CS 6601: Machine Learning and Decision Trees.

Daniel Kohlsdorf, Titilayo Craig, Thad Starner

This homework consists of three assignments from your textbook again. You will construct decision trees and implement a decision tree as well as a random forrest capable of dealing with missing data.

!!!!!!! NEW LATE POLICY !!!!!!!

1) SUBMISSION VIA EMAIL DOES NOT COUNT AND WILL RECEIVE A : 0
2) ONLY T-SQUARE SUBMISSIONS WILL BE GRADED.
3) HANDING IN AFTER LATE DEADLINE IS EQUIVALENT TO NOT SUBMITTING
4) IF YOU NEED AN EXTENSION ASK US 2 DAYS IN ADVANCE

WARMUP 20%:

18.12 Construct a *decision list* to classify the data below. Select tests to be as small as possible (in terms of attributes), breaking ties among tests with the same number of attributes by selecting the one that classifies the greatest number of examples correctly. If multiple tests have the same number of attributes and classify the same number of examples, then break the tie using attributes with lower index numbers (e.g., select A_1 over A_2).

Example	A_1	A_2	A_3	A_4	y
$\mathbf{x}_1$	1	0	0	0	1
$\mathbf{x}_2$	1	0	1	1	1
$\mathbf{x}_3$	0	1	0	0	1
$\mathbf{x}_4$	0	1	1	0	0
$\mathbf{x}_5$	1	1	0	1	1
$\mathbf{x}_6$	0	1	0	1	0
$\mathbf{x}_7$	0	0	1	1	1
$\mathbf{x}_8$	0	0	1	0	0

PRACTICE 20%:

18.13 Prove that a decision list can represent the same function as a decision tree while using at most as many rules as there are leaves in the decision tree for that function. Give an example of a function represented by a decision list using strictly fewer rules than the number of leaves in a minimal-sized decision tree for that same function.

IMPLEMENTATION 60%:

18.9 The standard DECISION-TREE-LEARNING algorithm described in the chapter does not handle cases in which some examples have missing attribute values.

a. First, we need to find a way to classify such examples, given a decision tree that includes tests on the attributes for which values can be missing. Suppose that an example **x** has a missing value for attribute A and that the decision tree tests for A at a node that **x** reaches. One way to handle this case is to pretend that the example has *all* possible values for the attribute, but to weight each value according to its frequency among all of the examples that reach that node in the decision tree. The classification algorithm should follow all branches at any node for which a value is missing and should multiply the weights along each path. Write a modified classification algorithm for decision trees that has this behavior.

b. Now modify the information-gain calculation so that in any given collection of examples C at a given node in the tree during the construction process, the examples with missing values for any of the remaining attributes are given "as-if" values according to the frequencies of those values in the set C.

After you did the above, implement a random forrest, with the above properties, too. Now run an experiment. Download the following data set and write code to read it: Banknote Authentification.
Now write code to split the data using 10 fold cross validation. Furthermore, for each fold choose 10% of the data and remove different random attributes in these. In the end report accuracy, precision, recall as well as the confusion matrix for the three following cases:

1) Decision Tree / No Attributes Deleted
2) Random Forrest / No Attributes Deleted
3) Decision Tree / Attributes Deleted
4) Random Forrest Tree / Attributes Deleted

Hand in a description of your implementation, the code for the above algorithms, a description of your experiments as well as a description of your results.

LATE:

How could you use a decision tree to classify temporal data. What features would you use? How would you organize the data. Implement your solution for time series and try it on the following data set:

[TRAIN, TEST] from Eamonn Keogh' s time series collection