## CSE 5693 Machine Learning

## HW2 Decision Tree Learning

Josias Moukpe

Written Assignment

**(a) 2.4**

**Instance space consist of integer points in x, y plane and H is the set of hypotheses consisting of rectangles. The hypotheses are of the form a ≤ x ≤ b, c ≤ y ≤ d where a, b, c, and d are integers.**

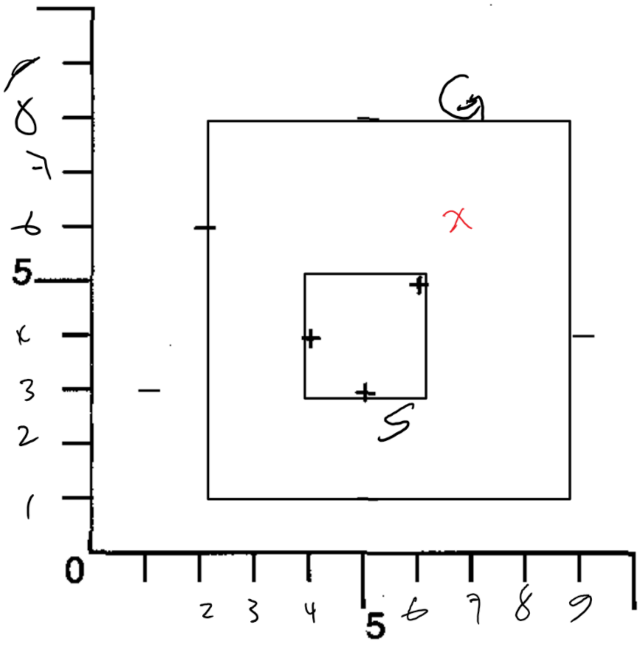


Figure 1. Drawing of S and G boundaries

**a. What is the S boundary of the version space?**

S: 4 ≤ x ≤ 6, 3 ≤ y ≤ 5

See figure 1 for drawing of S

**b. What is the G boundary of the version space?**

G: 2 ≤ x ≤ 9, 1 ≤ y ≤ 8

See figure 1. for drawing of G

**c. Suggest x, y instance**

P: x = 7, y = 6 is guaranteed to reduce the size of the version space

if P is positive (+), S: 4 ≤ x ≤ 7, 3 ≤ y ≤ 6

if P is negative (-), G: 2 ≤ x ≤ 7, 1 ≤ y ≤ 6

Q: x = 2, y = 1 is not guaranteed to reduce the size of the version space

Since if Q is negative (-), G doesn't change

**d. Teacher**

A minimum, you need 4 training examples, since the rectangle can be described by 2 pairs of points, one pair of positive points and another pair of negative points to set the S and G limits. For example, positive pair {(3,2), (5, 9)} and negative pair {(2,1), (6, 10)} is enough for candidate eliminate to learn target: 3 ≤ x ≤ 5 and 2 ≤ y ≤ 9

**(b) 2.7**

**Consider a concept learning problem in which each instance is a real number, and in which each hypothesis is an interval over the reals. More precisely, each hypothesis in the hypothesis space H is of the form a < x < b, where a and b are any real constants, and x refers to the instance. For example, the hypothesis 4.5 < x < 6.1 classifies instances between 4.5 and 6.1 as positive, and others as negative. Explain informally why there cannot be a maximally specific consistent hypothesis for any set of positive training examples. Suggest a slight modification to the hypothesis representation so that there will be.**

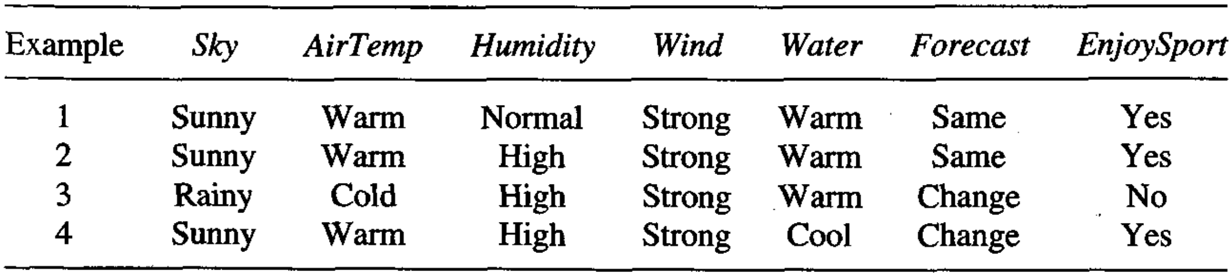
Instances are real numbers and a hypothesis is of the form a < x < b.

Between 2 real numbers, there is an infinity of real numbers so there cannot be any maximally specific hypothesis for positives. A modification of the hypothesis representation would be to use the form a ≤ x ≤ b so the maximally specific hypothesis for positives would be when a = b so a ≤ x ≤ a or b ≤ x ≤ b.

