Xudong LIU

7lxd@tongji.edu.cn

School of Software Engineering

Tongji University

Oct 31, 2015

Decision Tree with ID3 Learning Algorithm

ABSTRACT

There are many classifiers in Data Mining Field, Decision tree classifiers is a very classic one used widely in variety of systems. This paper is going to propose ID3 algorithm based Information Gain Criteria. I divided the **Iris data** (Data from UCI Machining Learning Repository) into two subset, one as the training set and another testing set. I used the training set to modelling the decision tree based on ID3 algorithm. Then use the testing set test the accuracy of the decision tree model which I just built. What's more I also compared our decision tree with the Matlab default decision tree based on GINI index. It is empirically shown that the ID 3 algorithm decision tree is more accurate than a standard decision tree classifier of Matlab API based on GINI Index.

Keywords: Decision Tree, Data Mining , ID3 algorithm,Classification, GAIN,GINI.

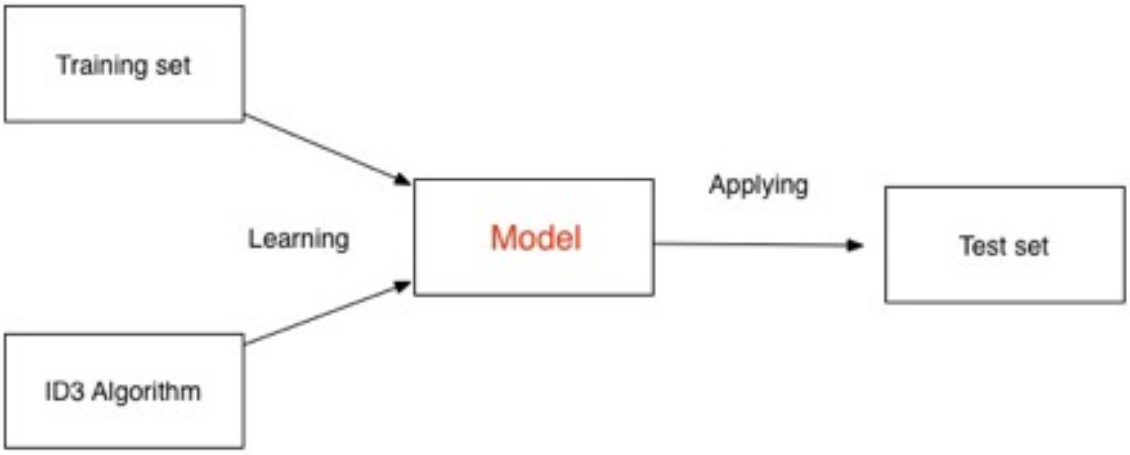
INTRODUCTION

Classification is based on a training set, to give a model for predicting class attribute as a function of the values of other attributes. The goal for classification is obviously that to predicting a class of previously unseen records as accurately as possible. Testing set is used to determine the accuracy of the model which based on the training set.

A decision tree is a very classic classifier that uses a tree structure of decisions' conditions, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal.

In my experiment, I download the Iris data from UCI Machining Learning Repository(*http://archive.ics.uci.edu/ml/index.html*) and divide it to training set (half mount of the data) and testing set, then modelling the decision tree based on training set data with ID3 algorithm. After builded, the model will be tested by the data from testing set, in order to get the accuracy of the model. Meanwhile, I built another tree with MATLAB default classification tree, train and test it with the same data set. The test results will be compared to demonstrate the accuracy of ID3 learning algorithm.

The process of my experiment is as follow:



ALGORITHM

The basic idea behind any decision tree algorithm is as follows:

•Choose the best attribute(s) to split the remaining instances and make that attribute a decision node.

