Statistical Learning **Deadline: November 7, 2015**

Homework 10. Tree ensembles: Boosting, Bagging, and Random Forest

Instructor: Yuan Yao

For the experimental problem, include the source codes (as Appendix) which are runnable under standard settings. On the head of your submitted homework, please mark *NAME - student ID*.

By (ESL), please refer to the Elements of Statistical Learning, Edition II, the 10^{th} print

`http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf`

By (ISLR), please refer to An Introduction to Statistical Learning, with applications in R,

`http://www-bcf.usc.edu/~gareth/ISL/`

1. Draw an example (of your own invention) of a partition of two-dimensional feature space that could result from recursive binary splitting. Your example should contain at least six regions $(R_1, R_2, \ldots, R_6, \ldots)$. Draw a decision tree corresponding to such a partition. Be sure to label corresponding regions and splitting points.
2. (ESL) Exercise 9.5
3. (ISLR) Explore the `Carseats` dataset (`library("ISLR")`) using classification tree, as Section 8.3.1 in ISLR shown in the following R commands.

```
# 8.3 in the book
# An Introduction to Statistical Learning

# Remember to install.packages("ISLR") before load it
library("ISLR")
attach(Carseats)

# Construct a binary response variable "High"
High = ifelse(Sales<=8,"No","Yes")
Carseats = data.frame(Carseats,High)

# Remember to install.packages("tree") before load it
library("tree")

tree.carseats=tree(High~.-Sales,Carseats)
summary(tree.carseats)

# Plot the tree and show the node labels
plot(tree.carseats)
text(tree.carseats, pretty=0)

# Print the tree on the screen.
tree.carseats
```

```
# Split the dataset into a training set and a test set.
set.seed (2)
train=sample(1:nrow(Carseats), 200)
Carseats.test=Carseats [-train ,]
High.test=High[-train]

# Train a model, get the test error,
tree.carseats=tree(High~.-Sales,Carseats,subset=train)
tree.pred=predict(tree.carseats,Carseats.test,type="class")

# Get the confusion matrix
table(tree.pred, High.test)
# Compute the prediction accuracy, which is 71.5%
sum(diag(table(tree.pred, High.test)))/200
# Use CV with misclassification error minimization to prune the tree
set.seed(3)
cv.carseats =cv.tree(tree.carseats ,FUN=prune.misclass )
names(cv.carseats )

# Note: cv.carseats$dev gives the misclassification rates
cv.carseats

# Error Plots against "size" and "k"
par(mfrow=c(1,2))
plot(cv.carseats$size, cv.carseats$dev, type="b")
plot(cv.carseats$k, cv.carseats$dev, type="b")

# Get the optimally pruned tree and plot it
prune.carseats=prune.misclass(tree.carseats,best=9)
plot(prune.carseats)
text(prune.carseats, pretty=0)

# New test performance, which is 77%
table(tree.pred, High.test)
sum(diag(table(tree.pred,High.test)))/200
```

4. (ISLR) Fit the real variable `Sales` in `Carseats` dataset without converting it to a binary one, using regression tree.

5. Explore the `Carseats` dataset with Random Forest (`library("randomForest")`), Bagging, and Boosting (`library("gbm")`). Compare your results with above.