CS 536 : Decision Trees

16:198:536

In both sets of problems for this homework, we take $\{(\underline{X}_1, Y_1), (\underline{X}_2, Y_2), \ldots, (\underline{X}_m, Y_m)\}$ to denote a data set, where $\underline{X}_i$ represents a vector of k (binary) feature values, and Y_i is a corresponding binary class or label that we need to learn and be able to predict from the $\underline{X}$-values. In both sections, we will generate simulated data, and then attempt to fit a decision tree to that data set.

1 Generating Decision Trees

We generate data via the following scheme, defining a distribution for our data set: Let $\underline{X} = (X_1, X_2, X_3, \ldots, X_k)$ be a vector of binary values, satisfying the following

- $X_1 = 1$ with probability 1/2, $X_1 = 0$ with probability 1/2
- For $i = 2, \ldots, k$, $X_i = X_{i-1}$ with probability 3/4, and $X_i = 1 - X_{i-1}$ with probability 1/4.

In this way, the first feature value is uniformly random, but every successive feature is strongly correlated with the value of the feature before it. We can then define Y to be a function of $\underline{X}$ as

$$Y = \begin{cases} X_1 \text{ if } w_2 X_2 + w_3 X_3 + \ldots + w_k X_k \geq 1/2 \\ 1 - X_1 \text{ else.} \end{cases} \tag{1}$$

In other words, if the 'weighted average' of $X_2, \ldots X_k$ tilts high, Y will agree with X_1; if the weighted average of $X_2, \ldots, X_k$ tilts low, Y will disagree with X_1. Take the weights to be defined by $w_i = 0.9^i / (0.9^2 + 0.9^3 + \ldots + 0.9^k)$.

1) For a given value of k, m, (number of features, number of data points), write a function to generate a training data set based on the above scheme.

2) Given a data set, write a function to fit a decision tree to that data based on splitting the variables by maximizing the information gain (**ID3**). Additionally, return the training error of this tree on the data set, $\text{err}_{\text{train}}(\hat{f})$ (*Hint: this should be easy - why?*). *It may be useful to have a function that takes a data set and a variable, and returns the data set partitioned based on the values of that variable.*

3) For $k = 4$ and $m = 30$, generate data and fit a decision tree to it. Does the ordering of the variables in the decision tree make sense, based on the function that defines Y? Why or why not? Draw the tree.

4) Write a function that takes a decision tree and estimates its typical error on the underlying distribution $\text{err}(\hat{f})$; i.e., generate a lot of data according to the above scheme, and find the average error rate of this tree over that data.

5) For $k = 10$, estimate the value of $|\text{err}_{\text{train}}(\hat{f}) - \text{err}(\hat{f})|$ for a given m by repeatedly generating data sets, fitting trees to those data sets, and estimating the true and training error. Do this for multiple m, and graph this difference as a function of m. What can you say about the marginal value of additional training data?

6) Design an alternative metric for splitting the data, not based on information content / information gain. Repeat the computation from (5) above for your metric, and compare the performance of your trees vs the ID3 trees.

2 Pruning Decision Trees

We generate data via the following scheme, defining a distribution for our data set: Let $\underline{X} = (X_0, X_1, X_2, X_3, \ldots, X_{20})$ be a vector of binary values, satisfying the following

- $X_0 = 1$ with probability $1/2$, $X_0 = 0$ with probability $1/2$
- For $i = 1, \ldots, 14$, $X_i = X_{i-1}$ with probability $3/4$, and $X_i = 1 - X_{i-1}$ with probability $1/4$
- For $i = 15, \ldots, 20$, $X_i = 1$ with probability $1/2$, $X_i = 0$ with probability $1/2$.

The first feature is uniformly random, and the next 14 features are strongly correlated, but the last 5 features are independent of everything else. There are 21 X-variables, so there are $2^{21} \approx 2$ mil possible input $\underline{X}$. Some of these are more likely than others. In general, we expect the training data to cover only a fraction of the total possible inputs, so consider data sets of size m where m ranges from 10 to $10,000$. We then define Y to be

$$Y = \begin{cases} \text{majority}(X_1, \ldots, X_7) \text{ if } X_0 = 0 \\ \text{majority}(X_8, \ldots, X_{14}) \text{ if } X_0 = 1. \end{cases} \tag{2}$$

That is, if $X_0 = 0$, we take the majority value of X_1 through X_7 - otherwise we take the majority value of X_8 through X_{14}. The values X_{15} through X_{20} are nothing but noise.

1) Write a function to generate m samples of $(\underline{X}, Y)$, and another to fit a tree to that data using **ID3**. Write a third function to, given a decision tree f, estimate the error rate of that decision tree on the underlying data, $\text{err}(f)$. Do this repeatedly for a range of m values, and plot the 'typical' error of a tree trained on m data points as a function of m. Does this agree with your intuition?

2) Note that X_{15} through X_{20} are completely irrelevant to predicting the value of Y. For a range of m values, repeatedly generate data sets of that size and fit trees to that data, and estimate the average number of irrelevant variables that are included in the fit tree. How much data would you need, typically, to avoid fitting on this noise?

3) Generate a data set of size $m = 10000$, and set aside 8000 points for training, and 2000 points for testing. The remaining questions should all be applied to this data set.

   a) **Pruning by Depth:** Consider growing a tree as a process - running ID3 for instance until all splits up to depth d have been performed. Depth $d = 0$ should correspond to no decisions - a prediction for Y is made just on the raw frequencies of Y in the data. Plot, as a function of d, the error on the training set and the error on the test set for a tree grown to depth d. What does your data suggest as a good threshold depth?

   b) **Pruning by Sample Size:** The less data a split is performed on, the less 'accurate' we expect the result of that split to be. Let s be a threshold such that if the data available at a node in your decision tree is less than or equal to s, you do not split and instead decide Y by simple majority vote (ties broken by coin flip). Plot, as a function of s, the error on the training set and the error on the testing set for a tree split down to sample size s. What does your data suggest as a good sample size threshold?

5) Repeat the computation of Problem 2, growing your trees only to depth d as chosen in 3.a. How does this change the likelihood or frequency of including spurious variables in your trees?

6) Repeat the computation of Problem 2, splitting your trees only to sample size s as chosen in 3.b. How does this change the likelihood or frequency of including spurious variables in your trees?