Contriving Decision Trees with Iterative Dichotomiser Algorithm(ID3)

Sreevani Tapila , Dharamvir
[1,2] Assistant professor
[1,2] Department of Master of computer Applications
[1,2] The Oxford College of Engineering,
Bangalore-560068

Abstract: Decision tree learning is one of the predictive modelling approaches used in statistics, data mining and machine learning. It uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves)**.**In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan[1] used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

Keywords: ***Decision trees,ID3,Entropy,Information Gain,C4.5 sucessor of ID3,Gain of Attributes.***

1. Introduction

A decision tree is a simple representation for classifying examples. For this section, assume that all of the input data have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a *class*. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes, signifying that the data set has been classified by the tree into either a specific class, or into a particular probability distribution.

In decision tree learning, ID3 (Iterative Dichotomiser 3) is an algorithm invented by Ross Quinlan[1] used to generate a decision tree from a dataset. ID3 is the precursor to the C4.5 algorithm, and is typically used in the machine learning and natural language processing domains.

1.1. ID3 Algorithm

```
ID3 (Examples, Target_Attribute, Attributes)
    Create a root node for the tree
    If all examples are positive, Return the single-node tree Root, with label = +.
    If all examples are negative, Return the single-node tree Root, with label = -.
    If number of predicting attributes is empty, then Return the single node tree Root,
    with label = most common value of the target attribute in the examples.
    Otherwise Begin
        A ← The Attribute that best classifies examples.
        Decision Tree attribute for Root = A.
        For each possible value, $v_i$, of A,
            Add a new tree branch below Root, corresponding to the test A = $v_i$.
            Let Examples($v_i$) be the subset of examples that have the value $v_i$ for A
            If Examples($v_i$) is empty
                Then below this new branch add a leaf node with label = most common target value in the examples
            Else below this new branch add the subtree ID3 (Examples($v_i$), Target_Attribute, Attributes – {A})
    End
    Return Root
```

1.2 Metrics

1..2.1 Formulae

Entropy is defined as below

$$\mathrm{H}(T) = \mathrm{I}_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

where are fractions that add up to 1 and represent the percentage of each class present in the child node that results from a split in the tree.[20]

$$\overbrace{IG(T,a)}^{\text{Information Gain}} - \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Sum of Entropy (Children)}}$$
$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{i=1}^{J} -\Pr(i|a)\log_2 \Pr(i|a)$$

Averaging over the possible values of ,

$$\overbrace{E_A(IG(T,a))}^{\text{Expected Information Gain}} = \overbrace{I(T;A)}^{\text{Mutual Information between T and A}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|A)}^{\text{Weighted Sum of Entropy (Children)}}$$
$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_{a} p(a) \sum_{i=1}^{J} -\Pr(i|a)\log_2 \Pr(i|a)$$

1.2.2 Dataset description

In this article, we'll be using a sample dataset of COVID-19 infection. A preview of the entire dataset is shown below.

ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
10	YES	YES	NO	YES
11	NO	YES	NO	NO
12	NO	YES	YES	YES
13	NO	YES	YES	NO
14	YES	YES	NO	NO

The columns are self-explanatory. Y and N stand for Yes and No respectively. The values or **classes** in Infected column Y and N represent Infected and Not Infected respectively.The columns used to make decision nodes viz. **'Breathing Issues', 'Cough'** and **'Fever'** are called feature columns or just features and the column used for leaf nodes i.e. **'Infected'** is called the target column.

1.2.3 Metrics in ID3

Before you ask, the answer to the question: 'How does ID3 select the best feature?' is that ID3 uses **Information Gain** or just **Gain** to find the best feature.Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes. The feature with the **highest Information Gain** is selected as the **best** one.In simple words, **Entropy** is the measure of disorder and the Entropy of a dataset is the measure of disorder in the target feature of the dataset.

In the case of binary classification (where the target column has only two types of classes) entropy is **0** if all values in the target column are homogenous(similar) and will be **1** if the target column has equal number values for both the classes.

We denote our dataset as **S,** entropy is calculated as:

Entropy(S) = - $\sum p_i$ * log□(p_i) ; i = 1 to n

Where,**n**,is the total number of classes in the target column(in our case n=2 i.e YES and NO)$\mathbf{p_i}$ is the probability of class **'i'** or the ratio of "number of rows with class I in the target column" to the 'total numbe of rows" in the dataset.

