A comparative study of decision tree ID3 and C4.5

Badr HSSINA, Abdelkarim MERBOUHA,Hanane EZZIKOURI,Mohammed ERRITALI

TIAD laboratory, Computer Sciences Department, Faculty of sciences and techniques

Sultan Moulay Slimane University

Beni-Mellal, BP: 523, Morocco

Abstract—Data mining is the useful tool to discovering the knowledge from large data. Different methods & algorithms are available in data mining. Classification is most common method used for finding the mine rule from the large database. Decision tree method generally used for the Classification, because it is the simple hierarchical structure for the user understanding & decision making. Various data mining algorithms available for classification based on Artificial Neural Network, Nearest Neighbour Rule & Baysen classifiers but decision tree mining is simple one. ID3 and C4.5 algorithms have been introduced by J.R Quinlan which produce reasonable decision trees. The objective of this paper is to present these algorithms. At first we present the classical algorithm that is ID3, then highlights of this study we will discuss in more detail C4.5 this one is a natural extension of the ID3 algorithm. And we will make a comparison between these two algorithms and others algorithms such as C5.0 and CART.

Keywords—*Data mining; classification algorithm; decision tree; ID3 algorithme; C4.5 algorithme*

I. INTRODUCTION

The construction of decision trees from data is a longstanding discipline. Statisticians attribute the paternity to Sonquist and Morgan (1963) [4] who used regression trees in the process of prediction and explanation (AID - Automatic Interaction Detection). It was followed by a whole family of method, extended to the problems of discrimination and classification, which were based on the same paradigm of representation trees (Thaid - Morgan and Messenger, 1973; CHAID - Kass, 1980). It is generally considered that this approach has culminated in the CART (Classification and Regression Tree) method of Breiman et al. (1984) described in detail in a monograph refers today. [4]

In machine learning, most studies are based on information theory. It is customary to quote the ID3 Quinlan method (Induction of Decision Tree - Quinlan 1979), which itself relates his work to that of Hunt (1962) [4]. Quinlan has been a very active player in the second half of the 80s with a large number of publications in which he proposes a heuristics to improve the behavior of the system. His approach has made a significant turning point in the 90s when he presented the C4.5 method which is the other essential reference when we want to include decision trees (1993). There are many other changes this algorithm, C5.0, but is implemented in a commercial software.

Classification methods aim to identify the classes that belong objects from some descriptive traits. They find utility in a wide range of human activities and particularly in automated decision making.

Decision trees are a very effective method of supervised learning. It aims is the partition of a dataset into groups as homogeneous as possible in terms of the variable to be predicted. It takes as input a set of classified data, and outputs a tree that resembles to an orientation diagram where each end node (leaf) is a decision (a class) and each non- final node (internal) represents a test. Each leaf represents the decision of belonging to a class of data verifying all tests path from the root to the leaf.

The tree is simpler, and technically it seems easy to use. In fact, it is more interesting to get a tree that is adapted to the probabilities of variables to be tested. Mostly balanced tree will be a good result. If a sub-tree can only lead to a unique solution, then all sub-tree can be reduced to the simple conclusion, this simplifies the process and does not change the final result. Ross Quinlan worked on this kind of decision trees.

II. INFORMATION THEORY

Theories of Shannon is at the base of the ID3 algorithm and thus C4.5. Entropy Shannon is the best known and most applied. It first defines the amount of information provided by an event: the higher the probability of an event is low (it is rare), the more information it provides is great. [2] (In the following all logarithms are base2).

A. Shannon Entropy

In general, if we are given a probability distribution P = (p_1, p_2,…, p_n) and a sample **S** then the **Information** carried by this distribution, also called the **entropy of P** is giving by:

$$Entropie(P) = -\sum_{i=1}^{n} p_i \times log(p_{i)} \quad (1)$$

B. The gain information G (p, T)

We have functions that allow us to measure the degree of mixing of classes for all sample and therefore any position of the tree in construction. It remains to define a function to select the test that must label the current node.

