Ensemble Methods using CGSS2015

郑军威

目录

本文使用 CGSS2015 数据，对成人总收入进行预测，自变量包括性别、教育年限、户口、业余学习时间、与邻居交往频率、是否参加工会、工作经历、自我感觉家庭收入水平。

```
library(MASS)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(rpart)
library(rpart.plot)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(gbm)
```

```
## Loaded gbm 2.1.5
```

```
rm(list=ls())
library(haven)
getwd()
```

```
## [1] "D:/Data Analysis for Economics/homework/hw5"
```

```
setwd("D:/Data Analysis for Economics/homework/hw5")
library(haven)
abc <- read_dta("cgss2015_8vars.dta")
```

```
set.seed(100)
train = sample(nrow(abc), nrow(abc)*0.6)
data_train = abc[train,]
data_test = abc[-train,]
ytrue=data_test$ln_income
```

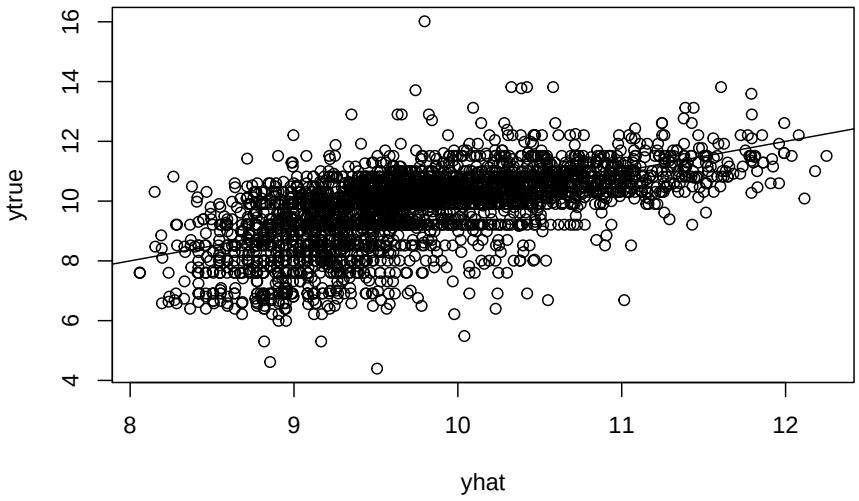
1 Linear Regression

```
fit = lm(ln_income~.,data_train)
summary(fit)
```

```
##
## Call:
## lm(formula = ln_income ~ ., data = data_train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2057 -0.5244  0.1132  0.6510  6.9945
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.878344   0.119574  74.250  < 2e-16 ***
## gender      -0.268480   0.030086  -8.924  < 2e-16 ***
## edu          0.088184   0.006418  13.739  < 2e-16 ***
## hukou        0.159508   0.012392  12.872  < 2e-16 ***
## study        0.122616   0.017155   7.148 1.02e-12 ***
## neighbor     0.045011   0.007842   5.740 1.01e-08 ***
## union       -0.177651   0.024441  -7.269 4.25e-13 ***
## experience  -0.130708   0.009531 -13.714  < 2e-16 ***
## level        0.358115   0.021917  16.340  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.009 on 4609 degrees of freedom
## Multiple R-squared:  0.3721, Adjusted R-squared:  0.371
## F-statistic: 341.5 on 8 and 4609 DF,  p-value: < 2.2e-16
```

```r
# test error
yhat = predict(fit,data_test)
plot(yhat,ytrue)
abline(0,1)
```

