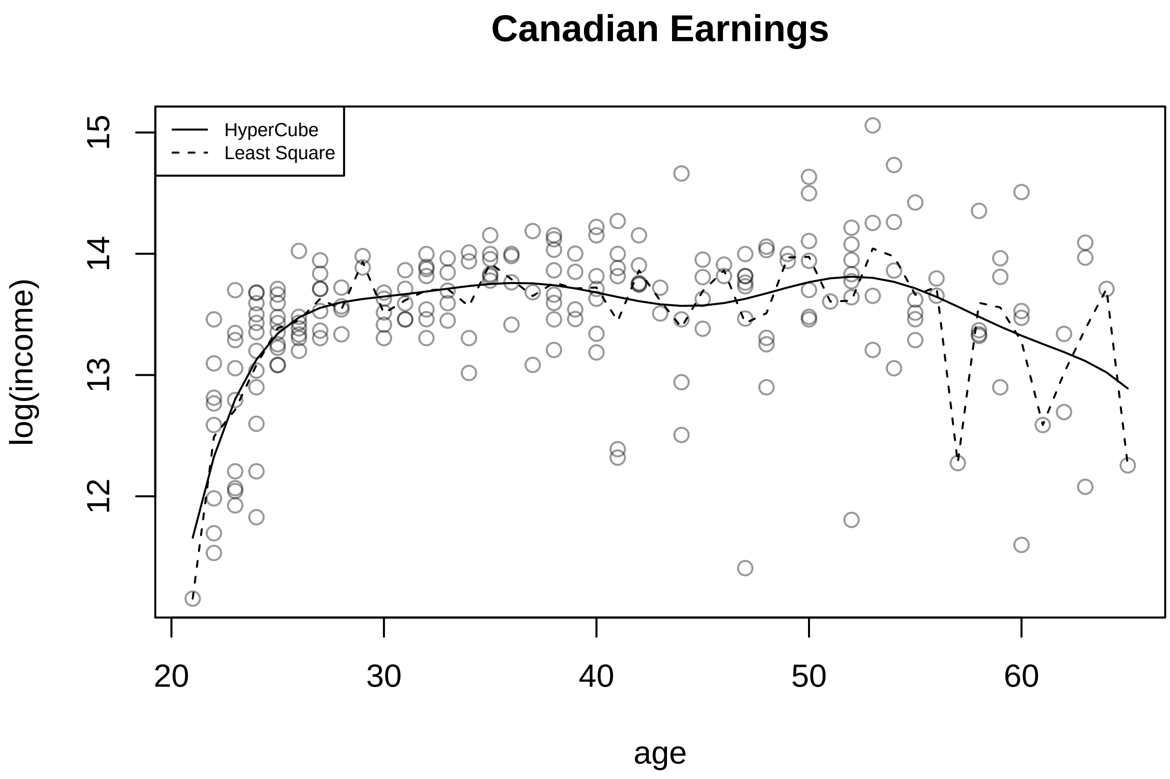
In this section, we shows some examples on how to use this package. Here, we skip the theoretical background. Nevertheless, we wrote up a vignette which introduces theoretical background along with the examples. Users can access the vignette by `vignette("introduction", package="HyperCube")`.

Example 1

Example 1 in (Beran 2014)

```r
library(HyperCube)

# The package includes the data set canadian.earnings.
# The age is considered as factor in the data set.
canadian.earnings$age <- factor(canadian.earnings$age)

# Plot the data
plot(as.numeric(as.character(canadian.earnings$age)),
     canadian.earnings$log.income,
     xlab = "age", ylab = "log(income)",
     main = "Canadian Earnings",
     col = rgb(0, 0, 0, 0.4)
    )
```

```r
# The number of ages in the data set, p, in Example 1 in Beran (2014).
p <- length(unique(canadian.earnings[,1]))

# D_5 as in equation (3.10) in Beran (2014)
D <- diffMatrix(p, 5)

# The matrix W in equation (3.11) in Beran (2014)
W <- 10000 * t(D) %*% D

# Convert W to V, as described in (1.6) in Beran (2014)
V <- plsW2V(W)

# Hyperpercube Estimator Fit
hcmod <- hypercube( log.income ~ age -1, data=canadian.earnings, V)

# Plot the Hypercube Estimator fit
lines(as.numeric(levels(canadian.earnings$age)),
      hcmod$coefficients, lty = 1)

# Least square fit
lmmod <- lm( log.income ~ age -1, data=canadian.earnings)

# Plot the Least saure fit
lines(as.numeric(levels(canadian.earnings$age)),
      lmmod$coefficients, lty = 2)

# Legend
legend("topleft", cex = 0.6,
       legend = c("HyperCube", "Least Square"),
       lty = c(1,2))
```



Example 2

Example 2 in (Beran 2014)

```
library(HyperCube)

# The package includes the data set litter.
# The formula specifying the two-way layout is "weight ~ mother:infant -1".
# hypercubeOptimization computes the optimal d
hcmodopt <- hypercubeOptimization( weight ~ mother:infant -1, data = litter)

# The optimal d
# Same result as stated in Example 2 in Beran (2014)
hcmodopt$projcoef
```

```
## [1] 0.9971043 0.6931901 0.0000000 0.4150027
```

```
# Compare the estimated risk
summary(hcmodopt$est)
```

```
## Call:
## hypercube.formula(formula = formula, data = data, V = V)
##
## The estimated risk of hypercube estimation: 16.1214335561813
## The estimated risk of least square estimation:  54.2403666666667
```

Future Works

Due to time constraint, we have only implemented some parts of the paper (Beran 2014): Penalized weighted least square and Shrinkage on ANOVA submodels. In future, we may also implement more parts, for example polynomial submodel fits, projection matrices from eigen-decomposition of matrix, Stein multiple shrinkage, etc.

One of the greatest challenge in implementing the Hypercube estimator is the design of the classes and methods. Although the mathematics of Hypercube Estimator is simple, the unification of different applications of Hypercube Estimator is not obvious. We admit that the current design is not perfect. The information stored in the objects is not compact, wasting some memory. Many codes have not been not optimized. Also, the user inferface with the "formula" class is not simple enough. The output of the fitted model is not informative enough. Last but not least, some of the functions are not properly documented. If time allows, we should improve the design of the classes and methods, optimize some of the codes, and write better documentations.

References

Beran, Rudolf. 2014. "Hypercube Estimators: Penalized Least Squares, Submodel Selection, and Numerical Stability." *Computational Statistics & Data Analysis* 71. Elsevier: 654–66.