It defines the gain for a test **T** and a position **p**

$$Gain(p,T) = Entropie(p) - \sum_{j=1}^{n} (p_j \times Entropie(p_j)) \quad (2)$$

where values (p_j) is the set of all possible values for attribute T. We can use this measure to rank attributes and build the decision tree where at each node is located the

attribute with the highest information gain among the attributes not yet considered in the path from the root.

III. ID3 ALGORITHM

J. Ross Quinlan originally developed ID3 (Iterative DiChaudomiser 3) [21] at the University of Sydney. He first presented ID3 in 1975 in a book, Machine Learning [21], vol. 1, no. 1. ID3 is based off the Concept Learning System (CLS) algorithm. The basic CLS algorithm over a set of training instances C. ID3 is a supervised learning algorithm, [10] builds a decision tree from a fixed set of examples. The resulting tree is used to classify future samples. ID3 algorithm builds tree based on the information (information gain) obtained from the training instances and then uses the same to classify the test data. ID3 algorithm generally uses nominal attributes for classification with no missing values. [10]

The pseudo code of this algorithm is very simple. Given a set of attributes not target C_1, C_2, ..., C_n, C the target attribute, and a set S of recording learning. [7]

Inputs: R: a set of non- target attributes, C: the target attribute, S: training data.
Output: returns a decision tree
Start
Initialize to empty tree;
    If S is empty then
        Return a single node failure value
    End If
    If S is made only for the values of the same target
then
        Return a single node of this value
    End if
    If R is empty then
        Return a single node with value as the most common value of the target attribute values found in S
    End if
D ← the attribute that has the largest Gain (D, S) among all the attributes of R
{d_j j = 1, 2, ..., m} ← Attribute values of D
{S_j with j = 1, 2, ..., m} ← The subsets of S respectively constituted of d_j records attribute value D
    Return a tree whose root is D and the arcs are labeled by d1, d2, ..., d_m and going to sub-trees ID3 (R-{D}, C, S1), ID3 (R-{D} C, S2), .., ID3 (R-{D}, C, Sm)
End

Fig. 1. Pseudocode of ID3 algorithm

EXAMPLE 1

Suppose we want to use the ID3 algorithm to decide if the time ready to play ball.
During two weeks, the data are collected to help build an ID3 decision tree (Table 1).

The classification of the target is "should we play ball?" which can be Yes or No.
Weather attributes outlook, temperature, humidity and wind speed. They can take the following values:
    Outlook = {Sun, Overcast, Rain}
    Temperature = {Hot, Sweet, Cold}
    Humidity = {High, Normal}
    Wind = {Low, High}
Examples of the set S are:

TABLE I.   DATA SET S

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sun	Hot	High	Low	No
D2	Sun	Hot	High	High	No
D3	Overcast	Hot	High	Low	Yes
D4	Rain	Sweet	High	Low	Yes
D5	Rain	Cold	Normal	Low	Yes
D6	Rain	Cold	Normal	High	No
D7	Overcast	Cold	Normal	High	Yes
D8	Sun	Sweet	High	Low	No
D9	Sun	Cold	Normal	Low	Yes
D10	Rain	Sweet	Normal	Low	Yes
D11	Sun	Sweet	Normal	High	Yes
D12	Overcast	Sweet	High	High	Yes
D13	Overcast	Hot	Normal	Low	Yes
D14	Rain	Sweet	High	High	No

We need to find the attribute that will be the root node in our decision tree. The gain is calculated for the four attributes.

