CS 189: Introduction to Machine Learning - Discussion 9

1. Decision Trees

   Recall that training a decision tree requires looking at every feature to find the best split, where the best split greedily maximizes the information gain. The information gain is defined as

   $$H - \left[\frac{n_1 H_1 + n_2 H_2}{n_1 + n_2}\right]$$

   where H is the entropy at the current node, H_1 is the entropy at the "left" split, and H_2 is the entropy at the "right" split. n_1 and n_2 are the number of data points at the "left" and "right" splits.

   (a) What are good values to choose to test the splits?

   (b) What is the running time for the naive approach to finding the best split (just finding the split, not training the entire tree)?

   (c) What is a smarter way to search for the best split, and what is the running time of this?

   Now consider decision trees for regression. We can no longer use our notion of entropy as a measure of how well our data is split, since our values for our labels are continuous. We want the data at the leaves to be spread as little as possible.

   (a) What is a good measure to use to determine how well our data is spread? (Hint: think back to all of our real valued problems. What error measure did we use?)

   (b) Write down the equation we want to maximize when searching over splits for regression trees.

2. Maximum Entropy Distribution

   Suppose we have a discrete random variable that has a Categorical distribution described by the parameters $p_1, p_2, \ldots, p_d$. Recall that the definition of entropy of a discrete random variable is

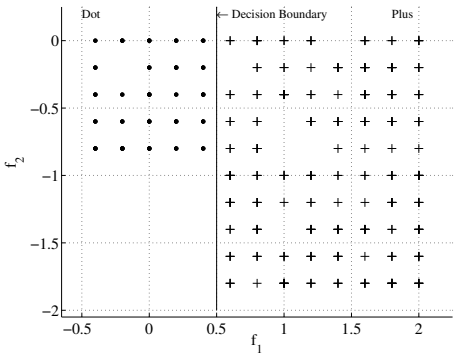
$$H(X) = E[-\log p(X)] = -\sum_{i=1}^{d} p_i \log p_i$$

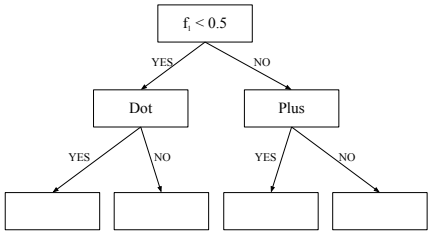
   Find the distribution (values of the p_i) that maximizes entropy. (Hint: remember that $\sum_{i=1}^{d} p_i = 1$. Don't forget to include that in the optimization as a constraint!)

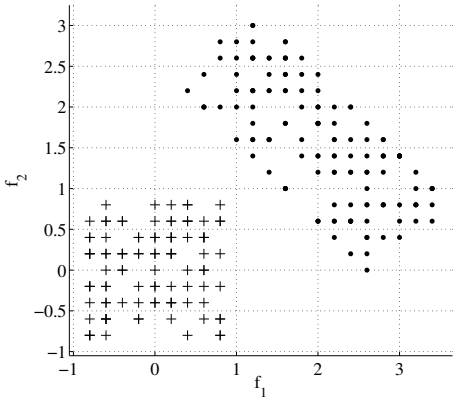
3. You are given points from 2 classes, shown as +'s and ·'s. For each of the following sets of points,

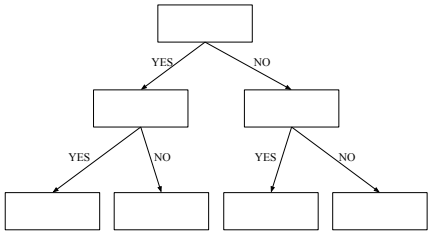
    1. Draw the decision tree of depth at most 2 that can separate the given data completely, by filling in binary predicates (which only involve thresholding of a *single* variable) in the boxes for the decision trees below. If the data is already separated when you hit a box, simply write the class, and leave the sub-tree hanging from that box empty.
    2. Draw the corresponding decision boundaries on the scatter plot, and write the class labels for each of the resulting bins somewhere inside the resulting bins.

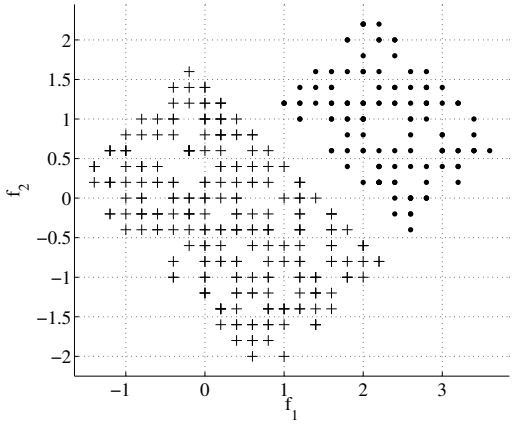
    If the data can not be separated completely by a depth 2 decision tree, simply cross out the tree template. We solve the first part as an example.

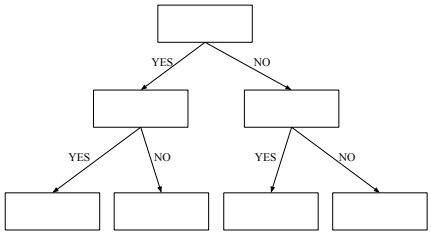

Dot
← Decision Boundary
Plus
f2
f1

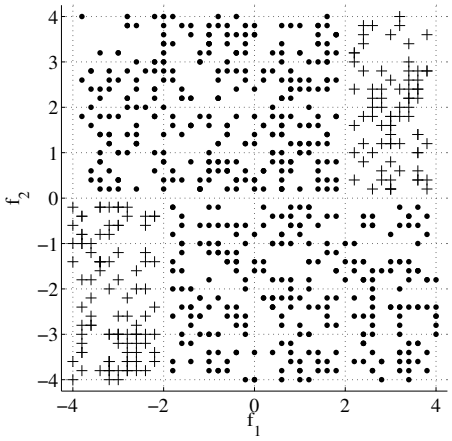

f1 < 0.5
YES
NO
Dot
Plus
YES
NO
YES
NO

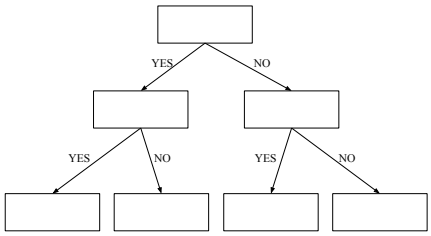

f2
f1


YES
NO
YES
NO
YES
NO


f2
f1


YES
NO
YES
NO
YES
NO


f2
f1


YES
NO
YES
NO
YES
NO