Information Gain for a feature column **A** is calculated as:

IG(S, A) = Entropy(S) - ∑((|S_v| / |S|) * Entropy(S_v))

where S_v is the set of rows in **S** for which the feature column **A** has value **v**, **|S_v|** is the number of rows in S_v and likewise **|S|** is the number of rows in **S.**

1.2.4 ID3 Steps

- Calculate the Information Gain of each feature.
- Considering that all rows don't belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
- Make a decision tree node using the feature with the maximum Information gain.
- If all rows belong to the same class, make the current node as a leaf node with the class as its label.
- Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

2.Implementation on our Dataset

we first need to calculate the entropy of **S.**From the total of 14 rows in our dataset **S**, there are **8** rows with the target value **YES** and **6** rows with the target value **NO**. The entropy of **S** is calculated as:

Entropy(S) = — (8/14) * log□(8/14) — (6/14) * log□(6/14) = 0.99

Note: If all the values in our target column are same the entropy will be zero (meaning that it has no or zero randomness).

We now calculate the Information Gain for each feature:

2.1 IG calculation for Fever:

In this(Fever) feature there are 8 rows having value YES and 6 rows having value NO.

As shown below, in the 8 rows with YES for Fever, there are 6 rows having target value YES and 2 rows having target value NO.

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	YES	NO	NO
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
YES	YES	NO	YES
YES	YES	NO	NO

As shown below,in the 6 rows with **No**,there are 2 rows having target value **YES** and 4 rows having target value **NO**

Fever	Cough	Breathing issues	Infected
NO	NO	NO	NO
NO	YES	NO	NO
NO	YES	YES	YES
NO	YES	NO	NO
NO	YES	YES	YES
NO	YES	YES	NO

The block,below,demonstrate the calculation of Information Gain for **Fever**

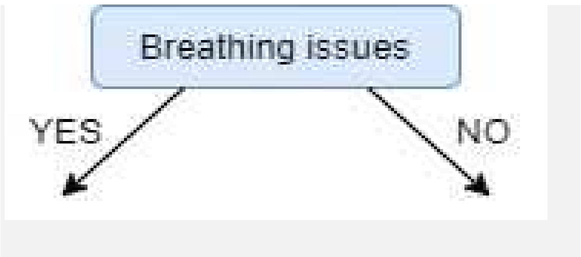
# total rows
|S| = 14For v = YES, |S_v| = 8
Entropy(S_v) = - (6/8) * log□(6/8) - (2/8) * log□(2/8) = 0.81For v = NO, |S_v| = 6
Entropy(S_v) = - (2/6) * log□(2/6) - (4/6) * log□(4/6) = 0.91
# Expanding the summation in the IG formula:
IG(S, Fever) = Entropy(S) - (|SYE□| / |S|) * Entropy(SYE□) -
(|SNO| / |S|) * Entropy(SNO).∵IG(S, Fever) = 0.99 - (8/14) * 0.81 - (6/14) * 0.91 = 0.13

Next,we calculate the IG for the features **"Cough"** and **"Breathing issues".**

IG(S,Cough)=0.04
IG(S,Breathing Issues)=0.40

Since the feature Breathing issues have the highest Information Gain it is used to create the root node.

Hence,after this intial step our tree looks like this:



Next, from the remaining two unused features, namely, **Fever** and **Cough**, we decide which one is the best for the left branch of **Breathing Issues**.

Since the left branch of **Breathing Issues** denotes **YES,** we will work with the subset of the original data i.e the set of rows having **YES** as the value in the Breathing Issues column. These **8 rows** are shown below:

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO

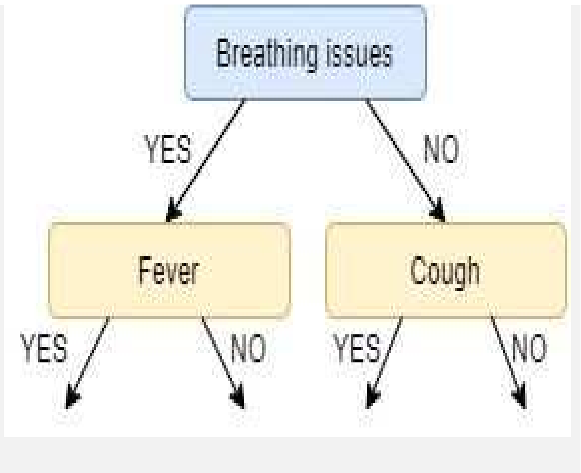
Next,we calculate the IG for the features Fever and Cough using the subset **SBY** (Set Breathing Issues Yes) which is shown above:note for **IG** calculated from the subset **SBY** and not the original dataset **S.**

