question 16 views

Problem with provided data!!

If you download the provided data from mimir (adult.data from the starter code), then search for the string below:
"Private,Some-college,Married-civ-spouse,Tech-support,Wife,White,Female,United-States,"

you will see that there are multiple lines with this exact same data, but they have different classifications.
For example:

- *line 7734:*
  Private,Some-college,Married-civ-spouse,Tech-support,Wife,White,Female,United-States,>50K

- *line 23519:*
  Private,Some-college,Married-civ-spouse,Tech-support,Wife,White,Female,United-States,<=50K

Perhaps I am misunderstanding something, but it seems like this would mess up our decision tree training. I would think that each unique vector would have to always have the same classification?

When I'm creating my decision tree it gets messed up when it reduces the remaining data points down to:

```
7137-> 1 [0, 1, 0, 0, 0, 0, 0, 0]
19391-> 1 [0, 1, 0, 0, 0, 0, 0, 0]
21793-> 0 [0, 1, 0, 0, 0, 0, 0, 0]
24082-> 1 [0, 1, 0, 0, 0, 0, 0, 0]
24868-> 1 [0, 1, 0, 0, 0, 0, 0, 0]
26241-> 0 [0, 1, 0, 0, 0, 0, 0, 0]
27029-> 1 [0, 1, 0, 0, 0, 0, 0, 0]
27490-> 0 [0, 1, 0, 0, 0, 0, 0, 0]
```

where the first number is the index in data, the second number is the data point's label, and the last part is the actual data member printed.
At this point the only attributes that the tree hasn't split on yet are 2, 6, and 7, but each has 0 gain. Yet the decision tree doesn't think it's done because the set of remaining data points have different classifications.

Is this a problem with the provided data or should my decision tree do something different here like just take the most common label and stop branching here?

project

Updated 1 day ago by Daniel Engbert

the students' answer, *where students collectively construct a single answer*

Click to start off the wiki answer

the instructors' answer, *where instructors collectively construct a single answer*

That's not a problem; real world data is not going to perfect. The decision tree built is not going to be able to classify everything correctly, and that is reflected in the tests (there is a threshold of correctness that is considered 'good enough').

Updated 1 day ago by Michael Neary

followup discussions *for lingering questions and comments*

● Resolved ○ Unresolved

Daniel Engbert 1 day ago
So should I have the tree stop branching at that point and in the future just classify everything that reaches that point in the tree as whatever is the most common label for the items that are remaining?

Michael Neary 1 day ago Yes, if you get to a point where you don't gain any information from any attribute it should be sufficient to just go with the majority class label of the subset you're looking at.

Daniel Engbert 1 day ago Got it, thanks for the quick answer!

James Sheehan 7 hours ago This may not help or you might already be aware, but to prevent your tree from continuing to branch in these situations, you can just check for an information gain of 0 across all attributes. Alternatively you could keep track of the attributes you have already hit but that is a bit harder.

● Resolved ○ Unresolved

Felipe Bastos 17 hours ago
What do we do if the labels for our data points are split exactly in half (i.e., the entropy of the dataset is 1)? Rounding 0.5 up or down makes the tree super biased towards classifying a sample whichever way you rounded

Michael Neary 17 hours ago Interesting point to bring up. Are you experiencing this in your tree? I can think of two ways to handle this: you can either hard code it to choose a specific one every time, or introduce some randomness (choose one label half the time, choose the other half the time).

Felipe Bastos 17 hours ago Yes; in the given dataset, 776 times. I originally opted for "rounding" 0.5 up (i.e., setting the return value of the node to 1 if entropy was exactly 0.5), but that made the tree heavily biased towards classifying samples as earning over 50k; similarly, assigning 0 to the return value when the entropy is exactly 0.5 made it heavily biased towards classifying samples as earning less than 50k.

I also thought of introducing randomness, but I'm not a fan of relying on nondeterministic behavior for passing tests (or at all); is there no other way to handle this, here?

For an idea of how biased the tree becomes, either way: of the 11 samples I'm testing against (the one given in the test driver + 10 from Mimir), only three are classified regardless of rounding (two positively, one negatively).

Ryan Hartig 15 hours ago The lowest certainty I get on the test set u is 70%

James Sheehan 7 hours ago I explicitly tried both rounding 1 way or the other in the event of 50/50 ties and neither affected my results on Mimir. There are several unresolved leaf nodes, but none in the test should be 50/50.

Michael Neary 3 hours ago If you are getting less than 70% on the known test case there might be something wrong with the way you build the tree in general and not just with the 50/50 split case

Felipe Bastos 3 hours ago Yeah, I realized that an hour ago. I hate python and I hope all of guido's computers explode