CS 189 Spring 2019 Introduction to Machine Learning

Exam Prep Dis 1 Solutio

Q1. Multiple Choice

(a) Which of the following are true statements about choosing the split for a decision tree?

- ● For any split, the information gain is non-negative.
- ○ It is always possible to choose a split on any feature.
- ○ For any split, neither child can have higher entropy than the parent.
- ○ The information gain from the split will be zero if and only if one of the children is empty.

Top left: Information gain is non-negative due to the concavity of the entropy curve

Top right: It is possible that a feature can only take on one value, e.g. the feature takes on values from a set with a single element. If that's the case, you cannot even define a decision boundary.

Bottom left: Lecture 15 has an example of this; the weighted average must be lower but it is in fact possible for the child to have higher entropy

Bottom right: The information gain may also be zero if every class is equally divided between the two children

(b) Which of the following are true statements about decision trees and random forests?

- ● Random forests have lower variance than a single decision tree.
- ● Decreasing the number of randomly chosen features available for each split in a random forest will increase the bias.
- ○ Using bagging to train an ensemble of decision trees has lower bias than training a single decision tree.
- ○ Decreasing the maximum depth of a decision tree will decrease its bias.

Top left: Introducing multiple trees makes it less likely that the overall classifier is sensitive to small changes in the data

Bottom left: Bagging will actually typically increase the bias (but only slightly)

Top right: Choosing fewer features limits the possible decision boundaries of your model; decreasing the features available will thus increase the bias

Bottom right: Decision trees of smaller depth will have more bias

(c) Suppose you train a decision tree with high training error. To improve it, you decide to retrain using early stopping conditions, where we leave the current node as a leaf if any of the following are true: depth > 10, > 90% of sample points in the node are in one class, < 5 sample points in the node. Which of the following changes to these stopping conditions, done in isolation, could decrease your training error?

- ● Stop when depth > 15.
- ○ Stop when < 10 sample points in node.
- ● Stop when > 95% of sample points are of one class.
- ● Stop when < 3 sample points in node.

(d) Which of the following are true statements about the entropy of a discrete probability distribution?

- ● It is a useful criterion for picking splits in decision trees.
- ○ It is a convex function of the class probabilities.
- ● It is maximized when the probability distribution

is uniform.

◯ It is minimized when the probability distribution is uniform.

Q2. 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?

Imagine we are deciding to split on a specific real-valued feature. Obviously, we should choose every value of the data to be the threshold for the split. Choosing a split value between two values in our data will give the same splitting of the data. For instance, imagine that we have data:

$$\begin{bmatrix} 5 & 0 \\ 6 & 0 \\ 7 & 1 \end{bmatrix}$$

where the first column is the features, and the second column is the labels (we have three data points). Note that splitting on 6.5 and 6.3 will result in the same splits. Thus, we want to choose each feature value for a split.

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

We must search through every possible split to find the best one, and there are dn of them, where d is the dimensionality of our data and n is the number of data points. For every split value, we must walk through all n data points and classify them accordingly. We can calculate the information gain from these two new subsets of our data in linear time. Thus, we have $O(dn^2)$.

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

Before we start scanning through a feature, we can sort the data with respect to that feature value. Then, every time we choose a new split value, only one data point will be classified differently. Calculating the new entropies can be done in constant time. Assuming we use a comparison based sorting algorithm, our new running time is $O(dn \log n)$.

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?)

We can use sample variance to determine how well our data is split. This is the sum of the squared error from the average.

(b) Write down the equation we want to maximize when searching over splits for regression trees. This equation looks very similar to our information gain equation, except the entropies are replaced with variances. We have

$$\sigma - \left[\frac{n_1 \sigma_1 + n_2 \sigma_2}{n_1 + n_2}\right]$$

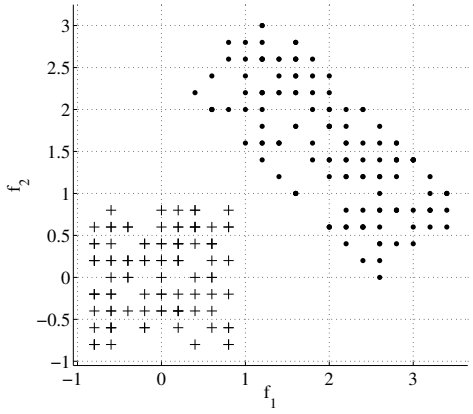
where σ is the variance at the current node, σ_1 is the variance at the "left" split, and σ_2 is the variance at the "right" split.

