Decision Tree Learning using RWeka

CIS400/600
Fundamentals of Data and Knowledge Mininng

Installing RWeka

1. Install the latest version of `JDK`.
   (`https://www.oracle.com/technetwork/java/javase/downloads/index.html`)
2. *Only for OS X*. Install `Java For OS X`.
   (`https://support.apple.com/kb/DL1572`)
3. *Only for OS X*. If you had installed `R` using homebrew, update `R` from `https://cran.r-project.org/bin/macosx/`. The homebrew version seems to be outdated.
4. Install `rJava` and `RWeka`.
   ```
   > install.packages("rRava", type = "source")
   > install.packages("RWeka", type = "source")
   ```
5. Install `pROC`.
   ```
   > install.packages("pROC")
   ```

Decision Tree Learning

For this exercise we are going to use the 1994 census data to predict income levels - >=50K and <50K.

1. Load `RWeka`.
   ```
   > library(RWeka)
   ```
2. Load the learning data `adult_train.csv`[1].
   ```
   > adult.train <- read.csv("adult_train.csv")
   ```
3. For this exercise, we are going to use `C4.5` decision tree. You can see the other types of decision tree available with,
   ```
   > ?Weka_classifier_trees
   ```
4. All the classifiers requires the following arguments:

[1] More information about the data at `https://archive.ics.uci.edu/ml/datasets/Adult`

(a) **formula** - If we want to predict `X` based on attributes `Y` and `Z`, the formula takes the form of `X ~ Y + Z`. If we want to predict `X` based on all the attributes, you can use the shorthand `X ~ .` [2]

(b) **data** - This is the training data.

(c) **control** - This is used to set various parameters for the decision tree. You can view the available controls with,

```
> WOW(J48)
```

5. Train the decision tree classifier. We are going to set the minimum number of instances per leaf as `10` and the pruning confidence threshold to `0.5`. We are also going to allow splits. As mentioned in 4c, there are other control options you can play around with.

   ```
   > mdl <- J48(income ~ ., data = adult.train, control = Weka_control(
   M = 10, C = 0.25, B = F))
   ```

6. Load the testing data `adult_test.csv` and test the model on the new data.

   ```
   > adult.test <- read.csv("adult_test.csv")
   > evaluate_Weka_classifier(mdl, newdata = adult.test, class = T)
   ```

7. The accuracy of the fitted model on the test dataset should be around 77%.

8. To predict from an unknown dataset, you can use the `predict()` function.

Model Evaluation with k-Fold Cross Validation

In this section, we will look at performance evaluation using k-fold cross validation.

1. Consider two models `mdl.1` and `mdl.2`.

   ```
   > mdl.1 <- J48(income ~ ., data = adult.train, control = Weka_control(M
   = 2, U = T, B = T))
   > mdl.2 <- J48(income ~ ., data = adult.train, control = Weka_control(M
   = 2, U = F, B = F))
   ```

2. For this exercise, we will compare these two models with k-fold cross validation with $k = 10$.

   ```
   > evaluate_Weka_classifier(mdl.1, class = T, numFolds = 10)
   > evaluate_Weka_classifier(mdl.2, class = T, numFolds = 10)
   ```

3. The second model (`mdl.2`) should have a lower error rate. This suggest that `mdl.2` would perform than `mdl.1` on an unknown dataset. To verify, test both models on `adult.test`.

   ```
   > evaluate_Weka_classifier(mdl.1, class = T, newdata = adult.test)
   > evaluate_Weka_classifier(mdl.2, class = T, newdata = adult.test)
   ```

[2] More at: https://faculty.chicagobooth.edu/richard.hahn/teaching/FormulaNotation.pdf

4. Compare the accuracy of the two models. The first model should have an accuracy of less than 70% on the unseen data, and the second should have accuracy of more than 75%.

Performance Evaluation with ROC

In the previous sections we evaluate the classification performance using accuracy. In this section we will use ROC analysis to compare classification performance on `mdl.1` and `mdl.2`.

1. Load `pROC`.
   ```
   > library(pROC)
   ```
2. Get the predicted class probabilities of `adult.test` using `mdl.1`.
   ```
   > p.1 <- predict(mdl.1, newdata = adult.test, type = c("prob"))
   ```
3. Calculate the ROC.
   ```
   > roc.1 <- roc(adult.test$income, p.1[,1])
   ```
4. Plot the ROC curve [3].
   ```
   > plot(roc.1)
   ```
5. Plot the ROC curve for `mdl.2`. Compare the plot and AUC with that of `mdl.1`.

[3] `FPR = 1 - Specificity`