**(c) 3.4**

**ID3 searches for just one consistent hypothesis, whereas the CANDIDATE-ELIMINATION algorithm finds all consistent hypotheses. Consider the correspondence between these two learning algorithms.**

a. **Show the decision tree that would be learned by ID3 assuming it is given the four training examples for the Enjoy Sport?**



| Example | *Sky* | *AirTemp* | *Humidity* | *Wind* | *Water* | *Forecast* | *EnjoySport* |
|---|---|---|---|---|---|---|---|
| 1 | Sunny | Warm | Normal | Strong | Warm | Same | Yes |
| 2 | Sunny | Warm | High | Strong | Warm | Same | Yes |
| 3 | Rainy | Cold | High | Strong | Warm | Change | No |
| 4 | Sunny | Warm | High | Strong | Cool | Change | Yes |

Entropy(S) = -3/4 log2 (3/4) – 1/4 log2 (1/4) = .811

Gain(S, Sky) = .811

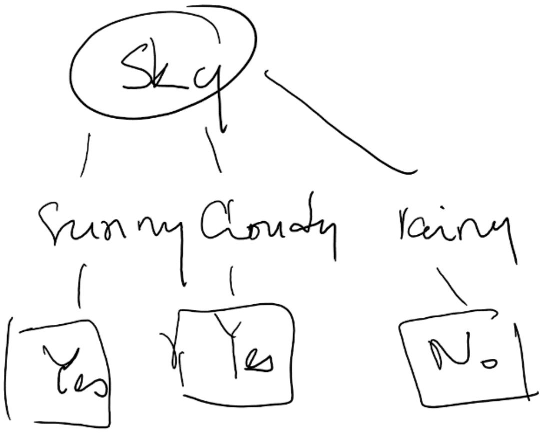
Gain(S, Airtemp) = .811

Gain(S, Humidity) = .123

Gain(S, Wind) = .0

Gain(S, Water) = .123

Gain(S, Forecast) = .123



b. **What is the relationship between the learned decision tree and the version space (shown in Figure 2.3 of Chapter 2) that is learned from these same examples? Is the learned tree equivalent to one of the members of the version space?**

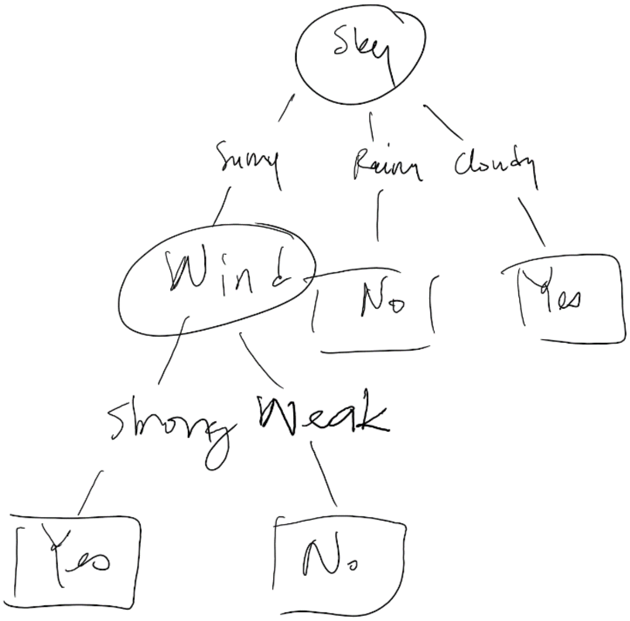
The decision tree learned represent all the elements of the version space shown in Figure 2.3 of Chapter 2 and more. The sunny branch alone represents all the enumerated elements of the version space where sky = sunny.

c. **Add the following training example, and compute the new decision tree. This time, show the value of the information gain for each candidate attribute at each step in growing the tree.**

Entropy(S) = -3/5log2(3/5) – 2/5log2(2/5) = .97

Iteration 1

Gain(S, Sky) = .322
Gain(S, Airtemp) = .322
Gain(S, Humidity) = .02
Gain(S, Wind) = .322
Gain(S, Water) = .17
Gain(S, Forecast) = .17



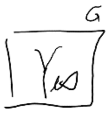
Iteration 2

Gain(S, Airtemp) = .0
Gain(S, Humidity) = .311
Gain(S, Wind) = .811
Gain(S, Water) = .123
Gain(S, Forecast) = .123

d. **Suppose we wish to design a learner that (like ID3) searches a space of decision tree hypotheses and (like CANDIDATE-ELIMINATION) finds all hypotheses consistent with the data. In short, we wish to apply the CANDIDATE-ELIMINATION algorithm to searching the space of decision tree hypotheses. Show the S and G sets that result from the first training example from Table 2.1. Note S must contain the most specific decision trees consistent with the data, whereas G must contain the most general. Show how the S and G sets are refined by these constraining example (you may omit syntactically distinct trees that describe the same concept). What difficulties do you foresee in applying CANDIDATE-ELIMINATION to a decision tree hypothesis space?**

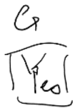
After 1 training example

The most general tree G would be



And the most specific S would be

After 2 training examples
The most general tree G is still



But the most specific S is now

The difficulty arises in the fact that CANDIDATE-ELIMINATION is an unbiased learner, and Decision Tree learning hypothesis space is an unbiased hypothesis space. As a result, we lose the ability to be robust to noise and to generalize to new unseen data.