IG(SBY ,Fever)=0.20

IG(SBY ,Cough)=0.09

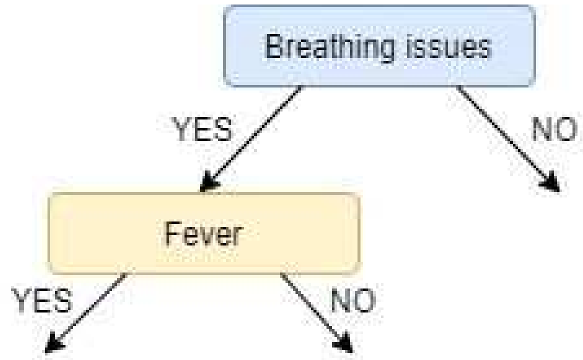
IG of Fever is greater than that of Cough,so we select **Fever** as the left branch of Breathing Issues:

Our tree now looks like this:



Next, we find the feature with the maximum IG for the right branch of **Breathing Issues**. But, since there is only one unused feature left we have no other choice but to make it the right branch of the root node.

So our tree now looks like this:



There are no more unused features, so we stop here and jump to the final step of creating the leaf nodes.

For the left leaf node of Fever, we see the subset of rows from the original data set that has

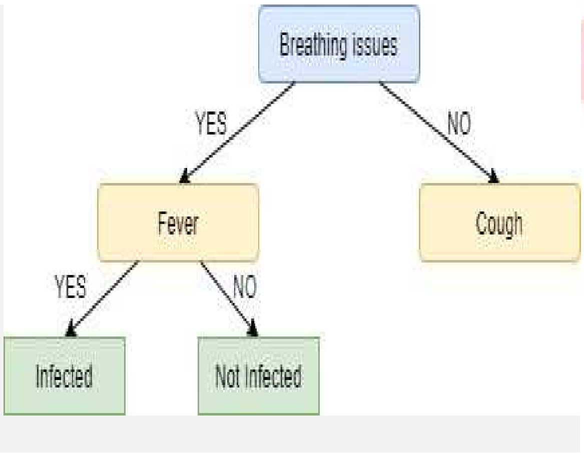
Breathing Issues and **Fever** both values as **YES**.

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES

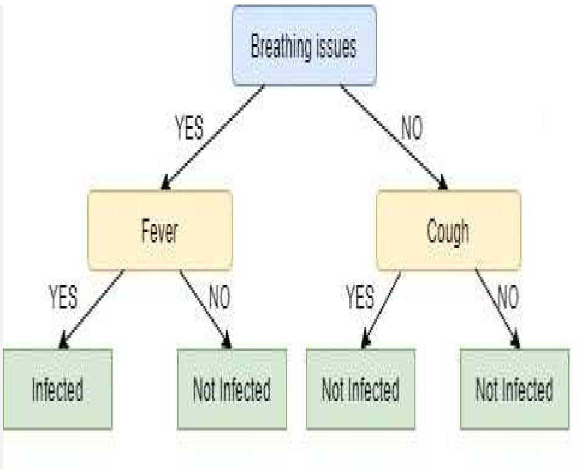
Since all the values in the target column are **YES,** we label the left leaf node as **YES**, but to make it more logical we label it **Infected.**Similarly, for the right node of Fever we see the subset of rows from the original data set that have **Breathing Issues** value as **YES** and **Fever** as **NO**.

Fever	Cough	Breathing issues	Infected
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO

Here not all but **most** of the **values** are **NO**,hence **NO** or **Not infected** becomes our **right leaf node.** Our tree now,look like this:



We repeat the same process for the node **Cough**, however here both left and right leaves turn out to be the same i.e. **NO** or **Not Infected** as shown below:



Looks Strange, doesn't it?
I know! The right node of Breathing issues is as good as just a leaf node with class 'Not infected'. This is one of the Drawbacks of ID3, it doesn't do pruning.

3.Conclusion

We covered the process of the ID3 algorithm in detail and saw how easy it was to create a Decision Tree using this algorithm by using only two metrics viz. Entropy and Information Gain .

4.Successor of ID3 (C4.5 Algorithm)

C4.5 is an algorithm used to generate a decision tree developed by Ross Quinlan.[1] C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. In 2011, authors of the Weka machine learning software described the C4.5 algorithm as "a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date".[2]

5.References

[1] Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 2016), 81–106

[2] Mitchell, Tom Michael (1997). Machine Learning. New York, NY: McGraw-Hill. pp. 55–58. ISBN 0070428077. OCLC 36417892.

[3] Grzymala-Busse, Jerzy W. (February 1993). "Selected Algorithms of Machine Learning from Examples" (PDF). Fundamenta Informaticae. **18** (2): 193–207 – via ResearchGate.

[4] Quinlan, J. R. (1986). "Induction of decision trees" (PDF). Machine Learning. **1**: 81–106. doi:10.1007/BF00116251. S2CID 189902138.

[5]Salzberg1994_Article_C45ProgramsForMachineLearningB.*pdf*