IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

DECISION TREES NEST POSITION TRAINING TRAINING ABO VE TREE DECIDED DETERMINED, REGULA TION SWITCH MAXIMIZE ENTROPY

[1]Trần Văn Anh 1st, [2]Lý Hải Sơn 2nd, [3]Nguyễn Thu Nguyệt Minh[3rd]

Tran Van Anh, Van Lang of University, Việt Nam. E-mail: anh.tv@vlu.edu.vn
Ly Hai Son, Van Lang of University, Việt Nam. E-mail: son.ly@vlu.edu.vn
Nguyen Thu Nguyet Minh, Van Lang of University, Vietnam. E-mail: minh.ntn@vlu.edu.vn
1,2,3 Faculty of Fundamental Science, Van Lang University, Ward 13, Binh Thanh District, Ho Chi Minh City.

Abstract : Decision trees are a very popular and effective data classification method. Because building a classification tree does not require parameters, decision trees can work effectively with multi-dimensional data, the results are easy to understand and interpret, the learning and classification steps are fast and accurate. good accuracy, so it has been voted by the machine learning community as the most used and most successful data mining method compared to other methods for many years in a row. Besides, the decision tree learning algorithm is successfully applied in many data mining fields such as: data analysis, text data classification, financial analysis, genetic data classification and regression.

Keywords: **Decision Tree, Maximize entropy, *Machine learning*.**

INTRODUCTION

A Decision Tree is a structured hierarchical tree used to classify objects based on a sequence of rules. Object properties can be of different data types such as Binary, Nominal, Ordinal, Quantitative, while class properties must have data types. is Binary or Ordinal.

In life there are many situations where we observe, think and make decisions by asking questions. Starting from there, in Machine Learning there is a model designed in the form of questions, where the questions are arranged in tree form. That is the decision tree model that we will learn about in this article.

So what is a decision tree? The essence of a decision tree is a directed graph used for decision making. For example, after knowing your high school graduation exam score, you want to build a major registration strategy with a series of options:

If the total of your three subjects is greater than 28.5, you will apply to the IT major.

On the contrary, if your test score is less than or equal to 28.5, there is still a chance for you if your Math score is high because Math score has a multiplier of 2. Therefore, you decide to still choose IT if your Math score is 10. School In the remaining case, you register for the field of Economics and Transport.

Your set of questions and options above can be generalized into a decision tree:

The concept of machine learning encompasses the process of instructing a computer to enhance its proficiency in a given task. To be more precise, machine learning entails a computer refining its performance through repeated iterations of a task. In essence, the fundamental essence of machine learning lies in the utilization of algorithms to scrutinize accessible data, acquire knowledge from it, and subsequently make informed decisions or predictions. Rather than training computers to learn tasks through data and algorithms, the creation of software with specific instructions and operations is employed to accomplish a designated task.

The capabilities of current AI would be significantly restricted in the absence of machine learning, as it enables computers to deduce solutions without explicit programming. To illustrate this, consider the scenario where a program is required to identify cats in images.First, you provide the AI with a set of cat characteristics for the machine to recognize, such as fur color, body shape, size, etc.

Afterwards, you input a series of images into the AI, with the possibility that some or all of the images may have the label "cat" attached to them. This enables the machine to better identify and focus on specific characteristics and attributes associated with cats. Once the machine has gathered all the essential information about cats, it must learn how to detect the presence of a cat in an

image - "If the image exhibits certain features X, Y, or Z, there is a high likelihood, around 95%, that it could be a cat.

2. LITERATURE REVIEW

Given a training data set consisting of m elements, each element has n attributes and has a label (class). The basic idea of decision trees can be summarized as follows:

A decision tree is a form of tree data structure. Each internal node has from 2 to many child nodes. Leaf nodes only contain data that belongs to one class, the purity node. The tree construction process is carried out according to top-down rules. Start the root node, all learning data is at the root node:

- At each node t in the tree, if all data elements belong to only one class c , then label node t as c and return t as the leaf node.
- On the contrary, find the possibility of cutting data (split) s* among all possibilities s.
- Create k child nodes of t corresponding to the division s* with the data set in t.

Label edges connecting t to child nodes based on the division of s*; At the same time, partition the data elements from t to the corresponding child nodes.

- Recursively step 1 for each child node (1.. k) of t.