•Repeat this process for recursively for each child. Stop when:

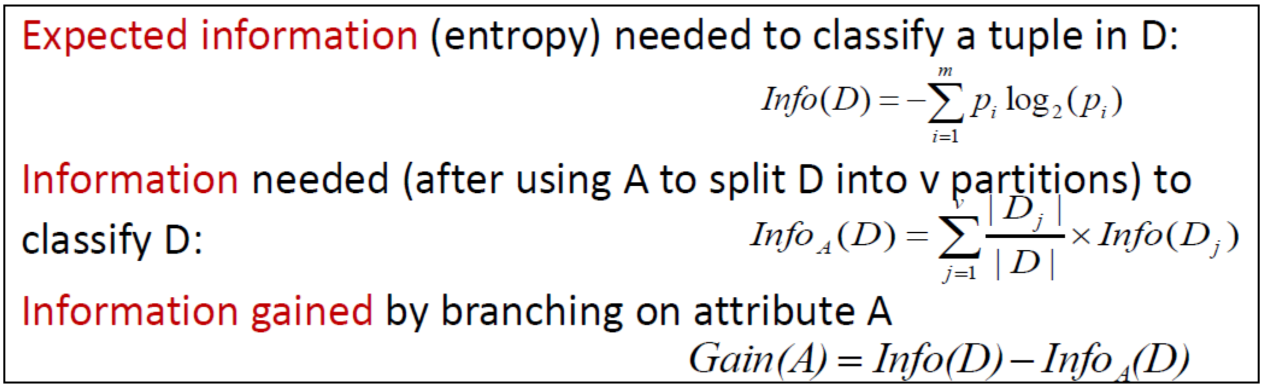
•All the instances have the same target attribute value.

•There are no more attributes.

•There are no more instances.

2.1 Information GAIN & Entropy

GAIN measures reduction in entropy achieved because of the split. In ID3 algorithm, we choose the split that achieves most reduction.(maximizes GAIN)



Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Information needed (after using A to split D into v partitions) to classify D:

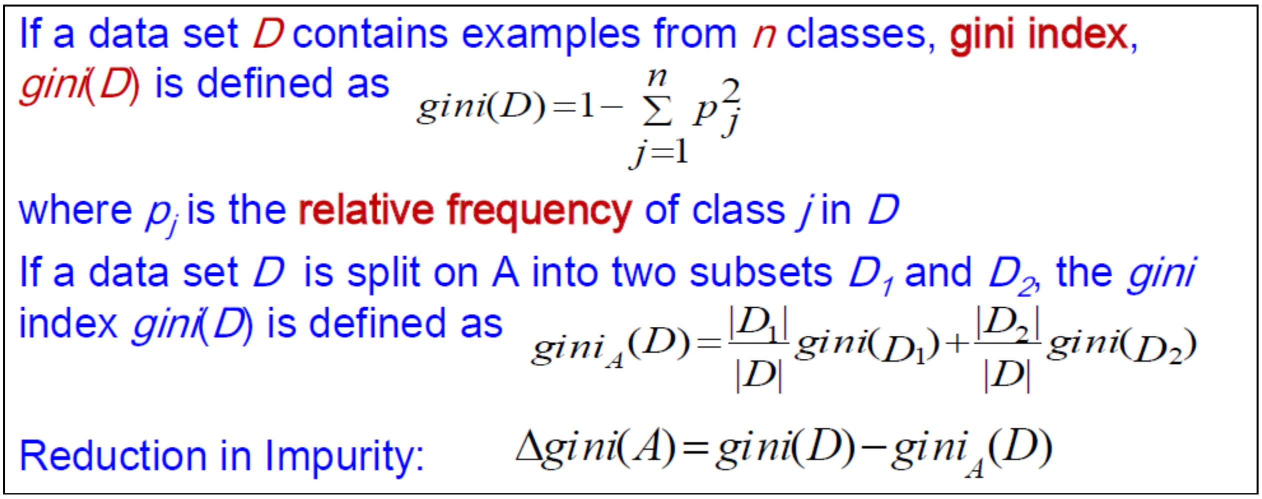
$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

figure 2.1

2.2 GINI Index



If a data set *D* contains examples from *n* classes, gini index, *gini*(*D*) is defined as

$$gini(D) = 1 - \sum_{j=1}^{n} p_j^2$$

where p_j is the relative frequency of class *j* in *D*

If a data set *D* is split on A into two subsets D_1 and D_2, the *gini* index *gini*(*D*) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

Reduction in Impurity: $\Delta gini(A) = gini(D) - gini_A(D)$

The classification tree building API of Matlab uses GINI to measure the reduction achieved for split.

figure 2.2

2.3 ID3 Algorithm

The ID3 algorithm begins with the original set S as the root node. On each iteration of the algorithm, it iterates through every unused attribute of the set S and calculates the information gain IG(A) of that attribute. It then selects the attribute which has the largest information gain value. The set S is then split by the selected attribute to produce subsets of the data.The tree is grown by recursively splitting each node using the feature which gives the best information gain until the leaf is consistent or all inputs have the same feature values.

