In-Class Computing Task 17

Math 253: Statistical Computing & Machine Learning

Growing and Pruning Trees

In this activity, you're going to investigate the process of growing trees and pruning them. You'll use some cartoon-like data:

```
Cartoon_data <- data.frame(
  x = 1:8,
  y = c(2,5,1,3,8,5,4,6),
  class = c("A", "B", "A", "A", "B", "B", "A", "B")
)
```

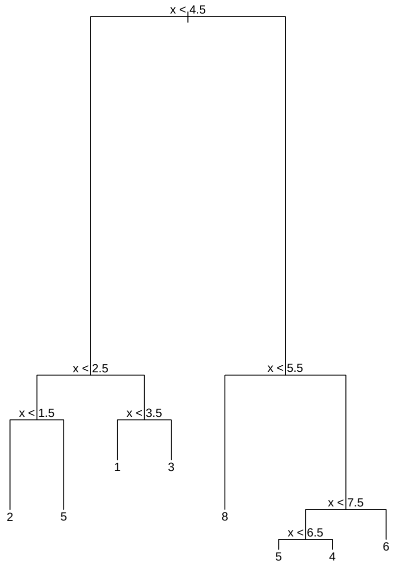
The software you'll be using is in the `tree` package.

```
library(tree)
```

Perfectly pure trees

You're first going to construct trees where every *leaf* (that is, "terminal node") perfectly matches the data. Like most perfect fits, this sort of pure tree is likely an overfit to the data with a sub-optimal out-of-sample fit. To achieve this, you'll set the controls for growing the trees to values that push the trees to grow until the in-sample fit is perfect.

```
pure <- tree.control(8, mincut = 0, minsize = 1, mindev = 0)
```



Regression tree

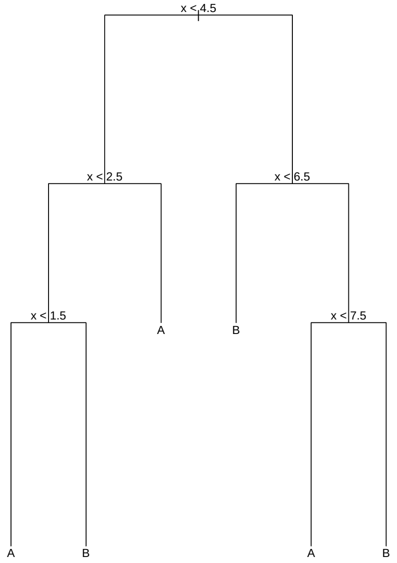
```
rtree_pure <- tree(y ~ x, data=Cartoon_data, control = pure)
plot(rtree_pure)
text(rtree_pure)
```

Traversing the tree visually, find the output for $x = 3$.

Classification tree

```
ctree_pure <- tree(class ~ x, data=Cartoon_data, control = pure)
plot(ctree_pure)
text(ctree_pure)
```

Traversing the tree visually, find the output for $x = 7$.



Evaluating the tree

As with other model-building programs, the R object returned by `tree()`, which is of class `tree`, has a predict function, `predict.tree()`. It is this special version that is invoked when you apply `predict()` to an object of class `tree`.

```
predict(rtree_pure)
predict(ctree_pure)
predict(rtree_pure, newdata = data.frame(x = 3))
predict(ctree_pure, newdata = data.frame(x = 7))
```

Deviance of each node

The printed version (e.g. `print(ctree_pure)`) of the tree gives the deviance of each node. For both `ctree_pure` and `rtree_pure`, annotate the graph of the tree (printed on page 1) with the deviance of each node. That is, just read the printed report and copy with pen or pencil the reported deviances to the to the appropriate node.

Then, label each split according to how much deviance it eliminates. This is the difference between the deviance at the node and the sum of the deviances of the two sub-nodes.

Deviance of a tree

Add up the deviance of all of the terminal nodes.

You can use `predict()` to get the predicted values from a regression tree and calculate the tree's deviance as the sum of squares of residuals.

For a classification tree, the deviance is -2 times the log likelihood of the actual classes when evaluated on the probability of that class at the appropriate node.

Pruning the tree

The pure trees have a node for each of the in-sample values of the response variable. Not all the splits may be worthwhile in terms of out-of-sample prediction. Pruning is the process of eliminating splits with cause only a small reduction in deviance, that is, eliminating splits of minor importance.

Think of one pruning cut as eliminating a split that leads to two terminal nodes. Once eliminated, the node that was previously split itself becomes a terminal mode.

In pruning, you make the pruning cut at the node that has the smallest deviance. Remember, consider only candidate splits that lead to two terminal nodes.

After each pruning cut, you are left with a tree with one fewer terminal node. Thus, there is a natural sequence of trees with different numbers of terminal nodes. There are two functions provided to prune trees, one for regression trees and one for classification trees.

```
rtree_5 <- prune.tree(rtree_pure, best = 5)
ctree_2 <- prune.misclass(ctree_pure, best = 2)
```

- Create a vector `nterminal` that is `2:8`
- Create a vector `tree_deviance` that holds the deviance of the regression tree for each value of `nterminal`.

Building a real classifier

Build a pure tree classifier for the `sector` variable in `mosaicData::CPS85` with predictors `wage`, `sex`, `educ`, and `experience`.

```
pure_for_cps <- tree.control(nrow(CPS85), mincut = 0, minsize = 1, mindev = 0)
Sector_classifier <- tree(sector ~ wage + sex + educ + exper,
                          data = mosaicData::CPS85,
                          control = pure_for_cps)
```

Prune the tree to about 20 terminal nodes. What's the tree deviance? What does the tree look like?