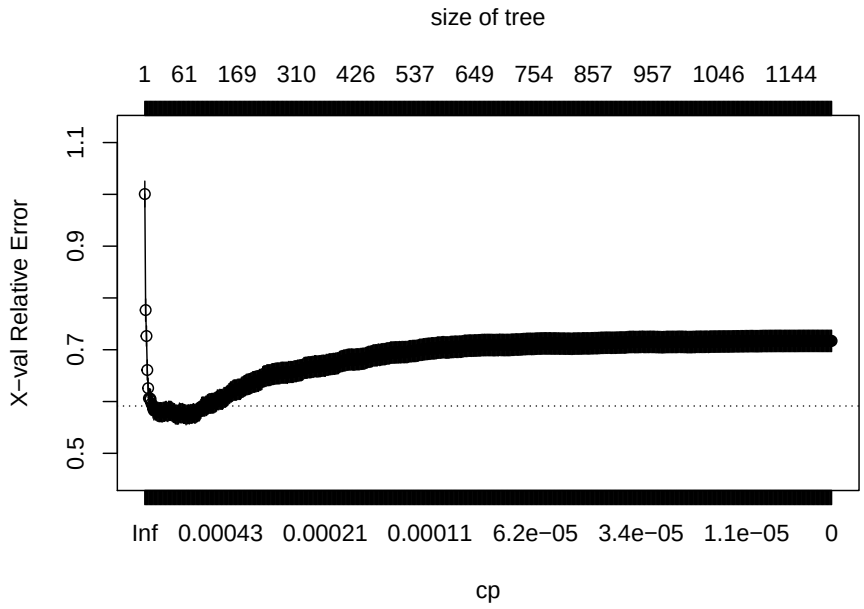



```
mean((yhat-ytrue)^2)
```

```
## [1] 0.9879113
```

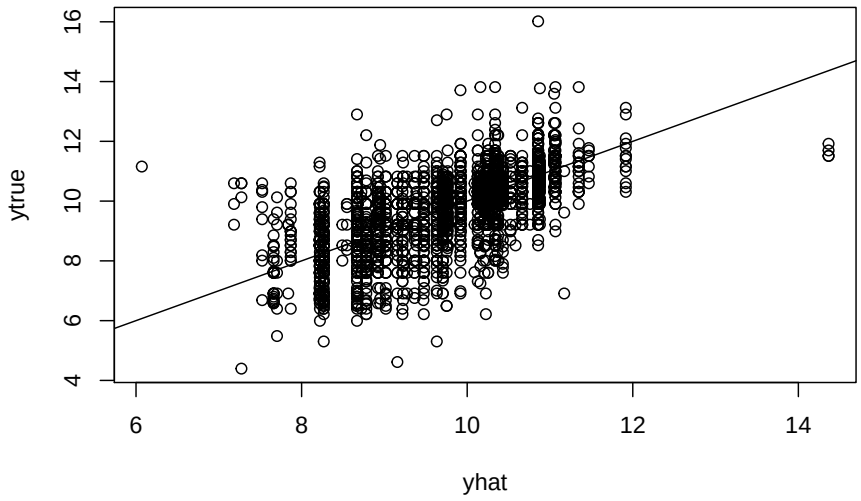
2 Regression Tree

```
set.seed(100)
fit0 = rpart(ln_income~.,data_train,
             control=rpart.control(cp=0,minbucket=2))
plotcp(fit0)
```




```r
# prune
fit = prune(fit0,cp=fit0$cptable[which.min(fit0$cptable[,"xerror"]),"CP"])

# test error
yhat = predict(fit,data_test)
plot(yhat,ytrue)
abline(0,1)
```



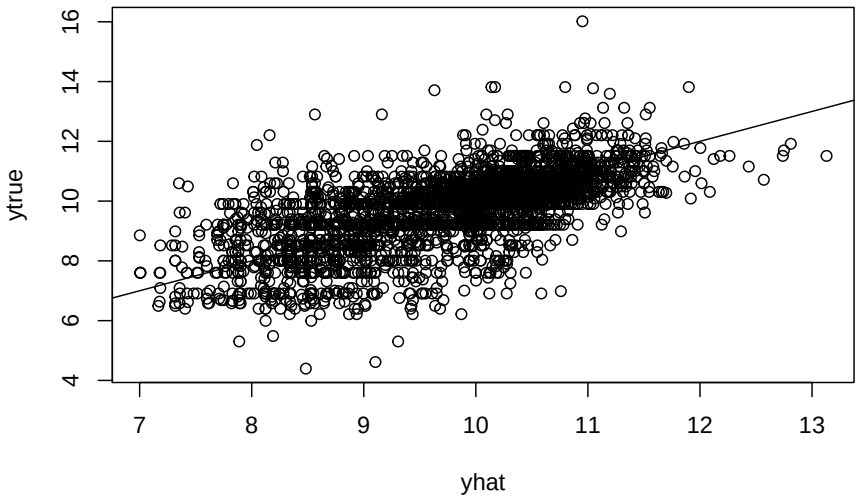

```
mean((yhat-ytrue)^2)
```

```
## [1] 0.9067623
```

