## ALGORITHM FOR MONITORING AND PREVENTING COFFEE LEAF RUST

Simón Marín Giraldo Universidad Eafit Colombia smaring 1 @eafit.edu.co Miguel Fernando Ramos García Universidad Eafit Colombia mframosg@eafit.edu.co Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co

#### **ABSTRACT**

Coffee is one of the most important products for colombian economy because it represents 6.9% of our exportations, contributing \$2.7 billion dollars to our country each year.

Coffee rust is a plague that affects plants leaving devastating loss in agricultural industry around the world. Coffee industry loses more than 30% of the production because of rust.

It is necessary to create a solution to reduce the impact of this plague and returning greater profits (or smaller loss) to colombian coffee industry. We proposed a decision tree data structure which determines if a coffee crop has coffee leaf rust. A decision tree algorithm learns from the data it analyzes and it determines the existence of rust with more ease each time you pass it new data.

### Keywords

Agriculture, Algorithm, Big-O Complexity, CART, Coffee, Data, Data Set, Data Structure, Decision Tree, Economy, Information Gain, Node, Rust, Training Data

# ACM CLASSIFICATION

Computing Classification System → Poly-Hierarchical → Semantic Web → Standard Classification → Evolves in the Future → Search Interface → ACM Digital Library → ACM Press → Quick Content → Related Literature → Online Resources → LaTeX → CCS → ACM → Full-Text Collection → Guide to Computing Literature → Connections → Hardware → Software → Theory of Computation → Applied Computing → Biographies → Data Management Systems → Database Design and Models → Networks → Public Internet → Classification Tree.

#### 1. INTRODUCTION

Agricultural industry is fundamental for all the world. It provides food from crops and the work of the people living in the countryside. For our country, agriculture is one of the basis in our national economy. In Colombia we generate most of our food and we export to all the world mainly fruits and coffee. Coffee is the most important of our agricultural exportations, being the 6.9% of these and bringing more than \$2.7 billion dollars every year.

Many plant plagues exist around the world affecting agricultural industry. One of the most lethal is Rust. It affects mostly coffee crops, leaving, for Latin America, devastating loss of more than 30% of the crops, discouraging the production and reducing the profits, affecting, mainly, the coffee producers.

#### 2. PROBLEM

In this agricultural problem, we must follow up a coffee crop, reporting opportunely about plagues emergence. Eafit researchers have developed a conservatory with the ability of monitoring multiple variables that are associated with rust emergence such as: illumination, environmental temperature, ground temperature and ground pH. With this variables we have the objective of predicting whether a caturra coffee, in a determined time interval, has coffee rust. Furthermore, for each data sample we have an expert's evaluation confirming if the crop is affected by the plague.

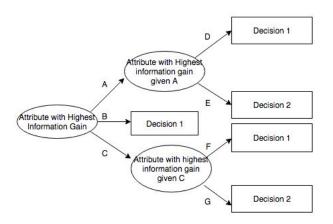
We are looking forward to give an early warning about rust occurrence in the crop in order to control it and reducing the loss because many crops get ruined due to this plague.

## 3. RELATED WORK

# 3.1 ID3 Algorithm

ID3 algorithm keeps generating a node and splitting the training instances until there is no more instance left. The attribute selection criterion is choosing the largest value of the information gain among the remaining attributes.

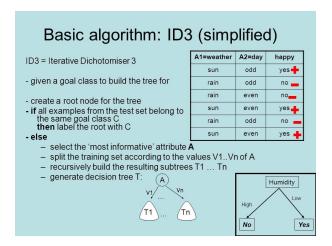
The examples set must be conformed by an ordered values series, each one of them known as an attribute, in which one of them, (the attribute to be classified) is the objective, that is binary (positive or negative, yes or no, valid or not valid, etc.).



# 3.2 C4.5 Algorithm

C4.5 is an extension of ID3 algorithm, developed previously by Quinlan. Decision trees generated by C4.5 can be used for classification. Due to this, C4.5 is frequently known as a statistic classifier. C4.5 generates decision trees from a data training set as ID3 does, using entropy of information concept. Training data is a data example group that has already been classified. Each example is a vector where the example's attributes or characteristics are represented. Training data are increased with a vector that represents the class where each data sample belongs.

In which tree node, C4.5 picks the attribute from the data that most efficiently divides the samples set in enriched subsets in one or other class. C4.5's criteria is normalizing for information gaining (entropy difference) that results in selecting an attribute for dividing the data. The attribute with the highest information gaining is picked as the decision parameter.



