CS478 – Homework 2
Solutions

March 27, 2000

Problem 3.1

The four decision trees are depicted in Figure 1

Grading: 2.5 points for each of the trees. If part of the tree was wrong I subtracted some points (typically 1 point). Some of you didn't have decision trees with A and B as attributes with values True or False, 0 or 1, + or -. For such a solution you got about 3 out of 10 points (since is fundamentally wrong).

I took 0.75 points if the tree was not minimal (if part of the tree can be replaced with a leaf).

Problem 3.4

a)

The resulting tree is depicted in Figure 2. Observe that after one node we get directly to leafs. A split on any of the attributes *Sky*, *AirTemp* has maximum gain.

Grading: this part was 3 points for any of the 2 possible trees.

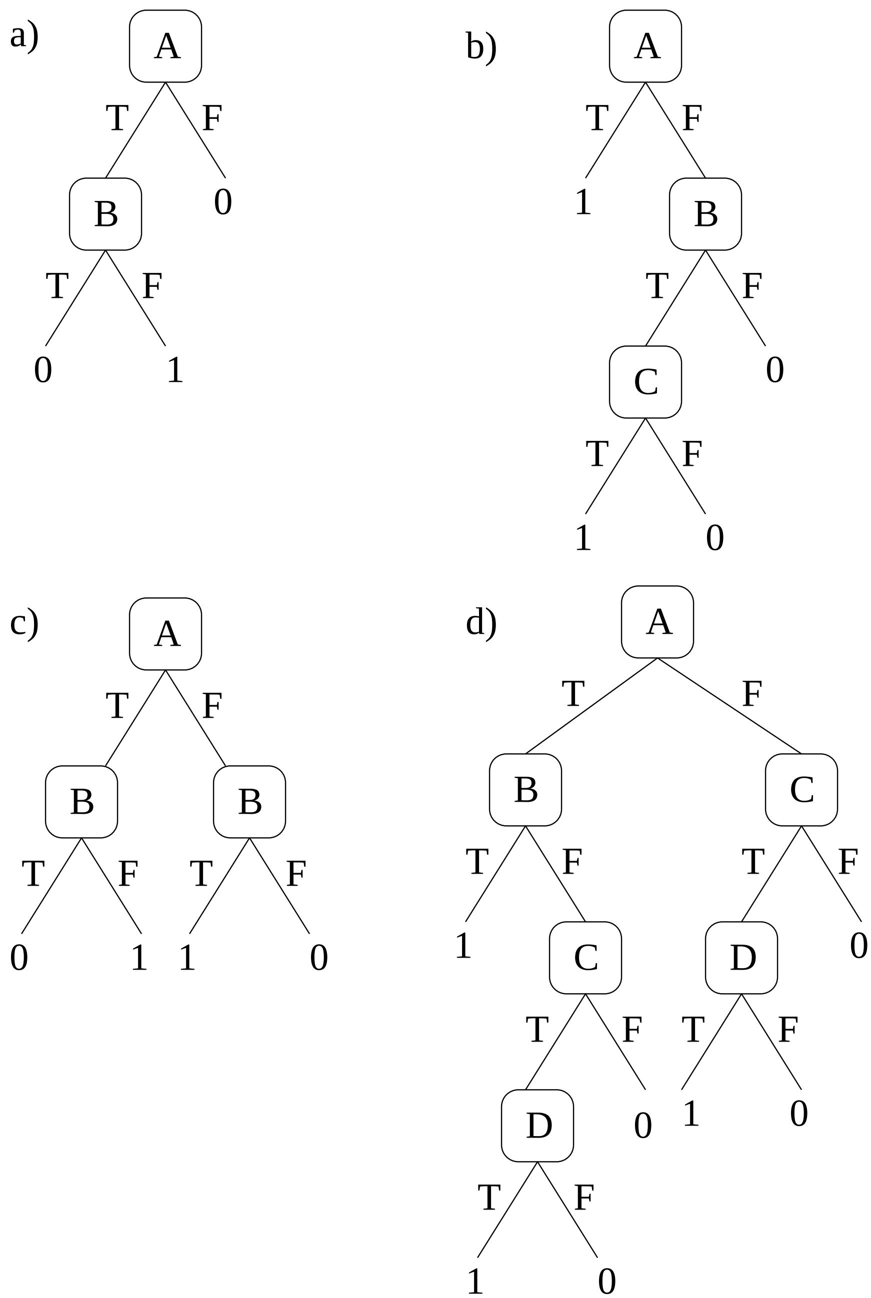

Figure 1: Solution for problem 3.1

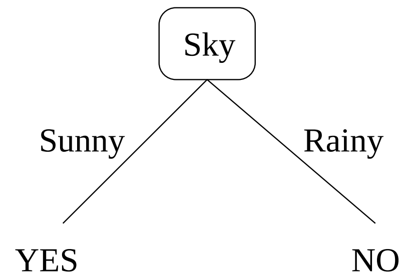

Figure 2: Solution for 3.4 a)

b)

The above decision tree corresponds to $<$ Sunny, ?, ?, ?, ?, ?, ? $>$ in the version space representation. Observe that this element is part of the most general border of the version space solution.

