PS2 Machine Learning: Project

Cai Glencross and Katie Li

February 7, 2018

1 Evaluation:

(a) Train and evaluate a DecisionTreeClassifier (using the class from scikit-learn and referring to the documentation as needed). Make sure you initialize your classifier with the appropriate parameters; in particular, use the 'entropy' criterion discussed in class. What is the training error of this classifier?

Training error for sci-kit learn decision tree: .014

(b) So far, we have looked only at training error, but as we learned in class, training error is a poor metric for evaluating classifiers. Let us use cross-validation instead. Implement the missing portions of error(...) according to the provided specifications. You may find it helpful to use train_test_split(...) from scikit-learn. To ensure that we always get the same splits across different runs (and thus can compare the classifier results), set the random_state parameter to be the trial number.
Next, use your error(...) function to evaluate the training error and (cross-validation) test error of each of your three models. To do this, generate a random 80/20 split of the training data, train each model on the 80% fraction, evaluate the error on either the 80% or the 20% fraction, and repeat this 100 times to get an average result. What are the average training and test error of each of your classifiers on the Titanic data set?

MajorityVoteClassifier average training error: 0.403
MajorityVoteClassifier average test error: 0.410
RandomClassifier average training error: 0.484
RandomClassifier average test error: 0.486
DecisionTree average training error: 0.011
DecisionTree average test error: 0.241

(c) (2 pts) One problem with decision trees is that they can overfit to training data, yielding complex classifiers that do not generalize well to new data. Let us see whether this is the case for the Titanic data. One way to prevent decision trees from overfitting is to limit their depth. Repeat your crossvalidation experiments but for increasing depth limits, specifically, 1, 2, . . .

, 20. Then plot the average training error and test error against the depth limit. (Also plot the average test error for your baseline classifiers. As the baseline classifiers are independent of the depth limit, their plots should be flat lines.) Include this plot in your writeup, making sure to label all axes and include a legend for your classifiers. What is the best depth limit to use for this data? Do you see overfitting? Justify your answers using the plot.

The best depth to use is around 6. Yes, we can see overfitting because while the training error continues to decrease after depth=6, the test error starts to increase.

(d) Another useful tool for evaluating classifiers is learning curves, which show how classifier performance (e.g. error) relates to experience (e.g. amount of training data). Run another experiment using a decision tree with the best depth limit you found above. This time, vary the amount of training data by starting with splits of 0.05 (5% of the data used for training) and working up to splits of size 0.95 (95% of the data used for training) in increments of 0.05. Then plot the decision tree training and test error against the amount of training data. (Also plot the average test error for your baseline classifiers.) Include this plot in your writeup, and provide a 1-2 sentence description of your observations.

We can see that the majority vote classifier needs at least 20% of the data to be able to perform well. We can also see that test error for the decision tree exponentially decreases with increased training data size, leveling off at around 80-90%. we can also see that training error increases with increased training size, which makes sense because for example with only one person to train on it is very easy to predict who survives with 100% accuracy.

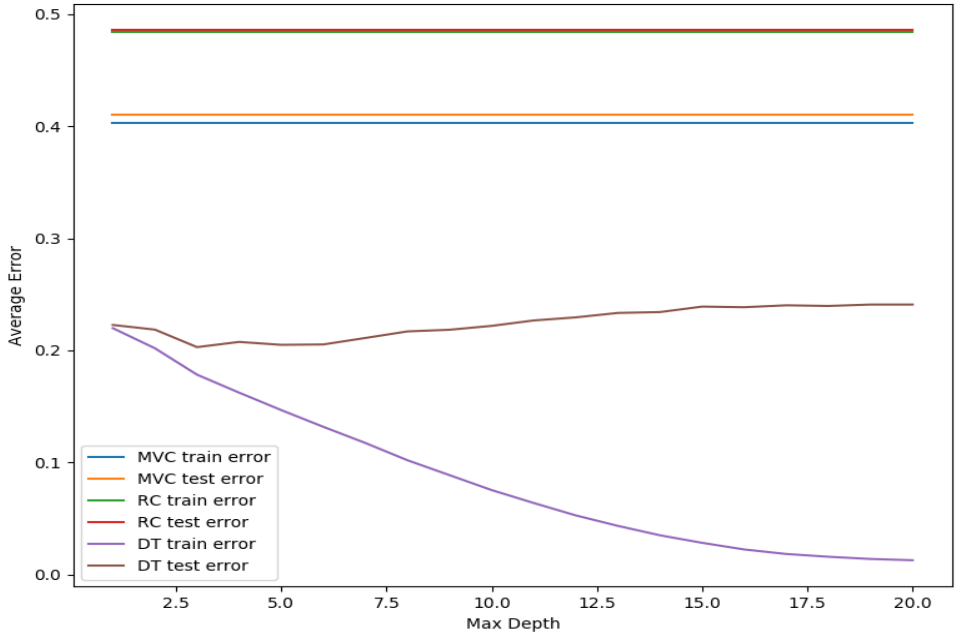

Figure 1: Problem (1c) Average training and test error for various classifiers.

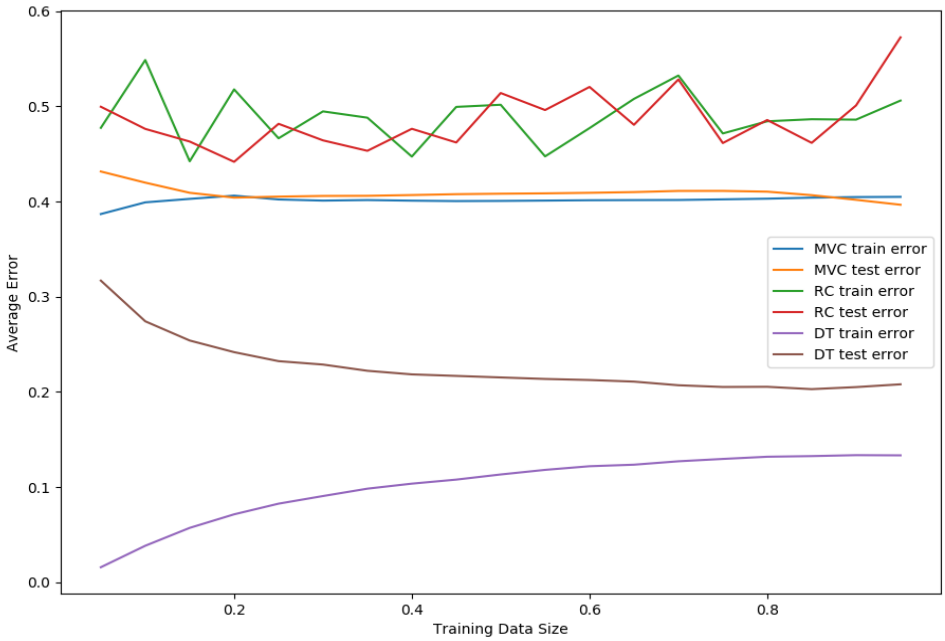

Figure 2: Problem (1d) - The effect of training data size on training and test error for various classifiers.