#### 3.3 CN2 Algorithm

The CN2 induction algorithm is a learning algorithm for rule induction. It is designed to work when the training data is imperfect. It is designed to work even when the training data is imperfect. It is based on ideas from the AQ algorithm and the ID3 algorithm. As a consequence it creates a rule set like that created by AQ but is able to handle noisy data like ID3.

The algorithm must be given a set of examples, TrainingSet, which have already been classified in order to generate a list of classification rules. A set of conditions, SimpleConditionSet, which can be applied, alone or in combination, to any set of examples is predefined to be used for the classification.

## CN2

```
CN2ForOneClass(examples, class)
Rules ← {}
Repeat
Bestcond ← FindBestCondition(examples, class)
If bestcond ⇔ null then
Add the rule "IF bestcond THEN PREDICT class"
Remove from examples all + cases in
class covered by bestcond
Until bestcond = null
Return rules
```

Keeps negative examples around so future rules won't impact existing negatives (allows unordered rules)

# 3.4 CHAID (Chi-square automatic interaction detection) Algorithm

Chi-square automatic interaction detection (CHAID) is a decision tree technique, based on adjusted significance testing (Bonferroni testing).

CHAID can be used for prediction (in a similar fashion to regression analysis, this version of CHAID being originally known as XAID) as well as classification, and for detection of interaction between variables. CHAID is based on a formal extension of the United States' AID (Automatic Interaction Detection) and THAID (THeta Automatic Interaction Detection) procedures of the 1960s and 1970s, which in turn were extensions of earlier research, including that performed in the UK in the 1950s.

In practice, CHAID is often used in the context of direct marketing to select groups of consumers and predict how their responses to some variables affect other variables, although other early applications were in the field of medical and psychiatric research.

Like other decision trees, CHAID's advantages are that its output is highly visual and easy to interpret. Because it uses multiway splits by default, it needs rather large sample sizes to work effectively, since with small sample sizes the respondent groups can quickly become too small for reliable analysis.

One important advantage of CHAID over alternatives such as multiple regression is that it is non-parametric.

height will be the same as the number of variables in the dataset.

## 4. CART

CART decision trees create child nodes with two branches for each conditional evaluated going through the tree. The

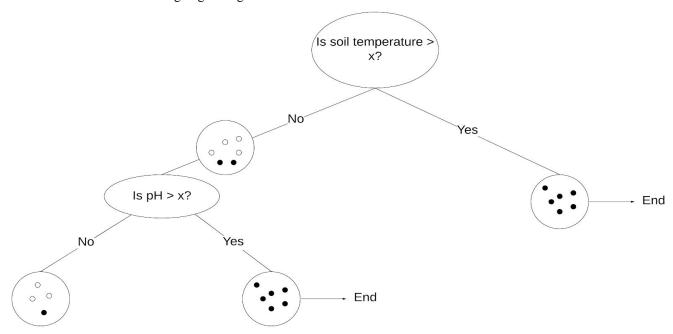
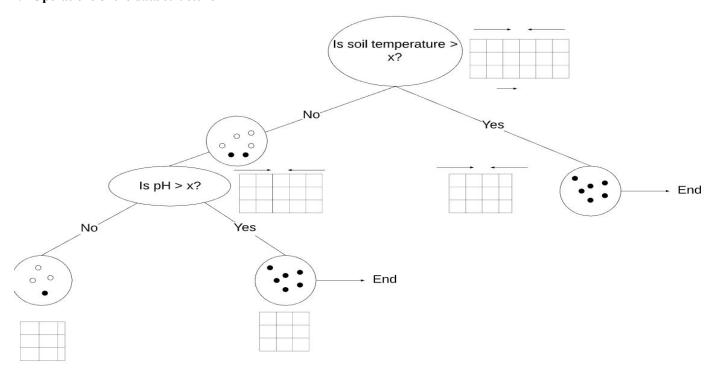


Figure 1: CART data structure evaluates and creates new child nodes based on the results it obtains.

# 4.1 Operations of the data structure



**Figure 2:** CART tree algorithm is going through the dataset. as it finds the lowest gini impurity which divides the data, it creates nodes with conditions and takes the variable processed out of the matrix to proceed with the next variables.

#### 4.2 Design criteria of the data structure

Our data structure is a CART binary tree. We built the algorithm functions that call themselves recursively until all the data is processed. This will ease the process because we can then make it work with any problem size. Also, recursion provides better Big-O complexity.