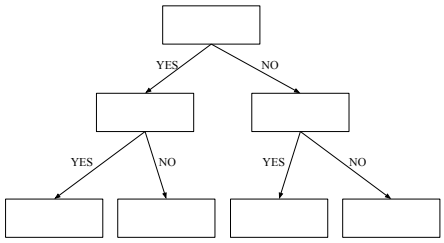
Q3. More Decision Trees

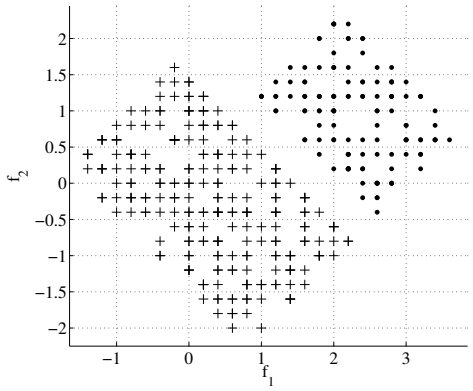
You are given points from 2 classes, shown as +'s and ·'s. For each of the following sets of points,

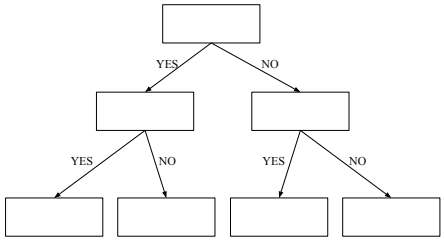
1. Draw the decision tree of depth at most 2 that can separate the given data completely, by filling in binary predicates (which only involve thresholding of a *single* variable) in the boxes for the decision trees below. If the data is already separated when you hit a box, simply write the class, and 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.

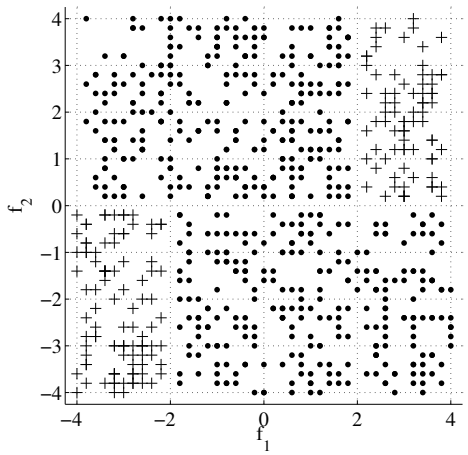
If the data can not be separated completely by a depth 2 decision tree, simply cross out the tree template. We solve the first part as an example.

