COMS 4771 Machine Learning
Problem Set #2

Jun Hu - jh3846@columbia.edu
Discussants:

July 12, 2017

Problem 1

(a) Execute:

```
library(tree)
library(ISLR)
set.seed(2388)
train = sample(1:nrow(OJ), 800)
OJ.train = OJ[train, ]
OJ.test = OJ[-train, ]
```

(b) Fit and summarize:

```
tree.oj = tree(Purchase~., data=OJ.train)
summary(tree.oj)
```

Will return:

```
Classification tree:
tree(formula = Purchase ~ ., data = OJ.train)
Variables actually used in tree construction:
[1] "LoyalCH"       "PriceDiff"     "StoreID"       "ListPriceDiff" "PctDiscMM"
Number of terminal nodes:  10
Residual mean deviance:  0.7451 = 588.6 / 790
Misclassification error rate: 0.1662 = 133 / 800
```

From the results, we can obtain that the classification tree actually uses only 5 variables, which are LoyalCH, PriceDiff, StoreID, ListPriceDiff, PctDiscMM. The tree has 10 terminal nodes, and the misclassification error rate is 0.1662.

(c) Execute:

```
tree.oj
```

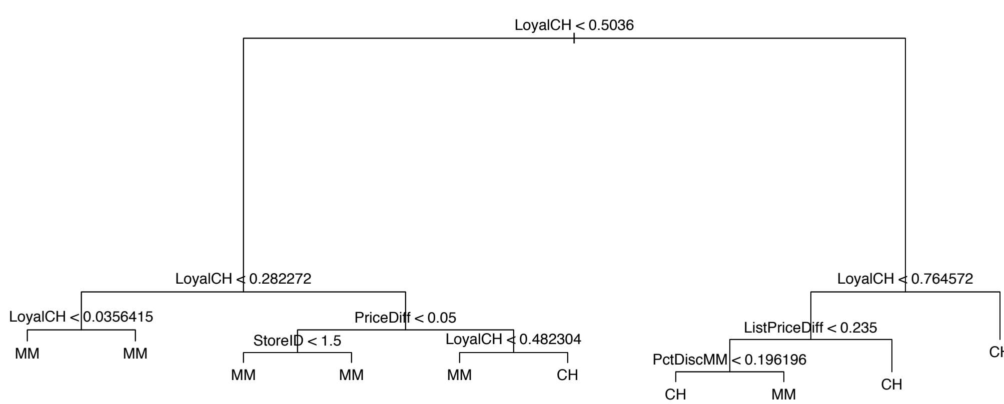
```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 800 1072.00 CH ( 0.60750 0.39250 )
   2) LoyalCH < 0.5036 349  418.10 MM ( 0.28653 0.71347 )
     4) LoyalCH < 0.282272 174  135.90 MM ( 0.13218 0.86782 )
       8) LoyalCH < 0.0356415 56   10.03 MM ( 0.01786 0.98214 ) *
       9) LoyalCH > 0.0356415 118  113.50 MM ( 0.18644 0.81356 ) *
     5) LoyalCH > 0.282272 175  240.10 MM ( 0.44000 0.56000 )
      10) PriceDiff < 0.05 70   72.74 MM ( 0.21429 0.78571 )
        20) StoreID < 1.5 19    0.00 MM ( 0.00000 1.00000 ) *
        21) StoreID > 1.5 51   61.79 MM ( 0.29412 0.70588 ) *
      11) PriceDiff > 0.05 105  142.10 CH ( 0.59048 0.40952 )
        22) LoyalCH < 0.482304 75  104.00 MM ( 0.49333 0.50667 ) *
        23) LoyalCH > 0.482304 30   27.03 CH ( 0.83333 0.16667 ) *
   3) LoyalCH > 0.5036 451  372.00 CH ( 0.85588 0.14412 )
     6) LoyalCH < 0.764572 192  228.10 CH ( 0.71875 0.28125 )
      12) ListPriceDiff < 0.235 74  102.40 MM ( 0.47297 0.52703 )
        24) PctDiscMM < 0.196196 58   79.30 CH ( 0.56897 0.43103 ) *
        25) PctDiscMM > 0.196196 16   12.06 MM ( 0.12500 0.87500 ) *
      13) ListPriceDiff > 0.235 118   89.89 CH ( 0.87288 0.12712 ) *
     7) LoyalCH > 0.764572 259   91.02 CH ( 0.95753 0.04247 ) *
```

Let's pick the terminal node (marked with *) labeled as 8). It shows that the split criterion is LoyalCH < 0.0356415, the number of observations in this branch is 56, the deviance is 10.03, the overall prediction for the branch is MM for Purchase, and 98.214% observations in this branch is MM for Purchase while 1.786% is CH for Purchase.