The entropy of the set S:

Entropy (S) = $-9/14*\log_2 (9/14)-5/14*\log_2 (5/14) = 0.94$

Calculation for the first attribute
Gain(S, Outlook) = Entropy (S)-5/14*Entropy (S_{Sun})
        -4/14*Entropy (S_{Rain})
        -5/14* Entropy ($S_{Overcast}$)
    =0.94 – 5/14*0.9710-4/14*0 – 5/14*0.9710
    Gain(S, Outlook) = 0 .246
Calculation of entropies:
Entropy (S_{Sunl}) = $-2/5*\log_2 (2/5)-3/5* \log_2 (3/5) = 0.9710$
Entropy (S_{Rain}) = $-4/4*\log_2 (4/4)-0* \log_2 (0) =0$
Entropy ($S_{overcast}$) = $-3/5* \log_2 (3/5) -5/5* \log_2 (2/5) =0.9710$
    As well we find for the other variables:
    Gain(S, Wind) = 0.048
    Gain(S, Temperature) = 0.0289
    Gain(S, Humidity) = 0.1515
Outlook attribute has the highest gain, so it is used as a decision attribute in the root node of the tree (Figure 2).

Since Visibility has three possible values, the root node has three branches (Sun, Rain and Overcast).
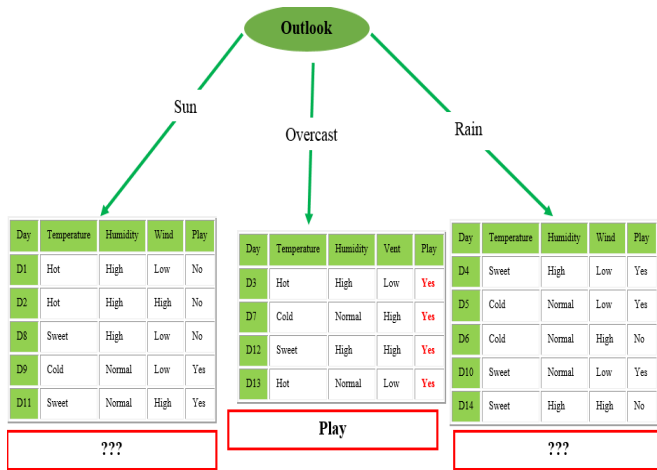
Fig. 2. Root node of the ID3 decision tree

So by using the three new sets, the information gain is calculated for the temperature, humidity, until we obtain subsets Sample containing (almost) all belonging examples to the same class (Figure 3).
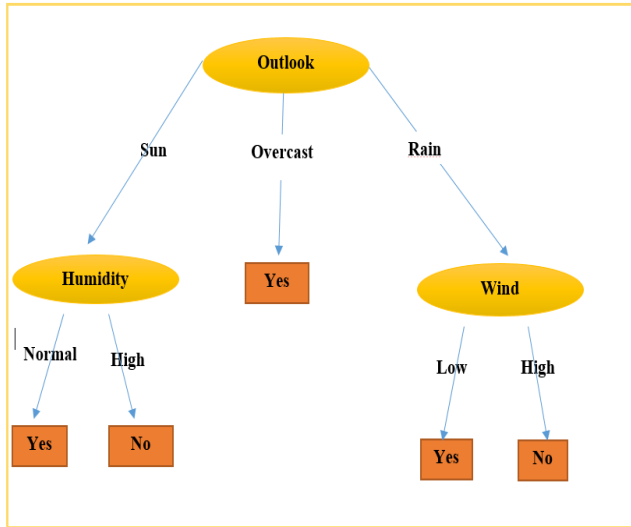

Fig. 3. ID3 Final tree

IV. C4.5 ALGORITHME

This algorithm was proposed in 1993, again by Ross Quinlan [28], to overcome the limitations of ID3 algorithm discussed earlier.

One limitation of ID3 is that it is overly sensitive to features with large numbers of values. This must be overcome if you are going to use ID3 as an Internet search agent. I address this difficulty by borrowing from the C4.5 algorithm, an ID3 extension. ID3's sensitivity to features with large numbers of values is illustrated by Social Security numbers. Since Social Security numbers are unique for every individual, testing on its value will always yield low conditional entropy values. However, this is not a useful test. To overcome this problem, C4.5 uses "Information gain," This computation does not, in itself, produce anything new. However, it allows to measure a gain ratio.

