Iterative Dichotomiser 3 (ID3)

1st Tomáš Vank
Katedra kybernetiky a umelej inteligencie
Technicka univerzita v Košiciach
Kosice, Slovenská republika
tomas.vank@student.tuke.sk

2nd Maroš Stredanský
Katedra kybernetiky a umelej inteligencie
Technicka univerzita v Košiciach
Kosice, Slovenská republika
maros.stredansky@student.tuke.sk

3nd Marek Tóth
Katedra kybernetiky
Technicka univerzita v Košiciach
Kosice, Slovenská republika
marek.toth.2@student.tuke.sk

***Abstract*—Machine learning is one of the ways in which we strive to achieve artificial intelligence. ML is the part of artificial intelligence that provides the system with the ability to learn and improve automatically, based on existing examples from the past or from personal experience. We consider a knowledge based system to be a computer program that will allow the user to reach a decision in a precisely defined problem area, which would be reached by an expert in the same situation.**

***Index Terms*—machine learning, iterative dichotomiser, knowledge based systems**

I. Introduction

Knowledge based systems form one of the important parts of artificial intelligence - they have their specific theoretical foundations and are important for their practical applicability. In our work we focused on the application of decision trees, namely the ID3 algorithm. In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan and it is used to generate a decision tree from a dataset. Iterative Dichotomiser 3 is the precursor to the C4.5 algorithm also developed by Ross Quinlan. The usage of ID3 is typically in the machine learning applications and natural language processing domains.

II. Algorithm

A. Scenario usage of algorithm

Decision Tree learning is used to approximate discrete valued target functions, in which the learned function is approximated by Decision Tree. To imagine, think of decision tree as if or else rules where each if-else condition leads to certain answer at the end. Decision tree should be used at any scenario where learning data has attribute value pair like in the example shown above: Wind as an attribute has two possible values, or where Target function has discreet output. Here, the target function is – should you play tennis? And the output to this discreet output (yes and no) or where training data might be missing or have error.

III. The principle of operation of ID3

Starting from the root of the tree, ID3 builds the decision tree one interior node at a time where at each node we select the attribute which provides the most information gain if we were to split the instances into subsets based on the values of that attribute. How is "most information" determined? It is determined using the idea of entropy reduction which is part of Shannon's Information Theory.

Entropy is a measure of the amount of disorder or uncertainty (units of entropy are bits). Entropy is sometimes described as a measure of randomness. In other words, its a measure of unpredictability.A data set with a lot of disorder or uncertainty does not provide us a lot of information. A good way to think about entropy is how certain we would feel if we were to guess the class of a random training instance. A data set in which there is only one class has 0 entropy (high information here because we know with 100 percent certainty what the class is given an instance).

A. Binary classification problem

In a binary classification problem with positive instances p (play tennis) and negative instances n (don't play tennis), the entropy contained in a data set is defined mathematically as follows (base 2 log is used as convention in Shannon's Information Theory).
Mathematical face of calculation of entropy:

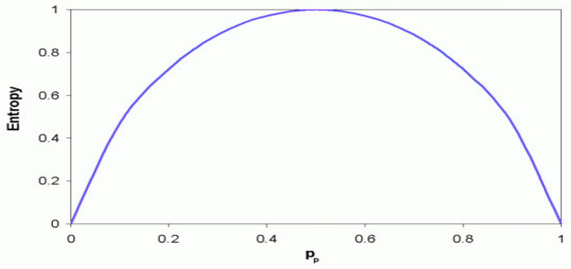
$$H(S) = -(prob(a)*log_2(prob(a)))-(prob(b)*log_2(prob(b))) \tag{1}$$

Where prob(a) is probability of getting head and prob(b) is probability of getting tail.

This mathematical formulae can be generalised for n discreet outcome as follow:

$$\mathrm{H}(S) = \sum_{x \in X} -p(x) \log_2 p(x) \tag{2}$$

Entropy or randomness is highest when chances of happening both the outcome are equal i.e. at p = 0.5. This gives us the following graph between entropy and probability.



Entropy is used in decision tree classifier in machine learning. Entropy is a interdisciplinary concept. It originated in thermodynamics and finds uses in evolutionary studies, information theory as well as quantum mechanics.

B. *Information Gain*

Information gain IG(A) is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set S on attribute A.

Mathematical face of calculation of information gain:

$$IG(S,A) = \mathrm{H}(S) - \sum_{t \in T} p(t)\mathrm{H}(t) = \mathrm{H}(S) - \mathrm{H}(S|A). \quad (3)$$

Where:

- H(S) – Entropy of set S
- T – The subsets created from splitting set S by attribute A
- p(t) – The proportion of the number of elements in t to the number of elements in set S
- H(t) – Entropy of subset t

The entropies of the partitions, when summed and weighted, can be compared to the entropy of the entire data set. The first term corresponds to the entropy of the data before the partitioning, whereas the second term corresponds to the entropy afterwards. We want to maximize information gain, so we want the entropies of the partitioned data to be as low as possible, which explains why attributes that exhibit high information gain split training data into relatively heterogeneous groups.

