Homework 3 Sample Solutions, 15-681 Machine Learning

Chapter 3, Exercise 1

(a)

$$m \geq \frac{1}{\epsilon}(ln|H| + ln\frac{1}{\delta})$$

$$m \geq \frac{1}{0.15}(ln((100 * 101)/2)^2 + ln\frac{1}{0.05})$$

$$m \geq 133.7$$

(b)

For the 1-D case (i.e. where rectangles = line segments), in the interval $[0, 99]$ there are 100 concepts covering only a single instance, and $\frac{100(100-1)}{2}$ concepts covering more than a single instance, yielding a total of 5050 concepts.

In d dimensions, there exists one hypothesis for each choice of a 1-D hypothesis in each dimension, or 5050^d concepts. So the number of examples necessary for a consistent learner to output a hypothesis with error at most ϵ with probability $1 - \delta$ is

$$m \geq \frac{1}{\epsilon}(ln5050^d + ln\frac{1}{\delta})$$

or

$$m \geq \frac{1}{\epsilon}(8.53d + ln\frac{1}{\delta})$$

which is clearly polynomial in $1/\delta$, $1/\epsilon$, and d.

(c)

Algorithm for learner L, *Find-Smallest-Consistent-Rectangle*:

- Hypotheses are of the form $(a \leq x \leq b)AND(c \leq y \leq d)$.
- Initially, let a, b, c, and d be set to values such that the hypothesis covers *no* instances.
- For the first postive example, (x, y), seen, set a and b to x and c and d to y.
- Thereafter, lower a and c and raise b and d as little as necessary to cover each positive example seen. That is, for each successive positive example,

$$a = min(a, x)$$

$$b = max(b, x)$$
$$c = min(c, y)$$
$$d = max(d, y)$$

- Negative examples are ignored.

Claim: C is PAC-learnable by L

Proof:

- L is a consistent learner. This can be seen by noticing that if L outputs an inconsistent hypothesis, it must include a negative example because the hypothesis is specifically constructed to contain all positive examples. Furthermore, there then could not exist any other hypothesis consistent with the examples because L chooses the smallest rectangle possible to cover the postive examples. So, because the failure of L to output a consistent hypothesis implies that there exists no such hypothesis, the existence of a consistent hypothesis implies that L will output one.
- Based on part B above, the number of examples necessary for a consistent learner such as L to output a hypothesis H in C of error no more than ϵ with probability $1 - \delta$ is polynomial in both $1/\epsilon$ and $1/\delta$.
- Because L only needs constant time per example, the time necessary for it to output hypothesis H is also polynomial in the PAC parameters.
- Therefore, C is PAC-learnable by L. ■

Chapter 4, Exercise 3

(a)

Depending on how ties are broken between attributes of equivalent information gain, one possible learned tree is:

```
          +-----+
          | Sky |
          +-----+
            / \
    Sunny /     \ Rainy
         /       \
       Yes        No
```

(b)

The learned decision tree is on the most-general boundry of the version space. Specifically, it corresponds to the hypothesis `<Sunny, ?, ?, ?, ?, ?>`.

(c) First stage:

$$Entropy(S) = 0.971$$
$$Entropy([3+, 1-]) = 0.811$$
$$Entropy([2+, 1-]) = 0.918$$
$$Entropy([2+, 2-]) = Entropy([1+, 1-]) = 1.0$$

$$Gain(S, Sky) = 0.971 - (4/5)0.811 - (1/5)0.00 = 0.321$$
$$Gain(S, AirTemp) = 0.971 - (4/5)0.811 - (1/5)0.00 = 0.321$$
$$Gain(S, Humidity) = 0.971 - (3/5)0.918 - (2/5)1.00 = 0.020$$
$$Gain(S, Wind) = 0.971 - (4/5)0.811 - (1/5)0.00 = 0.321$$
$$Gain(S, Water) = 0.971 - (4/5)1.0 - (1/5)0.00 = 0.171$$
$$Gain(S, Forecast) = 0.971 - (3/5)0.918 - (2/5)1.00 = 0.020$$

If ID3 ends up picking Sky again, the intermediate tree looks like:

```
          +-----+
          | Sky |
          +-----+
            / \
     Sunny /   \ Rainy
          /     \
        ???     No
```

Second stage:

$$S' = S - rainyexample$$
$$Entropy(S') = 0.811$$

$$Gain(S', AirTemp) = 0.811 - (4/4)0.811 = 0.0$$
$$Gain(S', Humidity) = 0.811 - (2/4)1.0 - (2/4)0.0 = 0.311$$
$$Gain(S', Wind) = 0.811 - (3/4)1.0 - (1/4)1.0 = 0.811$$
$$Gain(S', Water) = 0.811 - (3/4)0.918 - (1/4)1.0 = 0.127$$
$$Gain(S', Forecast) = 0.811 - (3/4)0.918 - (1/4)1.0 = 0.127$$

and the resulting tree looks like:

```
          +------+
          | Sky  |
          +------+
            /  \
     Sunny /    \ Rainy
          /      \
      +-------+   No
      | Wind  |
      +-------+
        /  \
Strong /    \ Weak
      /      \
     Yes      No
```

(d)

After example 1:

```
 G = Yes

 S =                                    +-----+
                                        | Sky |
                                        +-----+
                                          / \
                                   Sunny /   \ Rainy
                                        /     \
                              +----------+    No
                              | Air-Temp |
                              +----------+
                                  / \
                            Warm /   \ Cold
                                /     \
                          +------+    No
                          | Wind |
                          +------+
                             / \
                     Strong /   \ Weak
                           /     \
                    +-------+    No
                    | Water |
                    +-------+
                       / \
                 Warm /   \ Cool
                     /     \
           +----------+     No
           | Forecast |
           +----------+
               / \
         Same /   \ Change
             /     \
   +----------+    No
   | Humidity |
   +----------+
       / \
Norm  /   \ High
     /     \
    Yes     No

 and all other trees representing the same concept.
```

After example 2:

```
G = Yes

S =                                    +-----+
                                       | Sky |
                                       +-----+
                                         / \
                                  Sunny /   \ Rainy
                                       /     \
                             +----------+    No
                             | Air-Temp |
                             +----------+
                                 / \
                           Warm /   \ Cold
                               /     \
                         +------+    No
                         | Wind |
                         +------+
                            / \
                    Strong /   \ Weak
                          /     \
                   +-------+    No
                   | Water |
                   +-------+
                      / \
                Warm /   \ Cool
                    /     \
          +----------+     No
          | Forecast |
          +----------+
              / \
        Same /   \ Change
            /     \
          Yes     No

and all other trees representing the same concept.
```

There are a lot of things that one could say about the difficulties in applying Candidate Elimination to a decision tree hypothesis space. However, probably the single most important thing to note is that because of the fact that decision trees represent a complete hypothesis space and because Candidate Elimination has no search bias, the algorithm will only end up doing rote memorization, and will lack the ability to generalize to unseen examples.