Robert F.K. Martin
ID 1505151
Csci 5523
05 Oct 2011
Homework 2

Problem 1a: What is the overall Gini index?

The samples are evenly split to the Gini index is 0.5.

Problem 1b: What is the Gini index for Movie ID?

Since all the training examples are unique classes, every split will create a leaf with no impurity. Therefore, the Gini index for this attribute is 0. While mathematically this would make it seem like a good candidate for splitting, it is obvious this model would not extend well to test data.

Problem 1b: What is Gini index for Movie Format?

There are 6 DVDs, 4 which are in C0 and 2 in C1. There are 14 Online movies, 6 of which are C0 and 8 in C1.

$$\frac{6}{20} * \left(1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2\right) + \left(\frac{14}{20}\right) * \left(1 - \left(\frac{6}{14}\right)^2 - \left(\frac{8}{14}\right)^2\right) = .4762$$

Problem 1d: What is the Gini index for Movie Category using multiway split?

$$\frac{5}{20} * \left(1 - \frac{4}{25} - \frac{9}{25}\right) + \frac{9}{20} * \left(1 - \frac{36}{81} - \frac{9}{81}\right) + \frac{6}{20} * \left(1 - \frac{4}{36} - \frac{16}{36}\right) = .4533$$

Problem 1e: Which of the three has the lowest Gini index?

The Movie ID has the lowest Gini index but as mentioned, this is not the best choice since it will not extend well to new data.

Problem 1f: Which of the three attributes would you choose?

Since I've already said I would not choose Movie ID, I would choose Movie Category since it has a slightly lower Gini index even though it is a multiway split. This will result in more pure data in the successive nodes but possibly a more complex tree overall.

Problem 2 1a: Generalization errors.

$e(T)_{training,optimistic} = \frac{3+1+7+6+1+4}{100} = 0.22;$

$errors(T)_{validation} = 0 + 4 + 3 + 5 + 5 + 6 = 23$;
6 leaves $* 0.5 = 3$;
$e(T)_{validation,pessimistic} = \frac{23+3}{100} = 0.26$;

Problem 2 1b: Pruning

$e(T) = \frac{0+1+3+5+2+1}{100} = 0.12$;

Problem 2 1c: Comment on overfitting.

If we look at the XY branch of the training and validation sets, we see they are nearly opposite(1,5 vs 4,1). Also, although the distribution of the Z branch is nearly the same(11,14 vs 7,8), the resulting leaves are again nearly opposite(5,2 vs 1,10 and 10,4 vs 1,6). So for these particular branches the tree does poorly on the validation test set. We would be better off pruning these leaves and use the majority rule at the split.

Problem 2 2a: Training error T1 > Training error T2.

This cannot be true since we are only creating T2 from T1 by pruning. T1 only creates a new leaf if it reduces the classification error. If we remove a leaf from T1, we increase the classification error.

Problem 2 2b: min(test error T2, test Error T3) $\leq$ test error T1.

This could be true if the validation data wasn't a good representation of the training data. If could be that even through T2 or T3 had a higher classification error, it could do a better job on unseen data.

Problem 2 2c: Test error T2 $\leq$ test error T3.

This could be true if the unseen data of the test set isn't a good representation of the training data and we could have any combination of test errors for the new trees.

Problem 3a: Compute accuracy, precision, recall, and F-measure with respect to - class.

precision $= \frac{850}{850+50} = .944$
recall $= \frac{850}{850+80} = .9140$
f-measure $= 2 * \frac{.944*.914}{.944+.914} = .929$
accuracy $= \frac{20+850}{1000} = .870$

Problem 3b: Which is a good metric? Which is a bad metric?

Precision, recall, and f-measure are not the best metrics for overall performance since they all

only use three of the four data points in the confusion matrix. However, accuracy does use all four data points and is a much better reflection of overall performance for that reason.

Problem 3c: Construct a better classifier.

Simply choose 'no rain' all the time. The resulting confusion matrix would look like:

0	70
0	930

This would result in an accuracy of 0.930.

Problem 4a: Would a person more likely go for a run or not?