(d) Execute:

```r
plot(tree.oj)
text(tree.oj, pretty=0)
```

The plot is:



This plot indicates the importance of the variable LoyalCH. Because the first level split criterion is LoyalCH $\leqslant 0.5036$, then in the second level the two nodes still splits on LoyalCH (LoyalCH $\leqslant 0.282272$, LoyalCH $\leqslant 0.764572$).

(e) Execute:

Predict and produce a confusion matrix:

```
tree.pred = predict(tree.oj, OJ.test, type="class")
table(tree.pred, OJ.test$Purchase)
```

```
tree.pred  CH  MM
       CH 141  17
       MM  26  86
```

Calculate the test error rate:

```
mean(tree.pred!=OJ.test$Purchase)
```

```
[1] 0.1592593
```

So the test error rate is 15.93%.

(f) Execute cv.tree function as:

```
cv.oj = cv.tree(tree.oj, FUN=prune.misclass)
```

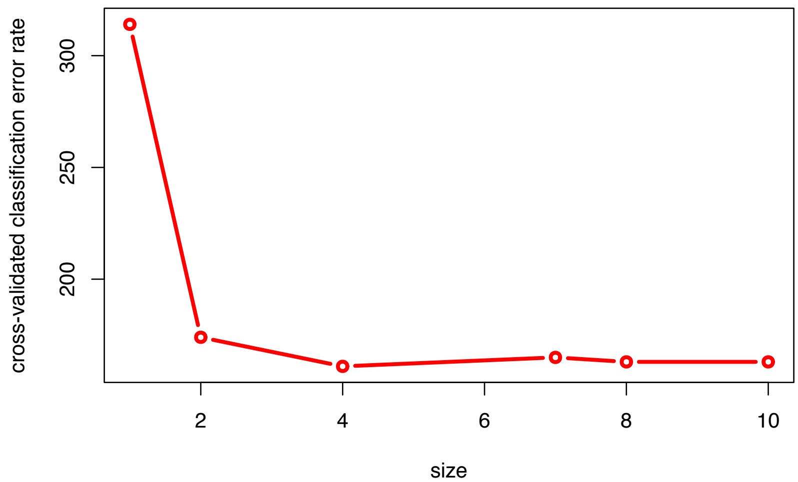
Check results:

```
cv.oj
```

```
$size
[1] 10  8  7  4  2  1

$dev
[1] 163 163 165 161 174 314

$k
[1]  -Inf   0.0   1.0   4.0   9.5 149.0

$method
[1] "misclass"

attr(,"class")
[1] "prune"         "tree.sequence"
```

(g) Plot as:

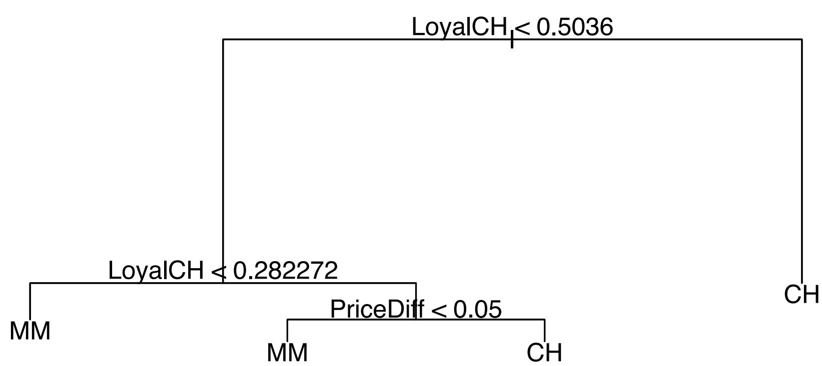
```
plot(cv.oj$size, cv.oj$dev, type='b', col="red", ylim=c(160, 315), lwd=3,
    xlab='size', ylab='cross-validated classification error rate')
```



(h) From **(f)** and **(g)** we can get the tree size of 4 returns the lowest cross-validated classification error rate.