Grading: this part was 3 points. I took off 1–1.5 points if you were not pointing to the element of the version space the tree corresponds to, but you mentioned is one of the trees in G depending on the situation.

c)

The gain information at every possible branching is shown below. Using this information the result of the ID3 learning algorithm is the tree in Figure 3.

Root of the tree

At the root of the tree the entropy is 0.970951. The computation of the gain for each possible split is shown in table 1.

Looking at the gains for attributes we can choose to do the split on any of *Sky*, *AirTemp* or *Wind* so we chose to make the split on *Sky*.

Sunny **branch of** ***Sky***

Observe that the entropy of the *Rainy* branch is 0 so this is a leaf (labeled **NO**). The entropy of the *Sunny* branch is 0.811278. Gains for the other attributes at this level are shown in table 2.

Attrib	Value1	Value2	Split1	Split2	Entropy1	Entropy2	Gain
Sky	Sunny	Rainy	[3+,1-]	[0+,1-]	0.811278	-0.000000	0.321928
AirTemp	Warm	Cold	[3+,1-]	[0+,1-]	0.811278	-0.000000	0.321928
Humidity	Normal	High	[1+,1-]	[2+,1-]	1.000000	0.918296	0.019973
Wind	Strong	Weak	[3+,1-]	[0+,1-]	0.811278	-0.000000	0.321928
Water	Warm	Cold	[2+,2-]	[1+,0-]	1.000000	-0.000000	0.170951
Forecast	Same	Change	[2+,1-]	[1+,1-]	0.918296	1.000000	0.019973

Table 1: Gains for possible attributes at the root level

Attrib	Value1	Value2	Split1	Split2	Entropy1	Entropy2	Gain
AirTemp	Warm	Cold	[3+,1-]	[0+,0-]	0.811278	0.000000	0.000000
Humidity	Normal	High	[1+,1-]	[2+,0-]	1.000000	-0.000000	0.311278
Wind	Strong	Weak	[3+,0-]	[0+,1-]	0.000000	0.000000	0.811278
Water	Warm	Cool	[2+,1-]	[1+,0-]	0.918296	-0.000000	0.122556
Forecast	Same	Change	[2+,1-]	[1+,0-]	0.918296	-0.000000	0.122556

Table 2: Gains for possible attributes on the *Sunny* branch of *Sky*.

From the table is clear that the split should be done on attribute *Wind* and label the *Strong* branch **YES** and the *Weak* branch **NO**.

The final tree is shown in Figure 3.

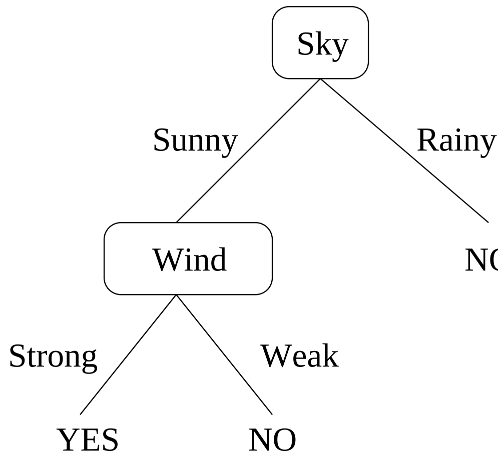

Figure 3: Final decision tree for Problem 3.4c)

Grading: this part was 10 points. I took off 5pt if you just produced the final tree with the gain computed only for the nodes in the tree. If you had mistakes in computing the gains I took 1-7 points off depending on the number of mistakes.

d)

A tree is more specific than an other tree if it classifies all the examples the first tree classifies as negative plus some more.

A tree is more general than an other tree if it classifies all the examples the first tree classifies as positive plus some more.

Thus before seeing any example in S there is only one tree consisting of the leaf **NO** and in G there is only one tree consisting of the leaf **YES**. When the *Candidate-Elimination* algorithm is applied for the first two examples in Table 2.1 in the textbook, since both are positive the set G remains the same (since both examples agree with the tree **YES**). After seeing the first example S contains only the tree in Figure 4a and after seeing the second example S contains only the tree in Figure 4b.