Newly arriving data is classified according to the path from the root to the leaves of the decision tree. The generated rule is based on the labels of the edges (IF - THEN), the classification results are based on the label of the leaf node into which the new data is placed. An example of building a decision tree is described in table 1 and figure 1

Table 1: Illustration of learning data to build the decision tree in Figure 1 (Weather episode)

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rain	mild	high	false	Yes
rain	cool	normal	false	Yes
rain	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rain	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rain	mild	high	true	No

Suppose new data arrives x = (rain, cool, high, true, ?) => Play=No

Figure 1: Illustration of building a decision tree with the Weather data set

The classification efficiency of the decision tree depends greatly on the choice of s* in step 2. This is a problem of class NP - Hard. There are many solutions that have been researched and applied based on the knowledge base of fields such as information theory, artificial intelligence, statistics, linear algebra (n-dimensional space), ...

The tree building process mainly depends on choosing the best attributes to partition the data. An attribute that is considered good is used to partition the data so that the smallest tree is obtained as a result. This selection is based on heuristics: choose the attribute that produces the purest nodes. Currently, there are two typical decision tree learning algorithms that have been successfully applied to select cross-section s* based on a single attribute: C4.5 by Quinlan, CART by Breiman and colleagues.

To evaluate and select attributes when partitioning data, Quinlan suggests using information gain (choose the attribute with the largest information gain) and gain ratio based on Shannon's entropy function. Meanwhile, Breiman proposed using the Gini index (choose the attribute with the smallest Gini index) to select the partition attribute.

Suppose at node t there are n data elements of k classes, the Gini index of node t is defined as follows:

$$Gini(t) = 1 - \sum_{i=1}^{k} p_i \quad (2.1)$$

Where p_i is the frequency (ratio) of the ith class in node t, equal to the number of elements of ith class divided by the total number of elements at node t.

If the data set in t is divided into two sets D_1 and D_2 (2 child nodes), each set has m_1 and m_2 elements then:

$$Gini(t)_{split} = \frac{m_1}{m} Gini(D1) + \frac{m_2}{m} Gini(D2) \quad (2.2)$$

The best attribute is the attribute with the value such that the Gini(t) - $Gini(t)_{\text{split index}}$ reaches the largest value (or the $Gini(t)_{\text{split index}}$ is the smallest).

Determining s* is based on information gain (Information Gain) based on Shannon's Entropy theory, the founder of information theory. The entropy of node t is calculated as follows:

$$E(t) = -\sum_{i=1}^{k} p_i \ log_2 p_i \quad (2.3)$$

Consider the case where set t is divided into two sets D_1 and D_2 (2 child nodes) as above. The average entropy after dividing node t is E_{new} which is defined as follows:

$$E_{new} = \frac{m_1}{m} E(D1) + \frac{m_2}{m} E(D2) \quad (2.4)$$

The information gain of node t is calculated as follows:

$$\text{Information Gain(t)} = E_{(t)} - E_{new} \quad (2.5)$$

The best attribute is the attribute with the highest information gain value or E_{new} is the smallest.

To clearly see the measurement value of the Entropy and Gini functions, we observe Figure 2:

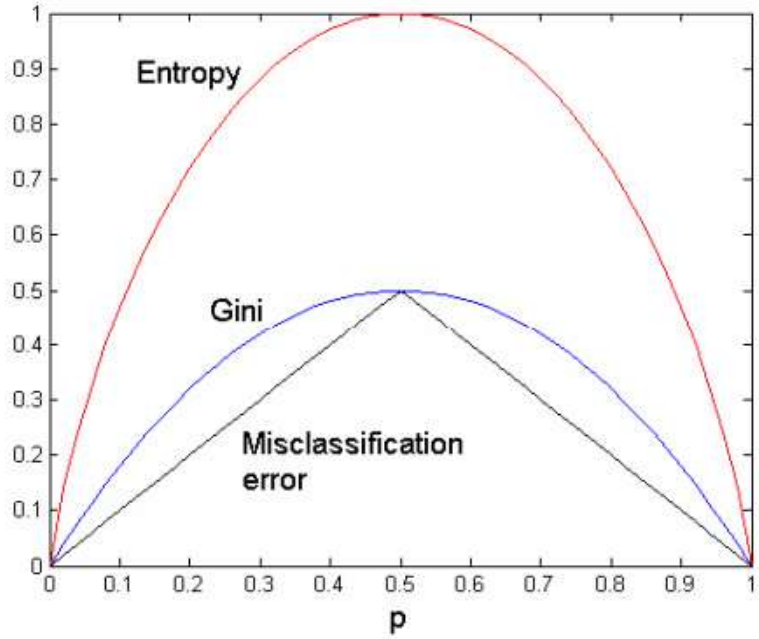

Figure 2: Comparison of impurity between Entropy and Gini functions with 2-class data

In there:

- P is the probability of the data belonging to 1 class (out of 2 classes).