Gain ratio, is defined as follows:

$$GainRatio(p,T) = \frac{Gain(p,T)}{SplitInfo(p,T)} \qquad (3)$$

where **SplitInfo** is:

$$SplitInfo(p, test) = -\sum_{j=1}^{n} P'\left(\frac{j}{p}\right) \times log\left(P'\left(\frac{j}{p}\right)\right) \qquad (4)$$

P' (j/p) is the proportion of elements present at the position p, taking the value of j-th test. Note that, unlike the entropy, the foregoing definition is independent of the distribution of examples inside the different classes.

Like ID3 the data is sorted at every node of the tree in order to determine the best splitting attribute. It uses gain ratio impurity method to evaluate the splitting attribute (Quinlan, 1993). [10] Decision trees are built in C4.5 by using a set of training data or data sets as in ID3. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision.

A. Attributes of unknown value

During the construction of the decision tree, it is possible to manage data for which some attributes have an unknown value by evaluating the gain or the gain ratio for such an attribute considering only the records for which this attribute is defined. [2]

Using a decision tree, it is possible to classify the records that have unknown values by estimating the probabilities of different outcomes.

The new criterion gain will be of the form:

$$\textbf{\textit{Gain (p) = F (Info (T) - Info (p, T))}} \qquad (5)$$

where :

$$\textbf{Info (p, T)} = \sum_{j=1}^{n}(p_j \times Entropie(p_j)) \qquad (6)$$

Info (T) = Entropy (T)

F = number of samples in the database with the known value for a given / total number of samples in a set of attribute data.

B. Attributes value on continuous interval

C4.5 also manages the cases of attributes with values in continuous intervals as follows. Let us say that Ci attribute a continuous interval of values. Examines the values of this attribute in the training data. Let that these values are in ascending order, A_1, A_2, ..., A_m. Then for each of these values, the partitioned between records those that have values of C, less than or equal to A_j and those which have a value larger then A_j values. For each of these partitions gain is calculated,

or the gain ratio and the partition that maximizes the gain is selected.

C. Pruning

Generating a decision to function best with a given of training data set often creates a tree that over-fits the data and is too sensitive on the sample noise. Such decision trees do not perform well with new unseen samples.

We need to prune the tree in such a way to reduce the prediction error rate. Pruning [5] is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. The dual goal of pruning is the reduction complexity of the final classifier as well as better predictive accuracy by the reduction of over-fitting and removal of sections of a classifier that may be based on noisy or erroneous data.

The pruning algorithm is based on a pessimistic estimate of the error rate associated with a set of N cases, E of which do not belong to the most frequent class. Instead of E/N, C4.5 determines the upper limit of the binomial probability when E events have been observed in N trials, using a user-specified confidence whose default value is 0.25.

Pruning is carried out from the leaves to the root. The estimated error at a leaf with N cases and E errors is N times the pessimistic error rate as above. For a sub-tree, C4.5 adds the estimated errors of the branches and compares this to the estimated error if the sub-tree is replaced by a leaf; if the latter is no higher than the former, the sub-tree is pruned.

D. Exemple 2

We will work with the same example used before but this time we will take continuous values for humidity attribute.

TABLE II.   DATA SET S

Day	Outlook	Temperature	Humidity	Wind	Play
D1	Sun	Hot	85	Low	No
D2	Sun	Hot	90	High	No
D3	Overcast	Hot	78	Low	Yes
D4	Rain	Sweet	96	Low	Yes
D5	Rain	Cold	80	Low	Yes
D6	Rain	Cold	70	High	No
D7	Overcast	Cold	65	High	Yes
D8	Sun	Sweet	95	Low	No
D9	Sun	Cold	70	Low	Yes
D10	Rain	Sweet	80	Low	Yes
D11	Sun	Sweet	70	High	Yes
D12	Overcast	Sweet	90	High	Yes
D13	Overcast	Hot	75	Low	Yes
D14	Rain	Sweet	80	High	No

⊕ **Treating numerical values**

As C4.5 is an improvement of ID3, then the first step of calculating the gain is the same except for the attributes to continuous values.