The recursive process is split current node. For each split, we loop over each feature instead of considering only attributes never selected before to find the feature giving the largest information gain value. There are two ways splitting on the attribute, Multiway-split and Binary-split. Multi-way split means using as many partitions as distinct values (Figure 2.3 a)

Binary-split means dividing values into two subsets(Figure 2.3 b). In our algorithm, we choose binary-split, so we should determine the value to split on for each feature with following way:

```
vals = unique(feat);
splits = 0.5*(vals(1:end-1) + vals(2:end));
```

Then we calculate the IG(A). If it’s not the best information gain, we move to next feature until we find the largest IG(A). If it’s the best information gain, we split the current node into two nodes and move to the new nodes and recursively split them.

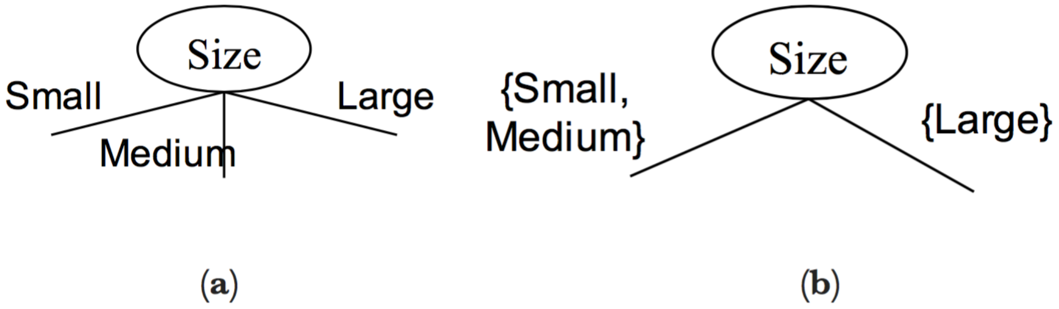


figure 2.3

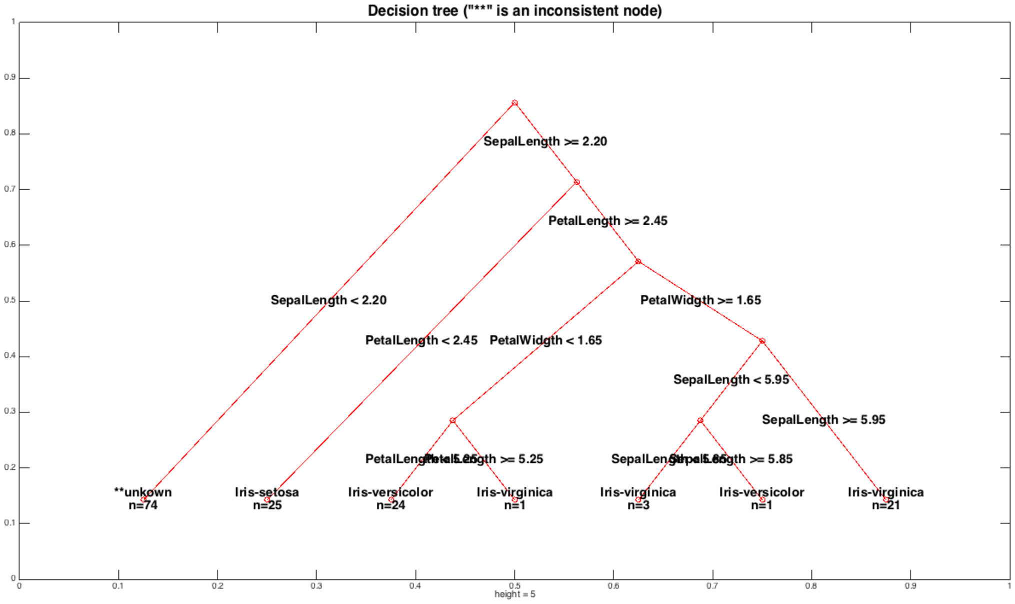
EXPERIEMENT

3.1 Structure of the Code

• build_tree.m	Build the decision tree based on ID3 algorithm
• cond_ent.m	Calculate the conditional entropy of y given x
• ent.m	Calculate the entropy of a vector of values
• SpiltDataToBuildTree.m	Import and format the data
• Predict.m	Testing the built tree to calculate the accuracy.
• dt_demo.m	Main script, the start of whole program.
• TranferToClassName.m	Transfer the result to real class name.