The only data we have is the class labels of Run=Yes and Run=No. (Run=No)= $\frac{4}{14}$. (Run=Yes)= $\frac{10}{14}$. So a person is more likely to go for a run than not.

Problem 4b: How would this classify X={Sunny, Mild, Normal}?

P(Sunny—No) = 0.5;
P(Mild—No) = 0.5;
P(Normal—No) = 0.25;
P(No—Sunny, Mild, Normal) = $0.5 * 0.5 * 0.25 * \frac{4}{14} = 0.0179$;
P(Sunny—Yes) = 0.3;
P(Mild—Yes) = 0.3;
P(Normal—Yes) = 0.6;
P(Yes—Sunny, Mild, Normal) = $0.3 * 0.3 * 0.6 * \frac{10}{14} = 0.0386$; Greater than 0.0179 so this person will run.

Problem 4c: Assume you only know it is mild. Run or not?

P(Mild—No) = 0.5;
P(No—Mild) = $0.5 * \frac{4}{14} = 0.1429$;
P(Mild—Yes) = 0.3;
P(Yes—Mild) = $0.3 * \frac{10}{14} = 0.2143$; Greater than 0.1429 so this person will run.

Problem 4d: Assume you know it is mild and high humidity. Run or not?

P(Mild—No) = 0.5;
P(High—No) = 0.75;
P(No—Mild, High) = $0.5 * 0.75 * \frac{4}{14} = 0.1071$;
P(Mild—Yes) = 0.3;
P(High—Yes) = 0.4;
P(Yes—Mild, High) = $0.3 * 0.4 * \frac{10}{14} = 0.0857$; Less than 0.1071 so this person will not run.

Problem 4e: Discuss missing data handling.

The classifier just ignores the missing data as if it wasn't relevant to the decision. As we add in more data, we find that our probablities get smaller and smaller due to having more information to make a decision.

Problem 4f: X={Overcast, Mild, High}. Run or not?

P(Overcast—No) = 0.5;
P(Mild—No) = 0.5;
P(High—No) = 0.75;
P(No—Overcast, Mild, High) = $0.5 * 0.5 * 0.75 * \frac{4}{14} = 0.0536$;
P(Overcast—Yes) = 0;
P(Mild—Yes) = 0.3;
P(High—Yes) = 0.4;
P(Yes—Overcast, Mild, High) = $0 * 0.3 * 0.4 * \frac{10}{14} = 0$; Less than 0.0536 so this person will run.

Problem 4g: What went wrong in f? How would you fix it?

There was no data for running on an overcast day. So any classification for running on an overcast day will always decide no. To overcame this, we can either use the Laplace substition or m-estimate so that the resulting probability for P(Overcast—Yes) is small but non-zero.

Problem 5ai: Which aspects would lead you to use decision trees over knn classifier?

The data in the data set come in a variety of types: nominal, ordinal, discrete, and continuous. And of the discrete and continuous attributes, there are several different ranges. These two facts would make it challenging to use a nearest neighbors type classifier since it is difficult to measure distances with these types of attributes.

Problem 5aii: Which aspects would lead you to use a Bayes network over a Naive Bayes?

The fact that many of the attributes are correlated would violate the major assumption of the naive Bayes. Wheelbase and length are correlated with type. Engine size, horsepower, and cylinders are correlated with MPG.

Problem 5b: Would you choose naive Bayes or knn for a dataset with missing data?

I would choose naive Bayes as it is better at handling missing data. Missing data for a knn

classifier potentially means a miscalculation of the actual nearest neighbors.

Problem 5c: Use Ripper or knn for predicting credit risk?

I would choose Ripper. Knn is difficult to use with differently scaled attributes or nominal and ordinal data.

Problem 6a: Which are suitable(NB, decision tree, knn)?

NB would probably be best since it is robust to irrelevant data, which it looks like the majority of the table is made up of. A decision tree also looks like it would work well since the classes are pretty well rectilinearly separated. The worst choice would be knn because of the noise.

Problem 6b: Which are suitable(NB, decision tree, knn)?

Knn would be the most appropriate choice here because of the geometric shape of the data. there is no way to discriminate between classes in a linear way.

Problem 6c: Which are suitable(NB, decision tree, knn)?

Decision trees would work here because the data is separated linearly.