In this example we are going to detail the calculation of information gain for an attribute of continuing value.

Gain (S, Humidity) =?
We must now sort the attribute values in ascending order, the set of values is as follows:
{65, 70, 70, 70, 75, 78, 80, 80, 80, 85, 90, 90, 95, 96}

we will remove values that are repeated:

{65, 70, 75, 78, 80, 85, 90, 95, 96}

TABLE III.   GAIN CALCULATION FOR THE ATTRIBUTE CONTINUOUS HUMIDITY USING C4.5 ALGORITHM

	65		70		75		78		80		85		90		95		96	
interval	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>	≤	>
Yes	1	8	3	6	4	5	5	4	7	2	7	2	8	1	8	1	9	0
No	0	5	1	4	1	4	1	4	2	3	3	2	4	1	5	0	5	0
Entropy	0	0.961	0.811	0.971	0.721	0.991	0.65	1	0.764	0.971	0.881	1	0.918	1	0.961	0	0.94	0
Info(S,T)	0.892		0.925		0.8950		0.85		0.838		0.915		0.929		0.892		0.94	
Gain	0.048		0.015		0.045		0.09		0.102		0.025		0.011		0.048		0	

Gain (S, Humidity) = 0.102

Then assigns Visibility has the largest value of Information Gain is the root node of the tree (Figure 4).
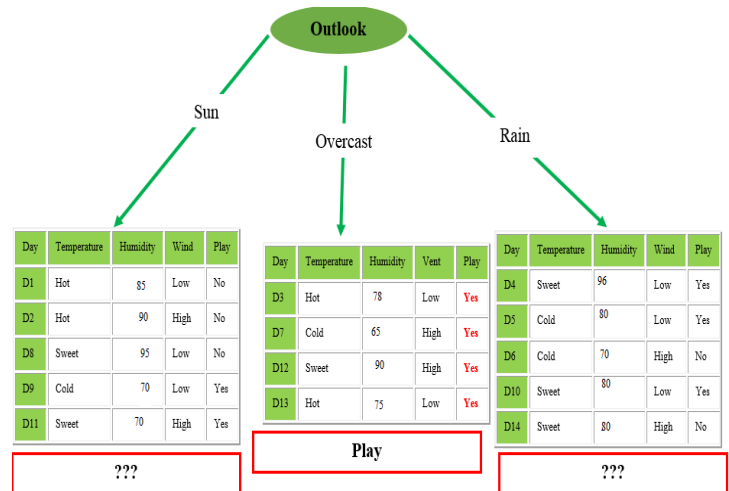

Fig. 4. Root node of the C4.5 decision tree

⊕ **Treating attributed to unknown value**

C4.5 accepted principle that a sample with unknown values are distributed based on the probability relative frequency of known values (Table 2).

Suppose the unknown value of D12 day for visibility attributes.

Info(S)= -8/13*$\log_2$ (8/13)-5/13* $\log_2$ (5/13)= 0.961
Info (Outlook, S) = 5/13*Entropy (S_{Sun})
$\qquad$ + 3/13* Entropy($S_{overcast}$)
$\qquad$ + 5/13* Entropy(S_{Rain})
$\qquad\qquad$ = 0.747

Entropy (S_{Sun}) =-2/5* $\log_2$ (2/5) –3/5* $\log_2$ (3/5)= 0.9710
Entropy $(S_{Overcast})$ =-3/3*$\log_2$ (3/3) –0/3* $\log_2$ (0/3)=0
Entropy (S_{Rain}) =-3/5* $\log_2$ (3/5) –2/5* $\log_2$ (2/5)= 0.9710
Gain (Outlook) = 13/14 (0.961 – 0.747) = 0.199

When a case of S with the known value is assigned to the subsets Si, the probability belonging to Si is 1, and in all other subsets is 0.