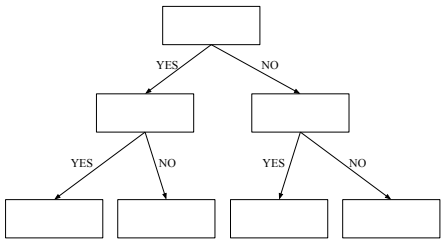


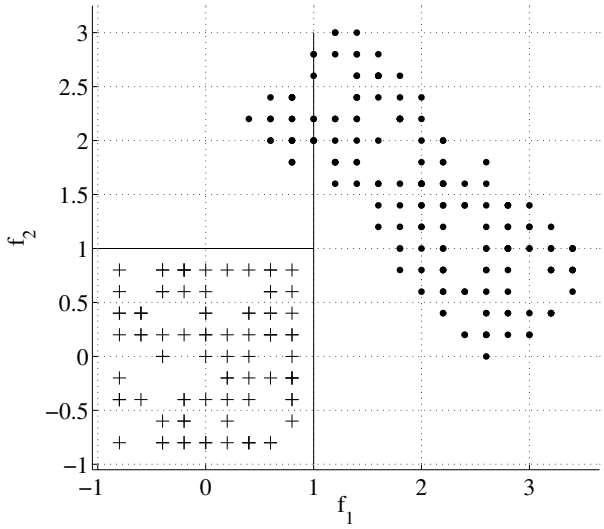


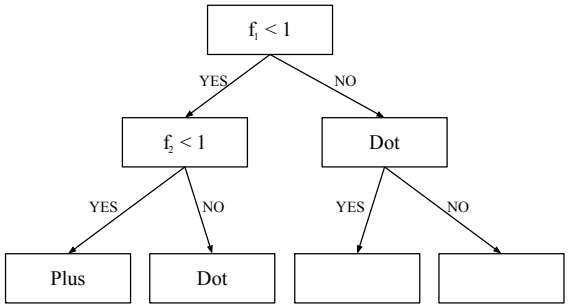


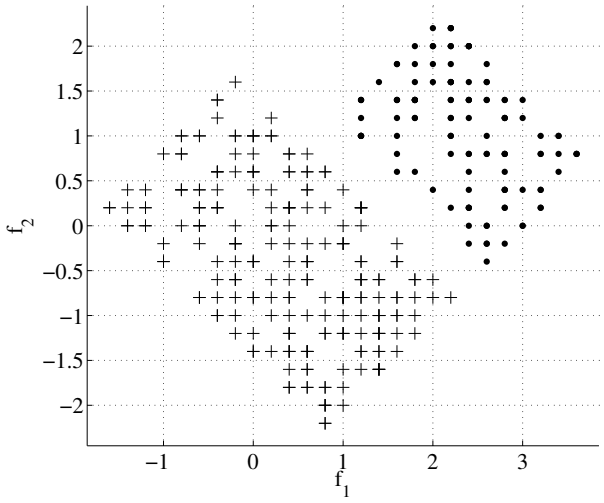


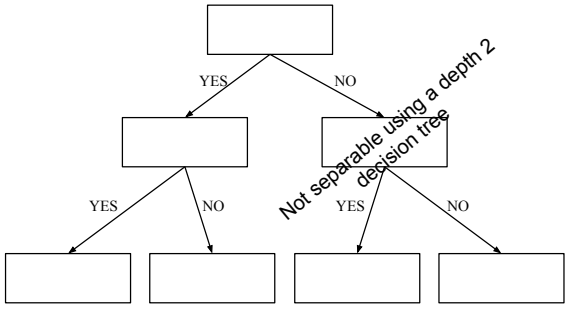


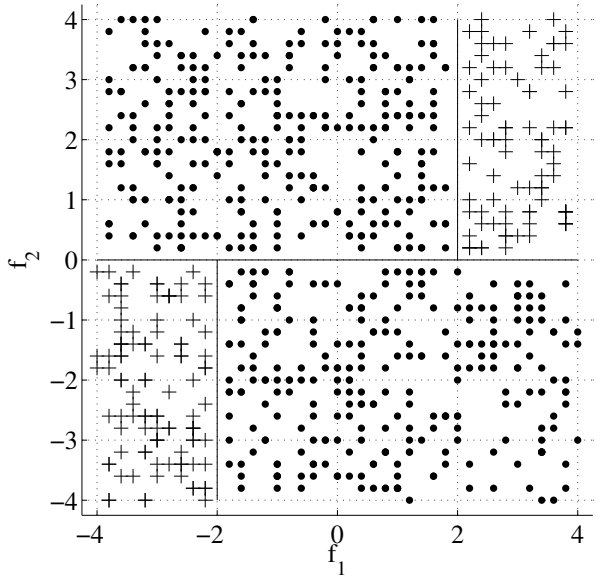


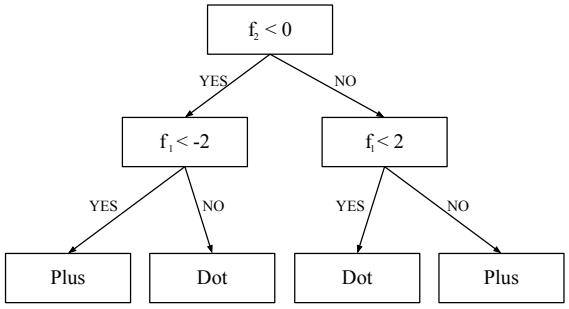

f2
f1


f1 < 1
YES
NO
f2 < 1
Dot
YES
NO
YES
NO
Plus
Dot


f2
f1


YES
NO
YES
NO
YES
NO
Not separable using a depth 2 decision tree


f2
f1


f2 < 0
YES
NO
f1 < -2
f1 < 2
YES
NO
YES
NO
Plus
Dot
Dot
Plus