(i) Prune the tree as:

```
prune.oj = prune.misclass(tree.oj, best=4)
plot(prune.oj)
text(prune.oj, pretty=0)
```



(j) We can obtain the training error for pruned tree:

```
summary(prune.oj)
```

```
Classification tree:
snip.tree(tree = tree.oj, nodes = c(4L, 10L, 11L, 3L))
Variables actually used in tree construction:
[1] "LoyalCH"   "PriceDiff"
Number of terminal nodes:  4
Residual mean deviance:  0.9079 = 722.7 / 796
Misclassification error rate: 0.1825 = 146 / 800
```

Compared to the unpruned tree training error rate is 0.1662, the pruned training error rate 0.1825 is higher.

To predict test data using pruned tree:

```
summary(prune.oj)
prune.pred = predict(prune.oj, OJ.test, type="class")
table(prune.pred, OJ.test$Purchase)
```

```
prune.pred  CH  MM
        CH 155  34
        MM  12  69
```

Calculate the pruned tree test error rate:

```
mean(prune.pred!=OJ.test$Purchase)
```

```
[1] 0.1703704
```

The pruned tree test error rate is 0.1703704, compared to the unpruned test error rate 0.1592593, the pruned tree test error rate is higher.

Problem 2

For the given logistics curve $\sigma(a)$, $\exists$:

$$
\begin{aligned}
\frac{\mathrm{d}\sigma(a)}{\mathrm{d}a} &= \frac{\mathrm{d}}{\mathrm{d}a}\frac{1}{1+e^{-a}} \\
&= 1 \times \frac{1}{2} \times (1 - \frac{1}{2}) \\
&= \frac{1}{4}
\end{aligned}
$$

On the other hand, for the given function $\Phi(a)$, $\exists$:

$$
\begin{aligned}
\frac{\mathrm{d}\Phi(\lambda a)}{\mathrm{d}a} &= \frac{\mathrm{d}}{\mathrm{d}a}\int_{-\infty}^{\lambda a} \mathcal{N}(\theta|0,1)\mathrm{d}\theta \\
&= \frac{\mathrm{d}}{\mathrm{d}a}\int_{-\infty}^{\lambda a} \left(\frac{1}{\sqrt{2\pi}}e^{-\frac{\theta^2}{2}}\right)\mathrm{d}\theta \\
&= \frac{\mathrm{d}}{\mathrm{d}a}(\lambda a)\frac{1}{\sqrt{2\pi}}e^{-\frac{(\lambda a)^2}{2}} \\
&= \lambda\frac{1}{\sqrt{2\pi}}e^{-\frac{(\lambda a)^2}{2}}
\end{aligned}
$$

Two functions are equal at $a = 0$, $\exists$:

$$
\lambda\frac{1}{\sqrt{2\pi}}e^{0} = \frac{1}{4}
$$

That is

$$
\begin{aligned}
\lambda &= \frac{\sqrt{2\pi}}{4} \\
\lambda^2 &= \frac{\pi}{8}
\end{aligned}
$$

Problem 3

Because the Gaussian function in d-dimension ($\mathcal{X} = \mathbb{R}^d$) is:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

And we have known that $p(t|x,w) = \mathcal{N}(t|y(x,w),\Sigma)$, so the error function (log form) is:

$$\begin{aligned}
E(\mathbf{w}) &= \sum_{i=1}^{N} E_i(\mathbf{w}) \\
&= \sum_{i=1}^{N}(-\log p(\mathbf{t_i}|\mathbf{x_i},\mathbf{w})) \\
&= -\sum_{i=1}^{N}\log \mathcal{N}(\mathbf{t_i}|\mathbf{y}(\mathbf{x_i},\mathbf{w}),\boldsymbol{\Sigma}) \\
&= -\sum_{i=1}^{N}\left[\left(-\frac{1}{2}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))\right)\log e + \log\frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}\right] \\
&= \frac{1}{2}\sum_{i=1}^{N}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))\log e - \sum_{i=1}^{N}\log\frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|}}
\end{aligned}$$

Because $\boldsymbol{\Sigma}$ is fixed and known, we can clear the error function by removing items irrelevant of $\mathbf{w}$. The final form of the error function that must be minimized in order to find the maximum likelihood solution for $\mathbf{w}$ is:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))^T\boldsymbol{\Sigma}^{-1}(\mathbf{t_i}-\mathbf{y}(\mathbf{x_i},\mathbf{w}))$$