C4.5 therefore associated with each sample (with missing values) in each subset Si weight w representing the probability that the case belongs to each subset (Figure 5).

Fractionation of the set S using the test on the attribute visibility. A new w_i weight is equal to the probability in this case: 5/13, 3/13 and 5/13, because the initial value (Table 2) w is $|S1|$ = 5+5/13, $|S2|$ = 3 +3/13, and $|S3|$ = 5+5/13.

⊕ **Generating decision rules**

To make a clearer decision tree model, a path of each leaf can be converted into a production rule IF-THEN.

If Outlook= Sun then

    If Humidity <= 70 Then

    Classification = Yes (2.0 / 0);

     else

    Classification = No (3.38 / 0.6);

Else if Outlook = Overcast

    Classification = Yes (3.2 / 0);

Else if Outlook= Rain then

    If Wind =High

    Classification = Not (2.0 / 0);

else

Classification = Yes (3.38 / 0).

Fig. 5.   Decision rules

V.   COMPARISON BETWEEN SEVERAL ALGORITHMS

A. *ID3 Vs C4.5*

ID3 algorithm selects the best attribute based on the concept of entropy and information gain for developing the tree.

C4.5 algorithm acts similar to ID3 but improves a few of ID3 behaviors:

⊕ A possibility to use continuous data.
⊕ Using unknown (missing) values
⊕ Ability to use attributes with different weights.
⊕ Pruning the tree after being created.
    • Pessimistic prediction error
    • sub-tree Raising

Performance Parameters:

Accuracy: The measurements of a quantity to that quantity's factual value to the degree of familiarity are known as accuracy.

The Table 4 presents a comparison of ID3 and C4.5 accuracy with different data set size, this comparison is presented graphically in Figure 6.

TABLE IV.   ACCURACY COMPARISON BETWEEN ID3 AND C4.5 ALGORITHM

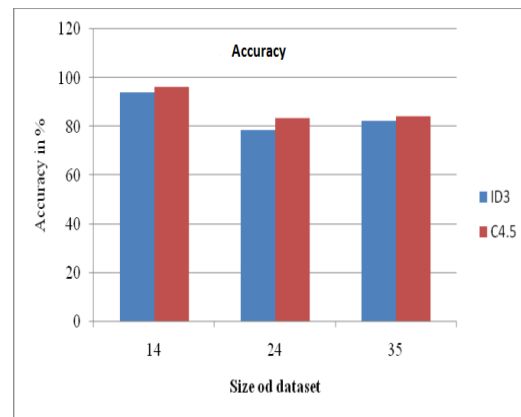| Size of Data Set	Algorithm	
	ID3 (%)	C4.5 (%)
14	94.15	96.2
24	78.47	83.52
35	82.2	84.12

Fig. 6.   Comparison of Accuracy for ID3 & C4.5 Algorithm

The 2nd parameter compared between ID3 and C4.5 is the execution time, Table 5 present the comparison.

This comparison is presented graphically in Figure 7.

TABLE V.   COMPARISON OF EXECUTION TIME FOR ID3 & C4.5 ALGORITHM

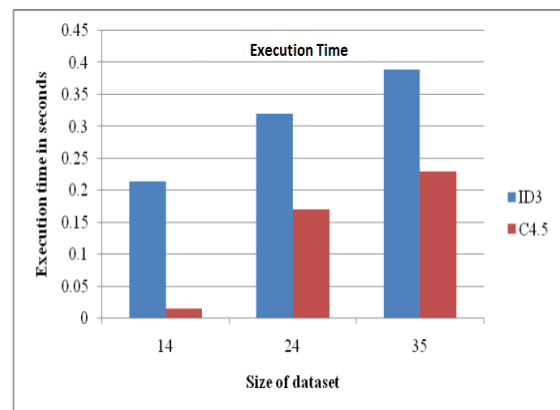| Size of Data Set	Algorithm	
	ID3 (%)	C4.5 (%)
14	0.215	0.0015
24	0.32	0.17
35	0.39	0.23