C. *Avoiding overfitting*

Because the ID3 algorithm continues splitting on attributes until either it classifies the data perfectly or there are no more attributes to split on, it's prone to creating decision trees that overfit by performing really well on the training data at the expense of accuracy with respect to the entire distribution of data.

There are two popular approaches to avoid this in decision trees: stop growing the tree before it becomes too large or prune the tree after it becomes too large. Typically, a limit to a decision tree's growth will be specified in terms of the maximum number of layers, or depth, it's allowed to have. The data available to train the decision tree will be split into a training set and test set and trees with various maximum depths will be created based on the training set and tested against the test set.

Crossvalidation can be used as part of this approach as well. Pruning the tree, on the other hand, involves testing the original tree against pruned versions of it. Leaf nodes are taken away from the tree as long as the pruned tree performs better against test data than the larger tree.

D. *Steps in ID3 algorithm*

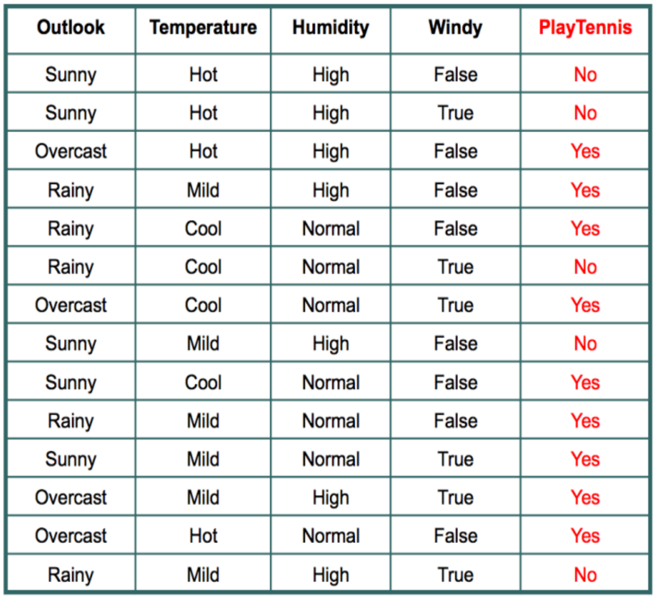
- 1. Calculate entropy for dataset.
- 2. For each attribute/feature calculate entropy for all its categorical values.
- 3. For each attribute/feature calculate information gain for the feature.
- 4. Find the feature with maximum information gain.
- 5. Repeat it until we get the desired tree.

The order of these steps as their number may vary depending on the complexity of solving the example.

IV. Practical Application of the Algorithm

A. *Chosen dataset*

In our application of the ID3 algorithm, we used the well-known Play Tennis data file, which is widely used for educational purposes by scientists and students from all over the world. The data contained in the dataset are structured, without error rows or incorrectly marked variables in individual columns. Each column represents one attribute, and each of these columns, for example Outlook, can take on several values - properties. The Play Tennis dataset we use discusses whether or not we have been playing tennis for the last few days. The dataset consists of fourteen lines where each line represents the day during which we were or were not playing tennis. Based on the attributes on each day, the classification is determined whether or not we played tennis. Based on the input dataset, the system should be able to generate a decision tree based on its attributes and properties that are contained in the dataset.



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Demonstration of the dataset used in the ID3 application.

B. *Calculating of entropy and information gain*

We have 14 total instances: 9 instances of prob(a) and 5 instances of prob(b). With the frequency counts of each unique class, we can calculate the prior entropy of this data set where:

prob(a) = number of positive instances (e.g. number of play tennis instances)

prob(b) = number of negative instances (e.g. number of don't play tennis instances)

In the section of information gain calculation and entropy calculation we mention the variables p and n, in our case we symbolize the variable p with prob (a) and the variable n with prob (b). Calculation of entropy we provided via above mentioned mathematical formula for calculation of entropy:

$$H(S) = -(prob(a)*log_2(prob(a)))–(prob(b)*log_2(prob(b))) \tag{4}$$

In the given 14 days, we played tennis on 9 occasions and we did not play on 5 occasions. Probability of playing tennis can be calculated as **Probability = (Number of favourable events) / (Number of total events)**. In our case number of favourable events is 9 and number of total events (number of days) is number 14.

$$prob(a) = (favourableevents)/(totalevents)$$

$$prob(a) = 9/14$$

$$prob(a) = 0.642$$

Our probability of playing tennis in those 14 days is 0.642.

Probability of not playing tennis can be calculated as **Probability = (Number of favourable events) / (Number of total events)**. In our case we did not play tennis on 5 occasions, our number of favourable events is then equal to number of occasions when we did not play tennis what is 5.

$$prob(b) = (favourableevents)/(totalevents)$$

$$prob(b) = 5/14$$

$$prob(b) = 0.357$$

Our probability of not playing tennis in those 14 days is 0.357.