- Error is the error function = 1 – max(p_i) , i=1..2 (2 classes). Looking at Figure 2, we see that all 3 functions reach a maximum when p=0.5 and a minimum when p=0 or p=1. That is, the Gini, Entropy and error indexes reach their smallest value when the data belongs to only one class and reach their maximum value when the probability of data belonging to each class is equal (p=0.5).

In general, to find s* of a node with m data elements we do the following:

Initialization s has a very large E_{new} (or Gini index);*
for (each i-th attribute=1..n)
for (each cut value - split s_{ij} of ith attribute) {
Split – split m data elements based on s_{ij} ;
Calculate E_{new} (or Gini index);
If (E_{new} (or Gini index) of s_{ij} < E_{new}
(or Gini index) of s) s*= s_{ij} ;*
}
return s is the j cutoff value of the ith attribute, where E_{new} (or Gini index) is minimum.*

3. EXAMPLE OF BUILDING A DECISION TREE BASED ON THE MAXIMIZE ENTROPY RULE

Example of the process of building a decision tree from a learning data table (table 1)

- Consider partitioning at the root node

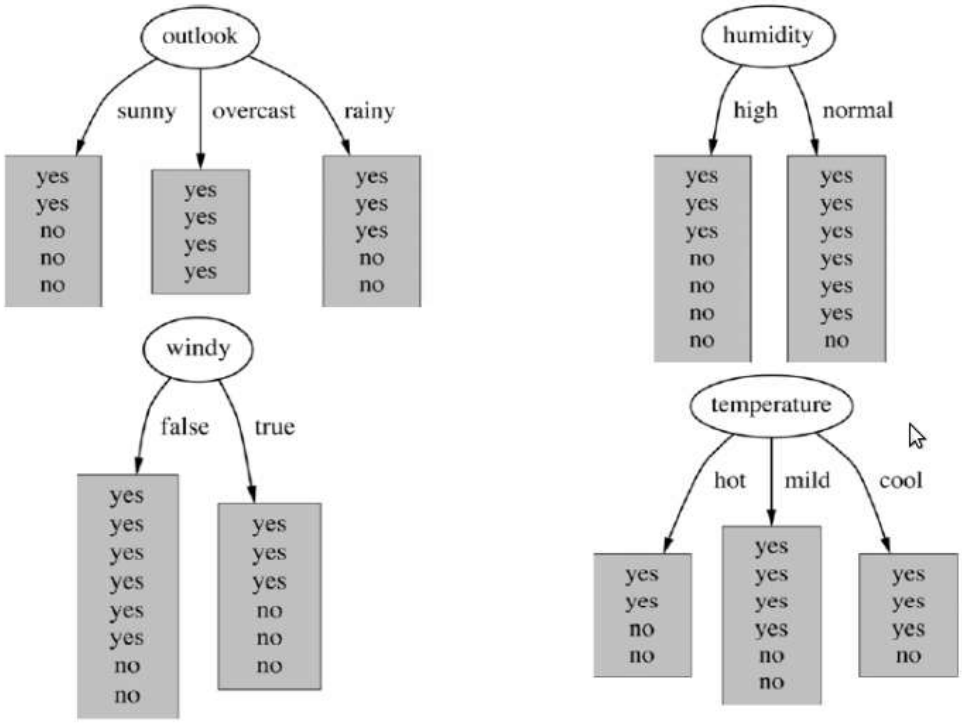

Figure 3: Single attribute partitioning based on information gain

- Outlook new E = 5/14 * E(2/5,3/5) + 4/14* E(4/4,0) + 5/14 * E(3/5,2/5)
  = 5/14 * (-2/5*log(2/5)-3/5*log(3/5))
  +4/14*(-1*log(1)-0*log(0))
  +5/14*(-3/5*log(3/5)-2/5*log(2/5))= 0.693
- with 0*log(0) considered = 0 , even though log(0) is undefined.
- E new of Humidity = 7/14*E(3/7,4/7) + 7/14* E(6/7,1/7) = 0.788
- new E = 8/14* E(6/8,2/8) + 6/14 * E (3/6,3/6) = 0.892
- E new of Temperature = 4/14 * E(2/4,2/4) + 6/14* E(4/6,2/6) + 4/14 * E(3/4,1/4) = 0.911