Fig. 7. Comparison of Execution Time for ID3 & C4.5 Algorithm

B. C4.5 Vs C5.0

C4.5 was superseded in 1997 by a commercial system See5/C5.0 (C5.0 for Unix / Linux, See5 pour Windows).

The changes encompass new capabilities as well as much-improved efficiency, and include [13]:

- A variant of boosting, which constructs an ensemble of classifiers that are then voted to give a final classification. Boosting often leads to a dramatic improvement in predictive accuracy.

- New data types (e.g., dates), "not applicable" values, variable misclassification costs, and mechanisms to pre-filter attributes.

- Unordered rule sets—when a case is classified, all applicable rules are found and voted.

- This improves both the interpretability of rule sets and their predictive accuracy.

- Greatly improved scalability of both decision trees and (particularly) rule sets. Scalability is enhanced by multi-threading; C5.0 can take advantage of computers with multiple CPUs and/or cores [13].

C. C5.0 Vs CART

Classification and Regression Trees (CART) is a flexible method to describe how the variable Y distributes after assigning the forecast vector X. This model uses the binary tree to divide the forecast space into certain subsets on which

Y distribution is continuously even. Tree's leaf nodes correspond to different division areas which are determined by Splitting Rules relating to each internal node. By moving from the tree root to the leaf node, a forecast sample will be given an only leaf node, and Y distribution on this node also be determined.

CART uses GINI Index to determine in which attribute the branch should be generated. The strategy is to choose the attribute whose GINI Index is a minimum after splitting.

Let S be a sample, a the target attribute, S1, ..., SK were starting from S, according to the classes of a

$$Gini(S) = \sum_{i=1}^{K} \frac{|S_i|}{|S|}\left(1 - \frac{|S_i|}{|S|}\right) = \sum_{i \neq j} \frac{|S_i| \times |S_j|}{|S|^2} \quad (7)$$

The C5.0 algorithm differs in several respects from CART, for example:

- The CART tests are always binary, but C5.0 allows two or more outcomes.

- CART uses the Gini diversity index for classifying tests, while C5.0 uses criteria based on the information.

- CART prunes trees using a complex model whose parameters are estimated by cross-validation; C5.0 uses a single-pass algorithm derived from binomial confidence limits.

- CART looks for alternative tests that approximate the results when tested attribute has an unknown value, but C5.0 distributes cases among probabilistic results.

- Speed of C5.0 algorithm is significantly faster and more accurate than C4.5.

VI. CONLUSION

Decision trees are simply responding to a problem of discrimination is one of the few methods that can be presented quickly enough to a non-specialist audience data processing without getting lost in difficult to understand mathematical formulations. In this article, we wanted to focus on the key elements of their construction from a set of data, then we presented the algorithm ID3 and C4.5 that respond to these specifications. And we did compare ID3/C4.5, C4.5/C5.0 and C5.0/CART, which led us to confirm that the most powerful and preferred method in machine learning is certainly C4.5.

REFERENCES

[1] Johan Baltié, DataMining : ID3 et C4.5, Promotion 2002, Spécialisation S.C.I.A. Ecole pour l'informatique et techniques avancées.

[2] Benjamin Devéze & Matthieu Fouquin, DATAMINING C4.5 – DBSCAN, PROMOTION 2005, SCIA Ecole pour l'informatique et techniques avancées.

[3] E-G. Talbi, Fouille de données (Data Mining) -Un tour d'horizon - Laboratoire d'Informatique, Fondamentale de Lille, OPAC.