3 Bagging

```
set.seed(100)
fit = randomForest(ln_income~.,data_train,mtry=8,importance =TRUE)

# test error
yhat = predict(fit,data_test)
plot(yhat,ytrue)
abline(0,1)
```



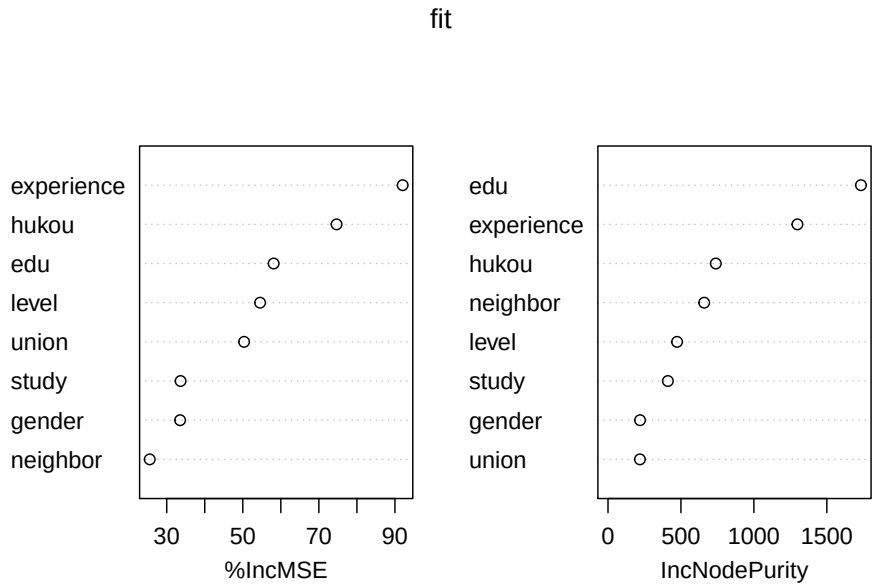

```
mean((yhat-ytrue)^2)
```

```
## [1] 0.9260671
```

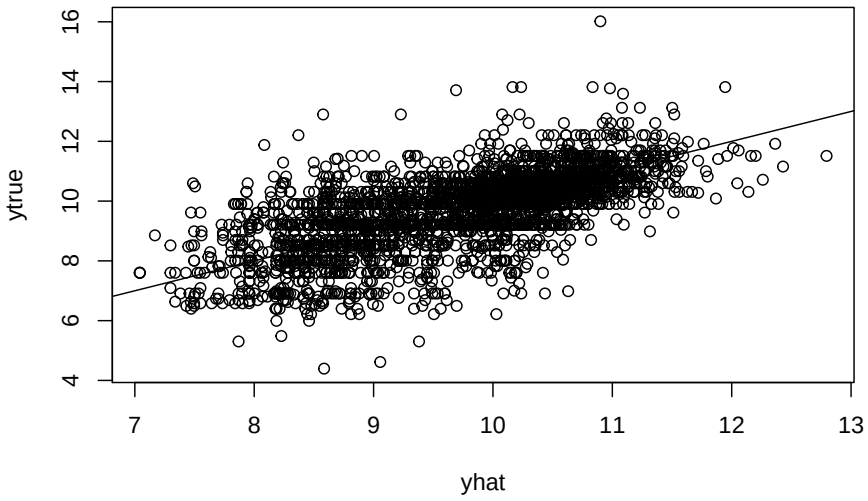
4 Random Forest

```
set.seed(100)
fit = randomForest(ln_income~.,data_train,mtry=5,importance =TRUE)

# variable importance and partial dependence plots
varImpPlot(fit)
```




```
# test error
yhat = predict(fit,data_test)
plot(yhat,ytrue)
abline(0,1)
```