# 4.3 Complexity analysis

Operation	Complexity
main(String [] args)	O(n.m)
leerArchivo(String data)	O(n.m)
seleccionarDataset()	O(n)
llenarMatriz(double [][] matriz)	O(n.m)
llenarImpureza(double [][] matriz)	O(n.m)
menores(double [][] matriz)	O(n)
addChild(Node childNode, int position)	O(1)
addNewNode(Node u, Object info, int i)	O(1)
numberOfNodesInTree(No de rootNode)	O(n)

Table 1: Table to report complexity analysis

# 4.4 Execution time

Operation	<b>Execution time</b>
leerArchivo(String data)	0.0ms
seleccionarDataset()	0.0ms
llenarMatriz(double [][] matriz)	0.0ms
llenarImpureza(double [][] matriz)	0.0ms
menores(double [][]	0.0ms

matriz)	
addChild(Node childNode, int position)	0.0ms
addNewNode(Node u, Object info, int i)	0.0ms
numberOfNodesInTree(No de rootNode)	0.0ms

**Table 2:** Execution time of the operations of the data structure for each data set.

# 4.5 Memory used

Data set	Memory used
data_set.csv	7.84 MB
data_set_balanced.csv	7.0 MB
data_set_train.csv	6.84 MB
data_set_test.csv	6.56 MB

**Table 3:** Memory used for each operation of the data structure and for each data set.

#### 4.6 Result analysis

Finally we solved the problem and our decision tree could finally determine if a coffee crop has leaf rust.

## 6. CONCLUSIONS

In this research we found out that decision trees can determine things and accelerate the solution time for a problem.

Our data structure could finally determine if a crop had coffee rust.

Decision trees can have many applications to real life problems. They can be used for predicting and preventing many negative things.

## 6.1 Future work

Actually our algorithm can only process six variables. In the future we expect to fix the code in order to make it work for any problem size. We would fix some functions for adapting the algorithm to particular cases.

## **ACKNOWLEDGEMENTS**

We specially thank for assistance given by Juan Sebastián Díaz Osorio, Computer Science and Engineering student at EAFIT who helped us understanding the problem and gave important recommendations for giving a solution.

We also thank Isabel Piedrahita Vélez, Computer Science and Engineering student at EAFIT and Data Structures and Algorithms instructor who helped us for understanding the concepts related with the course.

#### REFERENCES

- 1. Kaewrod, N. and Kietikul, J. 2018 22nd International Computer Science and Engineering Conference (ICSEC), (Chiang Mai, Thailand, Thailand), IEEE, 1-5.
- 2. Wikipedia. 2016. CN2 algorithm. (8 September 2016). Retrieved *August 11, 2019* from https://en.wikipedia.org/wiki/CN2 algorithm
- 3. Wikipedia. 2018. C4.5. (9 April 2018). Retrieved *August 11*, *2019* from <a href="https://es.wikipedia.org/wiki/C4.5">https://es.wikipedia.org/wiki/C4.5</a>
- 4. Cárdenas, J. Roya del cafeto. Retrieved *August 11, 2019* from <a href="https://www.croplifela.org/es/plagas/listado-de-plagas/roya-del-cafeto">https://www.croplifela.org/es/plagas/listado-de-plagas/roya-del-cafeto</a>
- 5. OEC. 2017. Colombia (COL) Exportaciones, Importaciones, y Socios comerciales. Retrieved *August 11*, 2019 from <a href="https://oec.world/es/profile/country/col/">https://oec.world/es/profile/country/col/</a>
- 6. Wikipedia. 2019. ID3 algorithm. (22 May 2019). Retrieved *August 11, 2019* from <a href="https://en.wikipedia.org/wiki/ID3\_algorithm">https://en.wikipedia.org/wiki/ID3\_algorithm</a>
- 7. synergy37AI. 2019. Tree algorithms: ID3, C4.5, C5.0 and CART. (20 February 2019). Retrieved *August 11, 2019* from

 $\frac{https://medium.com/datadriveninvestor/tree-algorithms-id3-c4-5-c5-0-and-cart-413387342164}{$ 

8. Wikipedia. 2019. Chi-square automatic interaction detection. (5 January 2019). Retrieved *August 11, 2019* from

https://en.wikipedia.org/wiki/Chi-square\_automatic\_interaction\_detection