CS 289 Homework 5

Jiaying Shi

(SID:24978491)

Decision Trees is one of the most intuitive models for classification or regression. A greedy decision tree algorithm and random forest are implemented in Java. The code is tested using the spam data set. The code can be devided into two parts. The first part of the code is about processing the data and obtaining features of the data set. The second part of the code is implementing the algorithm.

Data Processing

In this homework, I write a data process package by myself instead of using the features obtained using the python code given with the data set. In data processing, bag of words is implemented. A set of all the words in the training data is first generated. Then the frequency of those words is counted and will be use as features for a particular data point. When we take punctuation into consideration, there are over 50,000 features. When we build decision trees using all these features, there may be over-fitting and the time of training and predicting is long. To reduce the number of features, we can only keep the words that appears in over 30 data points. Then there are only features left. Since all the features are frequency of such a words in an email, the value of all features are integers. When we split the data points at a node in the decision tree, we can split those integer values. In the decision tree model, at the root node, we use a method to select the best feature and the boundary value. And recursively, similar things happens in the following nodes until the stopping criteria meets. In random forest, for each tree, the training data is a random subset of the original training set. At each node, features used to split the data are also randomly picked from the original features. The prediction of random forest is the average of the predictions of all the trees.

Decision Tree

A decision Tree algorithm is implemented in Java. The details about the code are listed as below.

Stopping Criteria

There are 5 stop criteria in my decision tree model:

- If maximum node number exceeds a certain number, the training will be terminated.

- If at a node the number of data points is smaller than a value, the data will not be splitted at this node.
- If at a node the depth is larger than some value, the data will not be splitted at this node.
- If all the labels of the data points at a node are the same, the data will not be splitted at this node.
- If at a node the over 98% nodes has the same label, the data will not be splitted at this node.

The purpose of having these stopping criteria is to avoid over-fitting. For example, if the number of data points at a specific node is smaller than some value, splitting at that node will cause over fitting since the data size is to small to be representative. So these stopping criteria helps to reduce the variance.

Spliting Criteria

The splitting criteria of my model is find the feature and feature value such that, if the data is splitted using that feature by that value, the information gain will be maximized. In the code, the information gains of all features at different values are calculated.

Features

Bag-of-words is implemented. And as mentioned before, to reduce the training time, only those words that appear in over 30 data points are kept as features. There are 2260 features used in training the decision tree. Those words includes:

```
subject, re:, from, production, high, inc, forwarded, cc,
@, and, for call, increase, following, information, ...
```

The frequency of those words in a email is stored as features of a data points.

Performance

The decision tree and random forest are tested using the files in folder in "/spam-test" and "/ham-test". The error rate of decision tree is **16.04%.** While the error rate of random forest is much smaller, only **9.02%**. The Kaggle data is tested on random forest, and the score is **0.83572.**

Example of the classification of one data point

For a data point 11.txt in Kaggle data set, the split of that point is:

- Node 0: Feature "cc">1
- Node 1: Stop branch, label "ham"

A email with "cc" can never be a spam. It is usually send by some one and copy to another one which can never happen for a spam.

Random Forest: Most Common Top Level Splits

The most common top level splits in random forest with 50 trees is listed here:

- Feature "cc" >=1 13 trees
- Feature "forwarded" >=1 5 trees
- Feature "thanks">=1 5 trees
- Feature "gas">=1 5 trees
- Feature "etc">=1 4 trees
- Feature "am">=1 4 trees
- Feature "pm">=1 3 trees
- Feature "hou" >=1 2 trees
- Feature "subject" >=2 2 trees
- Feature "deals" >=1 1tree
- Feature "attached">=1 1tree
- Feature "let" >=1 1 tree
- Feature "questions" >=1 1 tree
- Feature "free">=1 1 tree
- Feature " bob" >=1 1 tree
- Feature "here" >=1 1 tree

From the above results, we can see that words that indicate a email is forwarded or cc to someone or with time (so that include am/pm) can ensure one child node to be leaf. If there are many emails contains those words in the training set, to maximize the information gain, those words are always first chosen.

Summarize:

Bag-of-words, decision tree and random forest are implemented in Java in this homework. The code is tested using the spam data set. From the test results, we can see that random forest can improve the performance significantly. That is because random forest combines the "bagging" idea and the random selection of features and it is can construct a collection of decision trees with controlled variance.