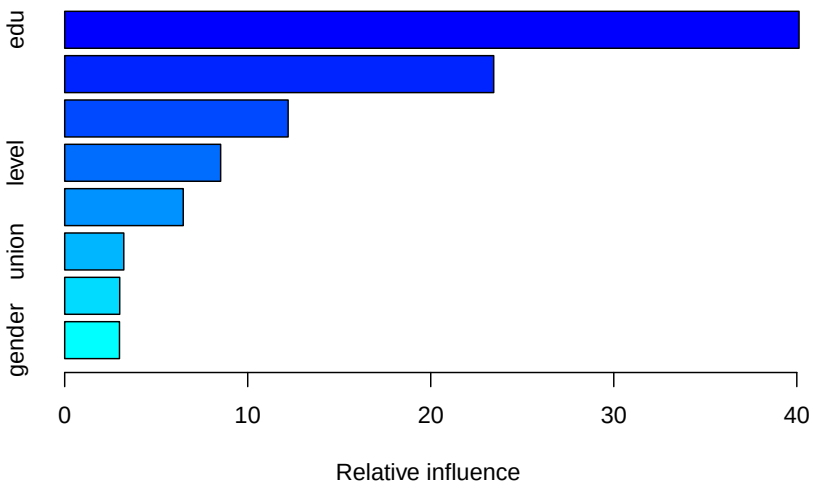



```
mean((yhat-ytrue)^2)
```

```
## [1] 0.8910578
```

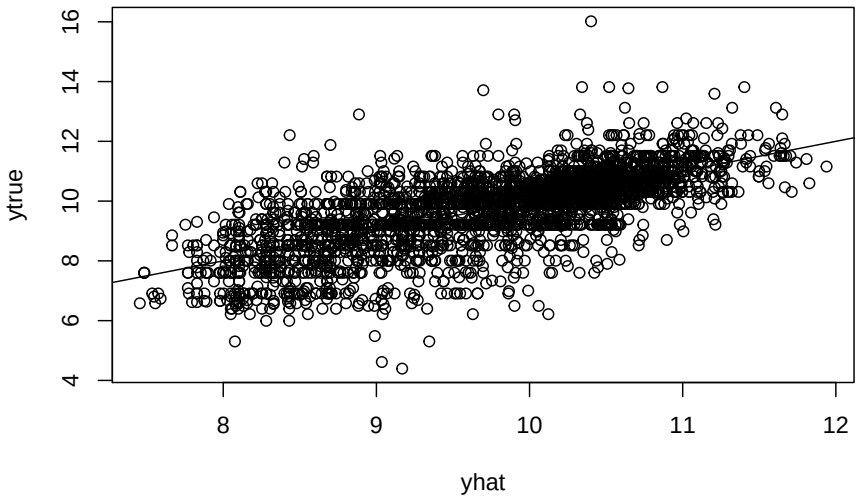
5 Boosting

```
set.seed(100)
fit = gbm(ln_income~.,data=data_train, distribution="gaussian",
          n.trees=10000,interaction.depth=5,shrinkage=0.001)
summary(fit)
```




```
##                   var   rel.inf
## edu               edu 40.122115
## experience experience 23.440201
## hukou           hukou 12.207491
## level           level  8.521822
## study           study  6.478044
## union           union  3.229409
## neighbor     neighbor  3.006561
## gender         gender  2.994357
```

```
# test error
yhat = predict(fit,data_test,n.trees=10000)
plot(yhat,ytrue)
abline(0,1)
```




```
mean((yhat-ytrue)^2)
```

```
## [1] 0.8070859
```

数据来源：zhengjunweizjw.github.io/cgss2015_8vars.dta