Entropy of outcome can be calculated as we mentioned in subsection calculation of entropy and is provided via mathematical formula

$$H(S) = -(prob(a)*log_2(prob(a)))–(prob(b)*log_2(prob(b))) \tag{5}$$

Our entropy of source H(S) is calculated as

$$H(S) = -(prob(a)*log_2(prob(a)))–(prob(b)*log_2(prob(b)))$$

Where prob(a) is our probability of playing tennis during those 14 days (0.642) and prob(b) is our probability of not playing tennis during those 14 days (0.357).

$$H(S) = -0.652 * log_2(0.652)–0.357 * log_2(0.357)$$

$$H(S) = 0.940$$

H(S) symbolizes the entropy of the whole system.

Now, we have four features to make decision (Outlook,Temperature,Windy,Humidity)

1) Outlook: If we make a decision tree division at this level 0 based on outlook, we have three branches possible; either it will be Sunny or Overcast or it will be Raining.

Sunny : In the given data, 5 days were sunny. Among those 5 days, tennis was played on 2 days and tennis was not played on 3 days

$$prob(a) = 2/5$$

$$prob(a) = 0.4$$

$$prob(b) = 3/5$$

$$prob(b) = 0.6$$

H(S) = entropy when sunny

$$H(S) = -0.4 * log_2(0.4)–0.6 * log_2(0.6)$$

$$H(S) = 0.97$$

Overcast : In the given data, 4 days were overcast and tennis was played on all the four days.

$$prob(a) = 4/4$$

$$prob(a) = 1$$

$$prob(b) = 0/4$$

$$prob(b) = 0$$

H(S) = entropy when Overcast

$$H(S) = 0$$

Rain : In the given data, 5 days were rainy. Among those 5 days, tennis was played on 3 days and tennis was not played on 2 days.

$$prob(a) = 2/5$$

$$prob(a) = 0.4$$

$$prob(b) = 3/5$$

$$prob(b) = 0.6$$

H(S) = entropy when Rain

$$H(S) = -0.4 * log_2(0.4)–0.6 * log_2(0.6)$$

$$H(S) = 0.97$$

Entropy among the branches Overcast, Sunny and Rain

H(S) = Entropy among the branches H(S) = ((number of sunny days)/(total days) * (entropy when sunny)) + ((number of overcast days)/(total days) * (entropy when overcast)) + ((number of rainy days)/(total days) * (entropy when rainy))

$$H(S) = ((5/14) * 0.97) + ((4/14) * 0) + ((5/14) * 0.97)$$

$$H(S) = 0.69$$

Information Gain, sometimes called as reduction in randomness can by calculated as difference between entropy of whole system and entropy among the branches.

$$I(G) = 0.940–0.69$$

$$I(G) = 0.246$$

By doing similar calculates for other features we receive:

- I(G) for Temperature = 0.029
- I(G) for Windy = 0.048
- I(G) for Humidity = 0.152

We can see that decrease in randomness, or information gain is most for Outlook. So, we choose first decision maker as Outlook.

C. Pseudocode of ID3 Class

ID3 Pseudocode ''' examples are the training examples. attributes is a list of attributes that may be tested by the learned decison tree. Returns a tree that correctly classifies the given examples. Assume that the targetAttribute, which is the attribute whose value is to be predicted by the tree, is a class variable. '''

```
id3(examples, attributes):
node = DecisionTreeNode(examples)
''' handle target attributes with arbitrary labels '''
dictionary = dictionary = summarizeExamples(examples, targetAttribute)
for key in dictionary: do
    if dicionary[key] == total number of examples then node.label = key return node
        test for number of examples to avoid overfitting
        if attributes is empty or number of examples ¡ minimum allowed per branch then node.label = most common value in examples return node
            bestA = the attribute with the most information gain
            node.decision = bestA
            for each possible value v of BestA do subset = the subset of examples that have value v for bestA
                if subset is not empty then
                    node.addBranch(id3(subset,targetAttribute,attributes-bestA))
                    return node
```

Conclusion

Decision tree is a very simple model that you can build from starch easily. One of popular Decision Tree algorithm is ID3. Basically, we only need to construct tree data structure and implements two mathematical formula to build complete ID3 algorithm. Algorithm ID3 symbolizes a way to build the fastest and short decision trees. This algorithm can create understandable prediction rules from the training dataset. The ID3 algorithm has many advantages where one of them is definitelly that ID3 only needs to test enough attributes until all data is classified. The main disadvantage od ID3 is fact only one attribute at a time can be tested for making a decision for our final tree.

References

- Breiman,Friedman,Olshen,Stone: Classification and Decision Trees Wadsworth, 1984
- Quinlan,J.R.: C4.5: Programs for Machine Learning Morgan Kauffman, 1993