[4] Ricco Rakotomalala, Arbres de Décision, Laboratoire ERIC, Université Lumière Lyon 2, 5, av. Mendés France 69676 BRON cedex e-mail : rakotoma@univ-lyon2.fr

[5] Arbres de décision, Ingénierie des connaissances (Master 2 ISC).

[6] Thanh Ha Dang, Mesures de discrimination et leurs applications en apprentissage inductif, Thèse de doctorat de l'Université de Paris 6, spécialité informatique, juillet 2007.

[7] Vincent GUIJARRO.K, Les Arbres de Décisions L'algorithme ID3, Elissa, "Title of paper if known," unpublished.

[8] Rakotoarimanana Rija Santaniaina, Rakotoniaina Solofoarisoa, Rakotondraompiana Solofo, Algorithmes à arbre de décision appliqués à la classification d'une image satellite.

[9] J. Fürnkranz, Entscheidungsbaum-Lernen (ID3, C4.5, etc.) (V1.1, 14.01.; neue Folie zu C4.5 Pruning)-- Site web : http://www.ke.tu-darmstadt.de/lehre/archiv/ws0809/mldm

[10] Ankur Shrivastava and Vijay Choudhary ,Comparison between ID3 and C4.5 in Contrast to IDS Surbhi Hardikar, VSRD-IJCSIT, Vol. 2 (7), 2012, 659-667.

[11] A.P.Subapriya, M.Kalimuthu, EFFICIENT DECISION TREE CONSTRUCTION IN UNREALIZED DATASET USING C4.5 ALGORITHM, ISSN: 2230-8563.

[12] Anurag Upadhayay ,Suneet Shukla, Sudsanshu Kumar, Empirical Comparison by data mining Classification algorithms (C 4.5 & C 5.0) for thyroid cancer data set. ISSN:2249-5789 .

[13] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg. Top 10 algorithms in data mining, Knowl Inf Syst (2008) 14:1–37 DOI 10.1007/s10115-007-0114-2

[14] Introduction to Data Mining and Knowledge Discovery, Third Edition by Two Crows Corporation.

[15] G. COSTANTINI R. Nicole, Probabilité conditionnelle. Indépendance, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[16] CELINE ROBARDET, Data Mining, http://prisma.insa-lyon.fr/sc.

[17] P. Habermehl et D. Kesner, Algorithmes d'apprentissage, Programmation Logique et IA

[18] Guillaume CALAS, Études des principaux algorithmes de data Mining, EPITA Ecole d'ingénieures en informatiques.

[19] , Payam Emami Khoonsari and AhmadReza Motie, A Comparison of Efficiency and Robustness of ID3 and C4.5 Algorithms Using Dynamic Test and Training Data Sets International Journal of Machine Learning and Computing, Vol. 2, No. 5, October 2012.

[20] Ron Kohavi, Ross Quinlan, Decision Tree DiscoveryOctober 1999, ISBN:0-19-511831-6.

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

[22] Leila BENAISSA KADDAR et Djamila MOKEDDEM, Construction Parallèle des Arbres de Décision, University of Science, Technology USTO- Algérie.

[23] Dr. Sanjay Ranka, Classification Part 1,,Professor Computer and Information Science and Engineering ,University of Florida, Gainesville.

[24] K. Ming Leung, Decision Trees and Decision Rules OLYTECHNIC UNIVERSITY, Department of Computer Science / Finance and Risk Engineering.

[25] http://id3alg.altervista.org/

[26] http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm

[27] J.R. Quinlan, C4.5 programs for machine learningMorgan Kaufmann Publishers, livres Google.

[28] http://rulequest.com/see5-comparison.html (cité dans l'article de R. Quinlan)

[29] http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie004.html#toc10

[30] http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/c4.5/tutorial.html

[31] Comparison of C5.0 & CART Classification algorithms using pruning technique International Journal of Engineering Research & Technology (IJERT),Vol. 1 Issue 4, June - 2012, ISSN: 2278-0181.