**(d) Play Tennis**

**Consider two attributes Outlook (sunny, rainy, cloudy) and Humidity (high) and outcome PlayTennis (yes, no) for the instance space (X).**

**a. Consider an unbiased hypothesis space (H1), enumerate all possible hypotheses (h1, h2, …) in terms of subsets of instances. What is the number of possible unique hypotheses in H1?**

Since we have 2 attributes Outlook with 3 values, and Humidity with 1 value, we have

|X| = 3 x 1 = 3

The number of possible hypothesis in H1 is |H1| = 2^|X| = 2^(3) = 8

They are:

h1: {},

h2: {<sunny, high>}

h3: {<cloudy, high>}

h4: {<rainy, high>}

h5: {<rainy, high>, <cloudy, high>}

h6: {<sunny, high>, <cloudy, high>}

h7: {<sunny, high>, <rainy, high>}

h8: {<sunny, high>, <rainy, high>, <cloudy, high>}

**b. For each hypothesis in H1, represent it as a boolean expression. What is the number of unique hypotheses semantically?**

h1: null

h2: sunny ^ high

h3: cloudy ^ high

h4: rainy ^ high

h5: rainy ^ high v cloudy ^ high

h6: sunny ^ high v cloudy ^ high

h7: sunny ^ high v rainy ^ high

h8: sunny ^ high v rainy ^ high v cloudy ^ high

We have in total 8 semantically unique hypotheses in H1.

**c. Consider a biased hypothesis space (H2) where each attribute can only have a value, ?, or null. What is the number of unique hypotheses semantically in the biased hypothesis space (H2)?**

The number of semantically unique hypotheses in H2 is 5 ( = 1 + (3 + 1)(1))

h1: <null, null> (equivalent to anything with null)

h2: <sunny, high> (or <sunny, ?>)

h3: <cloudy, high> (or <cloudy, ?>)

h4: <rainy, high> (or <rainy, ?>)

h8: < ? , ? >

**d. Identify hypotheses in the unbiased hypothesis space (H1) that are not in the biased hypothesis space (H2).**

The hypotheses present in H1 and absent in H2 are the following:

h5: rainy ^ high v cloudy ^ high

h6: sunny ^ high v cloudy ^ high

h7: sunny ^ high v rainy ^ high

Which represent disjunctions not present in the bias hypothesis space H2.

**(e) Discuss and compare accuracy of no pruning versus rule post-pruning in testIris and testIrisNoisy. Include plots for the comparisons.**

Here I am comparing and discussing my result on the effect of noise level on tree accuracy and how pruning helps mitigate those effects.

