Data Set Description

General Information

This data set includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes: voted for, paired for, and announced for (these three simplified to yea), voted against, paired against, and announced against (these three simplified to nay), voted present, voted present to avoid conflict of interest, and did not vote or otherwise make a position known (these three simplified to an unknown disposition).

Attributes information

1. handicapped-infants: 2 (y,n)
2. water-project-cost-sharing: 2 (y,n)
3. adoption-of-the-budget-resolution: 2 (y,n)
4. physician-fee-freeze: 2 (y,n)
5. el-salvador-aid: 2 (y,n)
6. religious-groups-in-schools: 2 (y,n)
7. anti-satellite-test-ban: 2 (y,n)
8. aid-to-nicaraguan-contras: 2 (y,n)
9. mx-missile: 2 (y,n)
10. immigration: 2 (y,n)
11. synfuels-corporation-cutback: 2 (y,n)
12. education-spending: 2 (y,n)
13. superfund-right-to-sue: 2 (y,n)
14. crime: 2 (y,n)
15. duty-free-exports: 2 (y,n)
16. export-administration-act-south-africa: 2 (y,n)
17. Class Label: 2 (democrat, republican)

Original data set link is https://archive.ics.uci.edu/ml/datasets/Congressional+Voting+Records
In this implementation, I converted the data into arff format and replace the missing value with mode of the attribute

Training Process

We can train a decision tree by using either ID3 or C4.5, for example, in the function **public void kFoldsCrossValidation**(**int** foldsNum,**Instances** dataSet), we can train the model by either use or trainingDecisionTreeByID3(trainSet) trainingDecisionTreeByC4_5(trainSet);

ID3 Training is performed in **public void trainingDecisionTreeByID3(Instances trainSet), which call the function private void buildTreeNodeByID3(Instances dataSet),** which recursively call itself to build successor tree nodes or stop until the stop condition of growing is met. **buildTreeNodeByID3(Instances dataSet)** calculates the information gain of all the attribute in the "dataSet" to find attribute with maximum information gain to be the splitting attribute. When calculating the information gain of every possible splitting attribute, the function **private double getInfoGain**(**Instances** dataSet, **Attribute** attrib) is used.

The information gain is determined by total entropy Info(dataSet) and the conditional entropy $\text{Info}_{\text{attribute}}$(dataSet). When calculating $\text{Info}_{\text{attribute}}$(dataSet), **Utility.splitDataSets(Instances dataSet, Attribute attrib)** is used to return all sub data set of "**dataSet**". Every sub data set has the same value of "**attrib**".

The training process of C4.5 is the same as that of ID3, except that C4.5 uses maximum gain ratio to select the splitting attribute.

Classification and Performance

When training is completed, we can use the function **public double classify**(**Instance** dataEntry) to classify a test example.

The average accuracy of ID3 trained model and C4.5 trained model by using 5-fold cross-validation is 93.10% and 93.56% respectively.

Reference

This implementation is based on modification of the course material of Northeastern University CS6220 Data Mining Techniques.