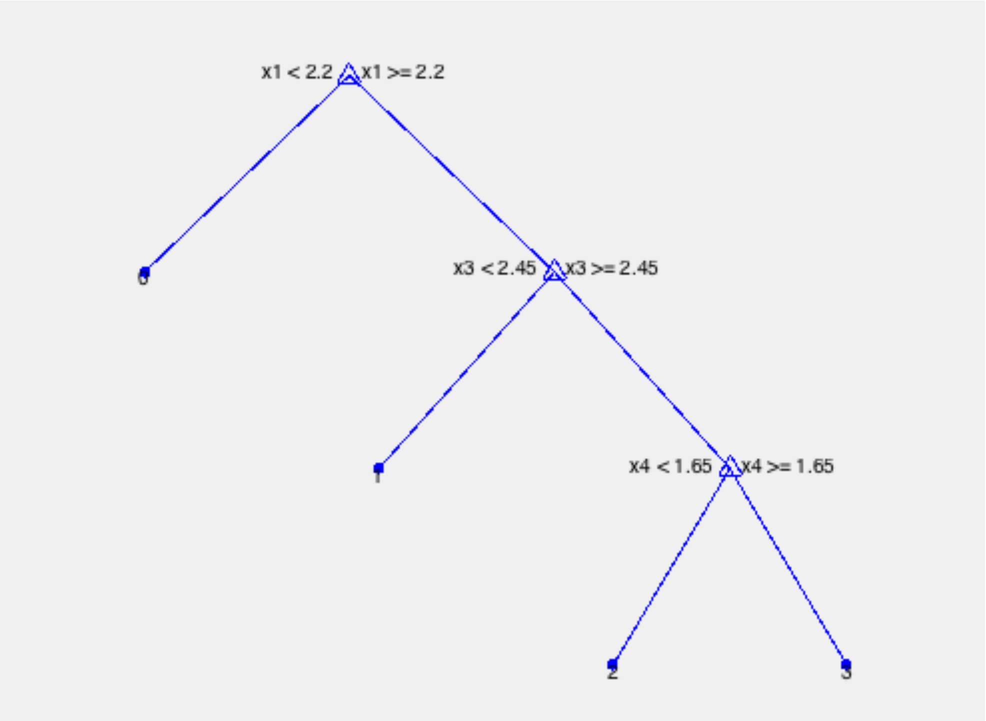
3.2 Result

The decision tree builded with ID3 Algorithm:



Testing set accuracy: 97.32%.

The decision tree modelling with Matlab Default algorithm:



Testing set accuracy: 93.33%.

CONCLUSION

In the experiment, I use the same data modelling two kind of trees, the first one using ID3 algorithm is based on Information GAIN and the other on GINI Index. The test result shows that for this condition, ID3 algorithm works better than GINI Index on accuracy.

After this experiment, I compare the difference of Information GAIN and GINI Index. In Information Gain Criteria, log-likelihood might give higher scores to balanced portions when there are too many classes though. GINI may be nicer because it doesn't have logarithms and you can find the closed form for its expected value and variance under random split assumption.

I think the matlab's default classifier is a perfect decision tree, it runs faster and can change accuracy if you needed.

BIBLIOGRAPHY:

[1] Wikipedia, Decision tree, https://en.wikipedia.org/wiki/Decision_tree

[2] J.R QUINLAN, Induction of Decision Trees, Machine Learning 1:81-106, 1986.

[3] Wikipedia, ID3 Algorithm, https://en.wikipedia.org/wiki/ID3_algorithm

[4] Yoav Freund, Loew Mason, The alternating decision tree learning algorithm, http://cseweb.ucsd.edu/~yfreund/papers/atrees.pdf

[5] UCI Machining Learning Repository. (http://archive.ics.uci.edu/ml/index.html)