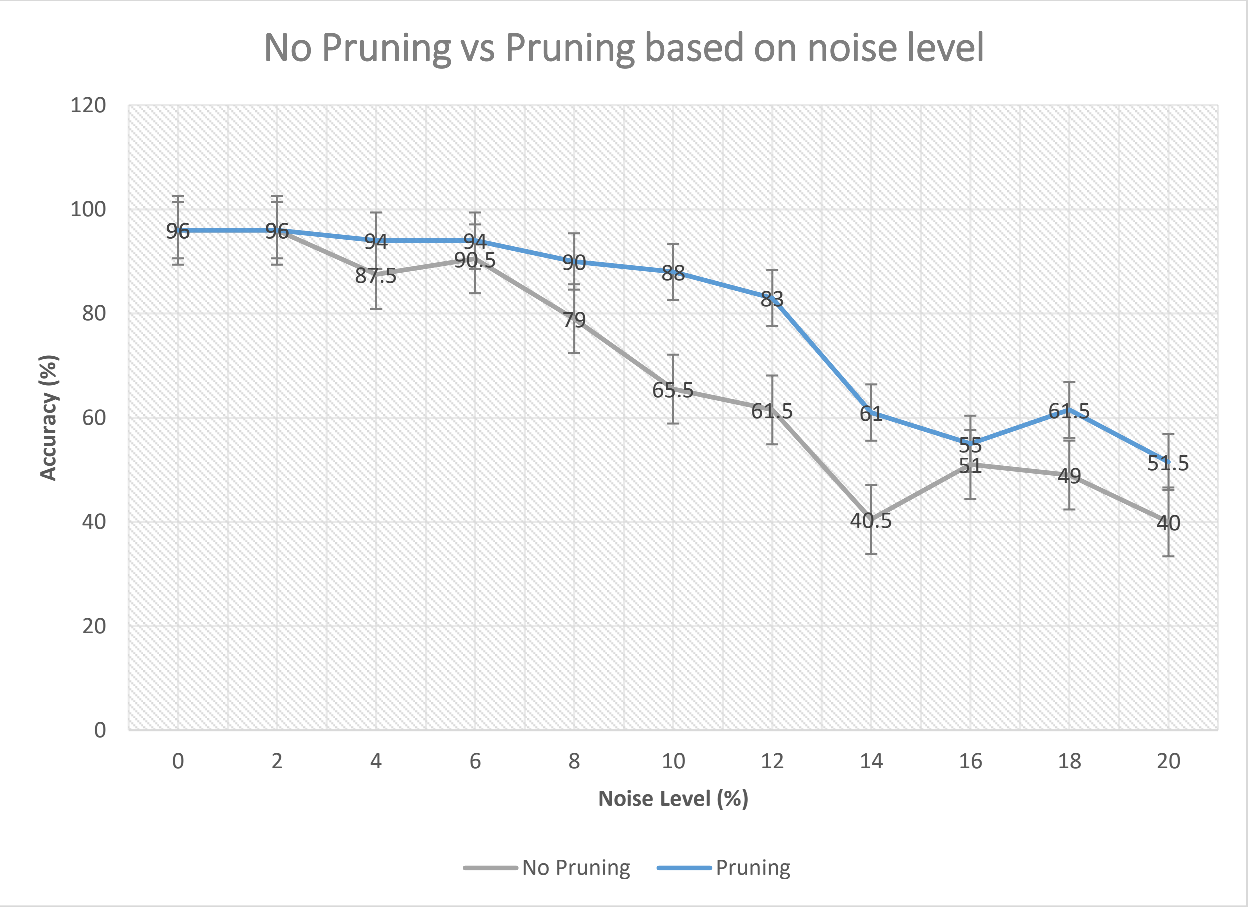


Figure 2. Comparison of pruning vs no pruning based on noise level

Here we are working with the Iris dataset. The noise is applied at random to the training data prior to the experiment and **30%** (found to be the sweet spot during experimentation) of the training data is reserved for validation and rule post-pruning. The following results are the averages of 4 independent trials to better capture the trend.

The Figure 2. above shows how, when the noise level increases from 0% to 20% (on the X-Axis), the accuracy on the test data (on the Y-Axis) of the base tree, with no pruning, quickly drops. By the time the noise level reaches 10%, the base tree has already lost about 30% of its accuracy on the test data. At 20% noise level, the base tree only has 40% of accuracy, meaning we can do better at prediction by flipping an unbiased coin than using the base tree. Meanwhile, the pruned tree still retains its resilience, only losing 30% in accuracy at 14% noise, and still doing slightly better than the coin flip at the extreme 20% noise level.

From the results, we can conclude that rule-post pruning is an effective way to deal with noisy data when learning a decision tree.

Appendix 1

Table 1. Data table of the experimental results.

| Noise Level (%) | No Pruning Accuracy (%) | | | | | Pruning Accuracy (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Trial 1 | Trial 2 | Trial 3 | Trial 4 | **Average** | Trial 1 | Trial 2 | Trial 3 | Trial 4 | **Average** |
| 0 | 96 | 96 | 96 | 96 | **96** | 96 | 96 | 96 | 96 | **96** |
| 2 | 96 | 96 | 96 | 96 | **96** | 96 | 96 | 96 | 96 | **96** |
| 4 | 92 | 94 | 88 | 76 | **87.5** | 96 | 96 | 88 | 96 | **94** |
| 6 | 84 | 92 | 94 | 92 | **90.5** | 90 | 96 | 94 | 96 | **94** |
| 8 | 74 | 70 | 86 | 86 | **79** | 90 | 78 | 96 | 96 | **90** |
| 10 | 60 | 64 | 80 | 58 | **65.5** | 90 | 76 | 90 | 96 | **88** |
| 12 | 42 | 66 | 70 | 68 | **61.5** | 54 | 90 | 92 | 96 | **83** |
| 14 | 22 | 54 | 40 | 46 | **40.5** | 30 | 64 | 64 | 86 | **61** |
| 16 | 36 | 52 | 54 | 62 | **51** | 48 | 46 | 76 | 50 | **55** |
| 18 | 46 | 30 | 56 | 64 | **49** | 50 | 60 | 88 | 48 | **61.5** |
| 20 | 50 | 32 | 40 | 38 | **40** | 54 | 62 | 32 | 58 | **51.5** |