# Sample Midterm 1
# Sample Answers

CS142 Spring 2018

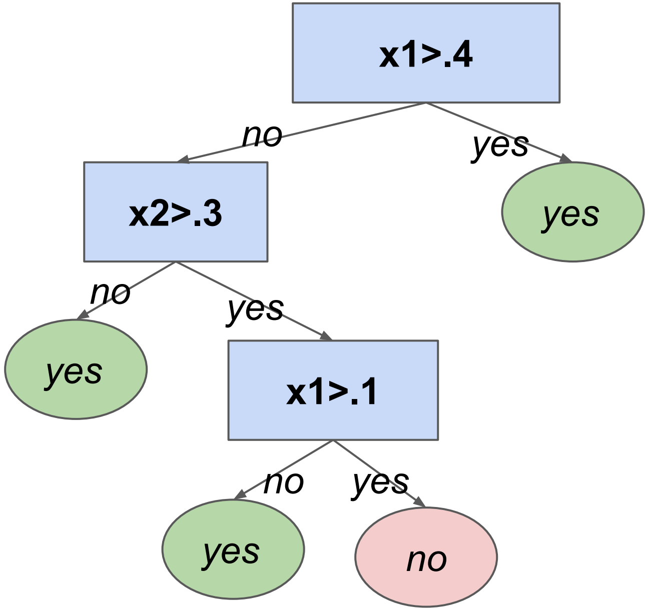
# Problem 1 (algorithms)

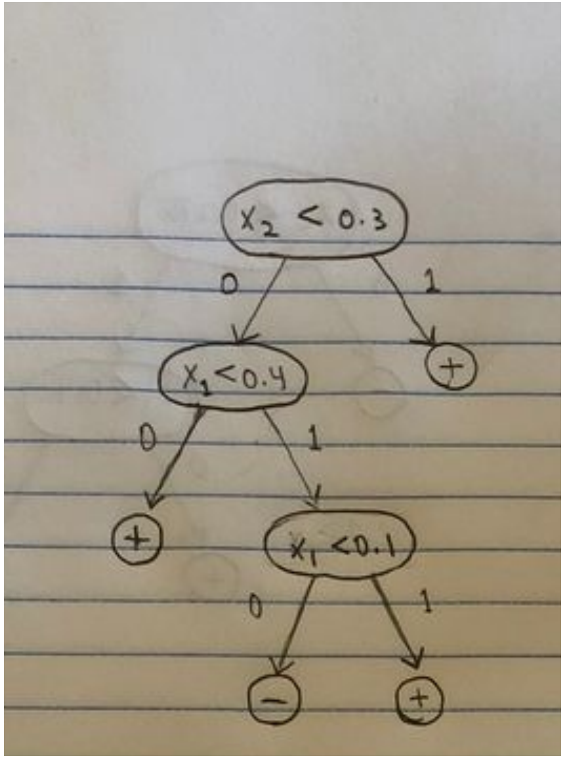
a. The performance is decreased by a very small amount, as duplicating an attribute gives slightly more weight to that attribute in Naive Bayes. Assuming all the attributes are distributed identically, this will decrease the accuracy of the model as the duplication will boost the error from the duplicated attribute.

b. Performance should improve a bit, but not be perfect. In particular, the new attribute will help move examples closer to other points with the same label, but it is just one component of the distance.

# Problem 2 (classifiers)

There are multiple possible answers. Here are two.





# Problem 3 (representation)

a. There are several possible answers. Here are two:

$b = ½$, $b^{10} = ½$, $b^{11} = 1$, $b^{20} = ½$, $b^{21} = 1$

$b = 0.4$, $b^{10} = 0.4$, $b^{11} = 0.6$, $b^{20} = 0.4$, $b^{21} = 0.9$

b. MLE results in: $b = 0.1$, $b^{10} = 0.44$, $b^{11} = 0.9$, $b^{20} = 0.44$, $b^{21} = 0.9$

These parameters do not correctly classify 11.

# Problem 4 (VC dimension)

## The VC dimension is $2^d$:

A decision tree of depth d can uniquely define $2^d$ intervals. Each decision takes a single interval as input, and generates two intervals after the decision is made. A full decision tree of depth d will have $1 + 2 + \ldots + 2^{d-1} = 2^d - 1$ decisions, resulting in $2^d$ uniquely defined intervals (each corresponding to a leaf in the tree). Each leaf may be assigned either class. If a set of $2^d$ points are selected such that one is in each of the $2^d$ intervals defined by the decision tree (sans labels), then this said of points may be shattered by the set of classifiers generated when considering all possibly labellings of the leaves.

Consider a set of $2^d + 1$ points. As each decision tree has at most $2^d$ uniquely defined intervals, each set of $2^d + 1$ points will have a pair of consecutive points within the same interval for any given tree. No decision tree will be able to successfully classify the set of points if they have alternating labels, as there is guaranteed to be a pair of consecutive points (which will have different labels) in the same interval. As this labelling can never be properly classified by a decision tree, the class of decision trees cannot shatter any set of $2^d + 1$ points.

The above proofs are sufficient to show that the VC dimension of decision trees of depth d is $2^d$.

# Problem 5 (loss)

a. With probability ½, the coin says heads, she reports p, and has loss 1-p. With probability ½, the coin says tails, she reports p twice, and has a loss of 2p. Total expected loss is: 1/2 (1-p) + 1/2 2p = 1/2 - 1/2 p + p = 1/2 + 1/2 p. Minimized for p=0 (always say tails).

b. With probability ½, the coin says heads, she reports p, and has loss (1-p)^2. With probability ½, the coin says tails, she reports p twice, and has a loss of 2p^2. Total expected loss is: $1/2\ (1\text{-}p)^2 + 1/2\ 2p^2 = 1/2\ (1\text{-}\ 2p + p^2) + p^2 = 1/2 - p + 1/2p^2 + p^2 = 1/2 - p + 3/2\ p^2$. Derivative: -1 + 3p, zero when p = ⅓, so loss minimized by saying p=⅓.

# Problem 6 (optimizers)

a. Let a be the smallest positive example, b be the largest positive example, c be the smallest negative example, and d be the largest negative example. We can compute these values in a single pass through the data. If b < c (positives all on the left) or d < a (positives all on the right) or (c<a and b < d) (positives on the inside), return a,b,+1. Otherwise (negatives on the inside), return c,d,-1. Complexity is O(m) to find the maxes and mins.

b. Sort the list of samples based on the input value. Next, try each possible pair of sample inputs, $x_i$ and $x_j$ s.t. i <= j, as a and b and calculate the resulting error of this hypothesis with both label 1 and label -1 on all of the inputs. Define h based on the $x_i$, $x_j$, and labels that minimized the error. We get a total complexity of $O(m^3)$, since we are iterating over every pair of samples and then iterating over each of the m samples with each label. Can probably improve running time to $O(m^2)$ by looping through the points and computing the errors at the same time.