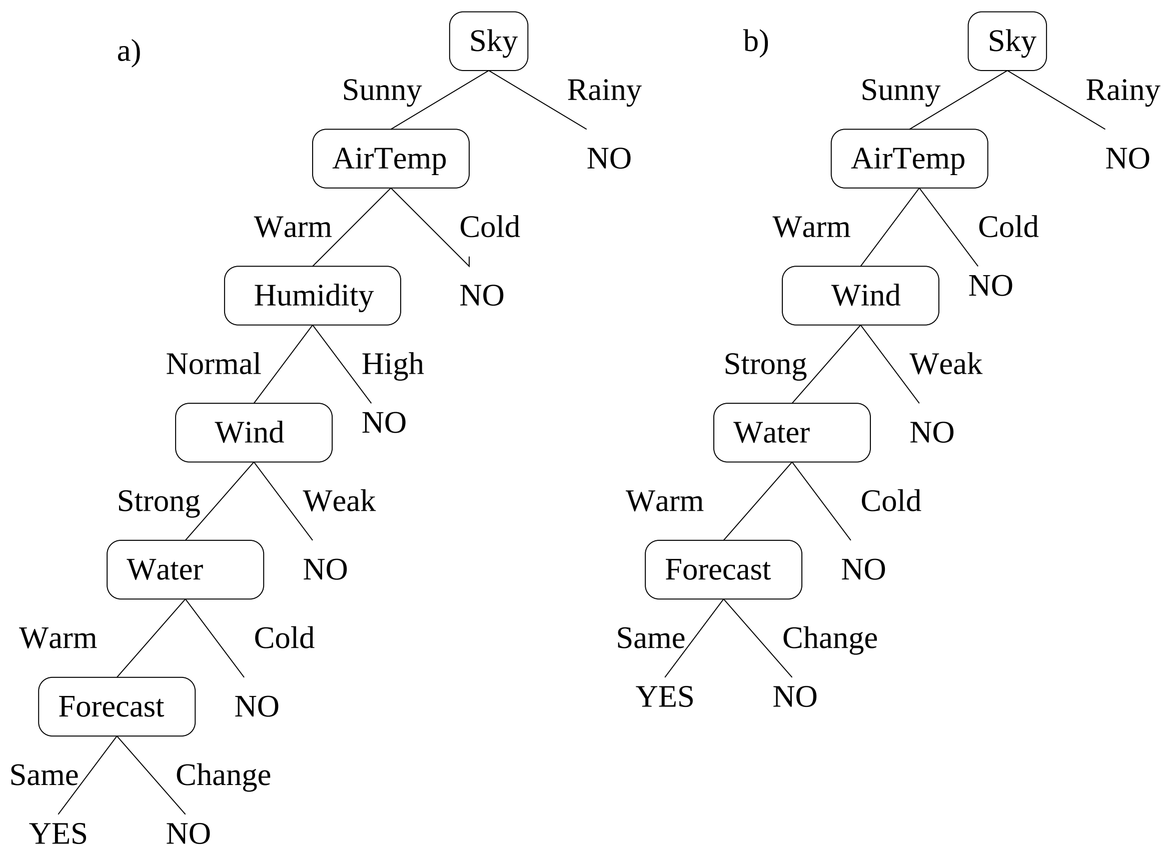

Figure 4: a) S_1, b) S_2

From the tree in S_1 the node *Humidity* was eliminated (thus generalizing the tree) in order to make the tree agree with the first two examples. Note that there are many trees that are the *most specific* tree after the first example is seen but they are syntacticly identical (permuting nodes in the tree will preserve the meaning in this example).

Some of the difficulties in applying the *Candidate-Elimination* to a decision tree hypothesis are linked to the specialization and generalization of a given tree to include a new example (which can become a difficult operation on complicated trees), detection of syntactic equivalence (can probably be dealt with by using a canonical representation by choosing a tree from the class of syntacticly equivalent trees in a systematic fashion). The first problem comes from the fact that is hard to decide in what relation are two given trees (if any).

Grading: this part was 9 points, 3 for S_1 and G_1, 3 for S_2 and G_2 and 3 for mentioning at least a plausible difficulty. I took 2 points off if G's were wrong (some of you suggested that G is all the trees which is not the case since some of the trees are more specific and they should be eliminated from G, in the end G being only label YES). I also took 1 point off if you didn't had all the branches in the tree (only a path from root to the YES label). I took 1-2 points off is the difficulty you mentioned was not a difficulty or the statement was not clear.

Problem 3

Grading: In order to get full credit (10pts), your answer had to address the advantages and disadvantages of the two solutions w.r.t. their relative complexity, accuracy, and interpretability.

Some possible items to include in the discussion

The single decision tree is a more compact, concise representation of the concept being learned. Using multiple trees requires encoding redundant information and although each single-class tree may be smaller than the single multi-class tree (although not all that much smaller in practice), the combined set of trees that must be traversed in the multiple-tree solution will be much larger. This means a corresponding loss in efficiency for the multiple-tree solution, both for building the trees and using the trees. The multiple-tree solution also requires extra steps in preprocessing the data.

A single tree would tend to be easier to understand; a multiple-tree solution distributes the concept description across a set of trees.

In the multiple-tree solution, it is possible for the learned trees to give conflicting class information for any particular test instance. A decision procedure would be needed to make a final class decision for domains where a single prediction is required. This complication can be viewed as an advantage in some cases, however — for problems where more than one classification is allowed or in domains where you'd prefer the learning algorithm to indicate uncertainty in class prediction whereever such uncertainty existed.

In terms of accuracy, fewer training examples per class are available when training the single decision tree when compared to the multiple-tree solution. This might mean that the single-tree solution is less accurate. In practice, however, this is not the case [Dietterich and Bakiri, JAIR, 1995].