CS151 - Written Problem 7
Solutions

1. Think about 18.1 and 18.2

2. Draw a decision tree for deciding whether or not to move forward at a road intersection. Use variables such as `FrontOfQueue`, `CarAheadMoving`, `IntersectionBlocked`, `CrossTraffic`, `Pedestrians`, `TurningDirection`, `Cyclist`.

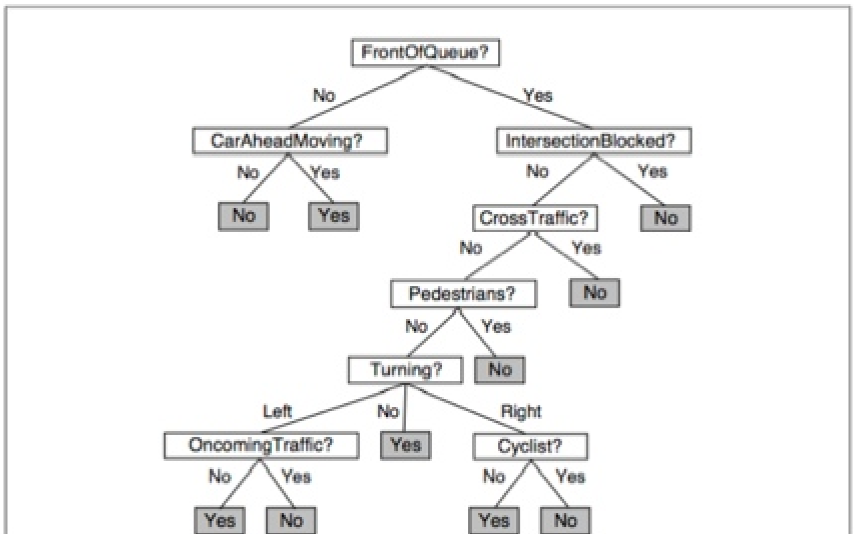

Figure S18.1 A decision tree for deciding whether to move forward at a traffic intersection, given a green light.

3. 18.25

   (a) For $d = 2$ the linear separator is just a line. With $N = 3$ points, we can have one of two situations:

- If all of the points have the same label, then creating a linear separator that separates positive and negative is trival: any line that does not go in between the three points will separate them.
- The other case is when two points have the same label and the third a different label. Draw a line through the two points with the same label. The third point will be on one side of this line (since the points are not colinear). If we move this line just slightly in the direction of this third point (i.e. ϵ in the direction of the vector that is tangential to the line) then this move line will have the two points of the same class on one side and the third point on the other side.

(b) The so called "XOR" set of examples cannot be separated by a line, for example, take:

$[(0,1), positive], [(0,-1), positive], [(1,0), negative], [(-1,0), negative]$

(c) Similar to part b above, we can break it into different situations:

- If they are all of the same class, we can put a plane anywhere not in between the points.
- If three of the four are in the same class. Draw a plan through the three points (this will always be possible since it is a 3-dimensional space). Since the points are not coplanar, the fourth point must be on one side of the plane and we can move the plane a small distance in the direction of that point (again ϵ in the direction of the vector that is tangential to the plane).
- If there are two in each class, pick two points in the same class. We can draw a line between these two points. We can define the plane that is tangential to this line (in this case it will be a plane in 2 dimensions). Project all of these points on to this plane. This will now result in the two points that were along the dimension of v on top of eachother and the other two points that were of a different class in other locations. This now defines a recursive case where we now want to find a line that separates the two points of the other class and the one point (representing the two collapsed points). If we then take this line and turn it into a plane by extending it out in the direction of v we get a hyperplane that separates the 4 points.

- Again, we can do an XOR type set of examples. Take the examples from part **b** and set their third dimension to 0. These cannot be separated by a plane, but are not coplanar. Add the fifth point to either class anywhere that is not along this plane. The points are not coplanar, but still cannot be separated.

(d) In parts **c** and **d** the descriptions are recursive on lower dimensions. To show this for any d dimensions, the argument follows a similar inductive approach . To show that n points can be separated in $d = n - 1$ dimensions, we take two points and then project the points to get a problem of $n-1$ points in $d = n-2$ dimensions. To show that $n+1$ points cannot always be separated in $d = n-1$ dimensions, we can build up a solution from the basic XOR example in one plane and then add points to maintain the non-coplanar requirement.

4. 18.18, but you don't have to actually calculate the values (though feel free to if you want to). *Hint 1:* draw out the tree of possibilities. For example, with $K = 1$ there are just two possibilities, right or wrong. What is the probability of this happening. With $K = 2$ there are now four possibilities (all combinations of the two classifiers getting it right and wrong). *Hint 2:* If you follow this logic, a pattern should start to emerge. The "binomial coefficients" (i.e. "n choose k" may be useful).

   The key to this is to draw out a binary tree with the possible options. For K classifiers, the tree will have depth K and 2^K leaves representing all possible choices the classifiers could make. Think of it as follows: the first classifier chooses and is right or wrong. Then, the next classifier chooses and is again right or wrong. We now have four different possible combinations of right and wrong. If we add another classifier, we now have 8 different combinations. The key is noticing that some of these combinations are the same (e.g. there are multiple ways that two of the three different classifiers can get it right, but these all of the same probability of happening).

   For $K = 1$ we have:

   wrong: with probability ϵ
   right: with probability ϵ

   For $K = 2$ our set of possibilities increases:

both wrong: with probability ϵ^2
both right: with probability $(1-\epsilon)^2$
one right, one wrong: with probability $2\epsilon(1-\epsilon)$ (two ways this can happen)

For $K = 3$ our set of possibilities is:

all wrong: with probability ϵ^3
2 of 3 wrong: with probability $3\epsilon^2(1-\epsilon)$ 1 of 3 wrong: with probability $3\epsilon(1-\epsilon)^2)$ all right: with probability $(1-\epsilon)^3$

Notice there are three different ways that we can get two wrong with three classifiers.

If we continue this out, we will notice that these numbers are the binomial coefficients (aka Pascal's triangle).

We can bound the error by assuming that we get it wrong every time there is a tie (we could also assume we get it right half of the time). The general equation then is a sum over all the ways we can get it wrong:

$$error(K, \epsilon) = \sum_{i=0}^{k+1/2} \binom{K}{i} \epsilon^K (1-\epsilon)^{K-i}$$

To calculate this at each stage, we multiplied the probabilities because we assumed the classifiers made mistakes independently. If they did not, then we could easily get an error larger than epsilon. Take $K = 2$ and let the first and second classifiers miss disjoint sets of examples. The error would now be 2ϵ.