Because Outlook's E new = 0.693 is the smallest, choose the Outlook attribute for partitioning

- Continue to consider the branch partition Outlook=Sunny

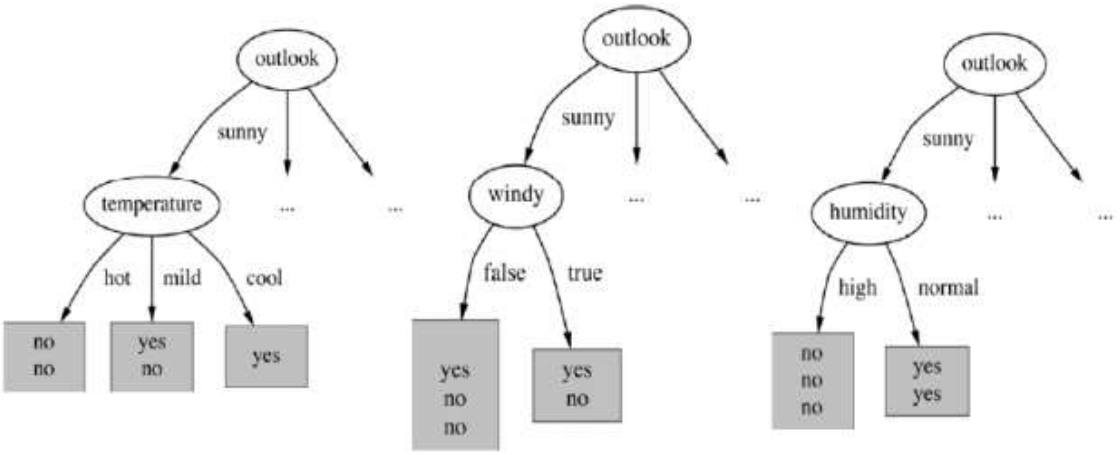

Figure 4: Outlook branch partition = Sunny

It is easy to see that the above partition belonging to Humidity hearing is the best because it results in 2 pure child nodes ($E_{new} = 0$).

- The Outlook= Overcast branch is pure.
- Outlook branch= Rain

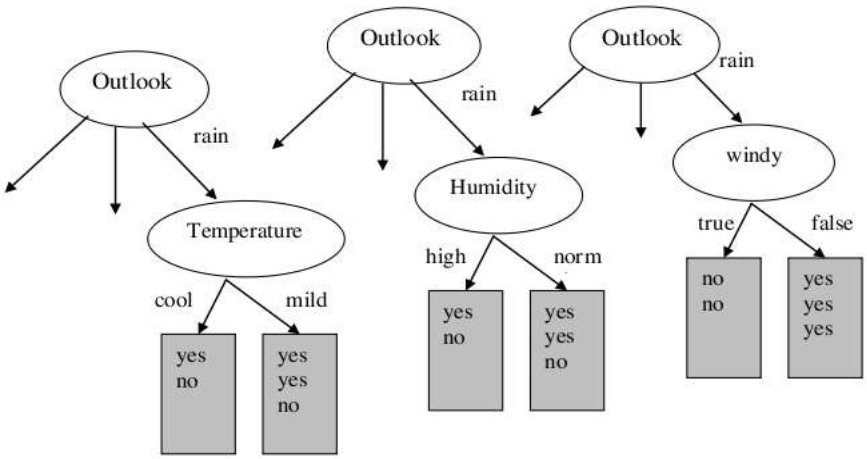

Figure 5: Branch partitioning Outlook = Rain

It is easy to see that the above partition belonging to the Windy hearing is the best because it yields 2 pure child nodes (E new = 0).

- The final result is a decision tree model as shown in Figure 1
- In the case of attributes with continuous numeric values, it is necessary to sort the data according to this attribute value and choose the best s* cutting values at the points where there is a change from one class to another (Fayyad & Irani, 1992).

❖ **Consider the numeric Weather set in table 2**

Table 2: Numerical Weather set

Outlook	Temperature	Humidity	Windy	Play
Sunny	85	85	FALSE	No
Sunny	80	90	TRUE	No
overcast	83	eighty six	FALSE	Yes
Rainy	70	96	FALSE	Yes
Rainy	68	80	FALSE	Yes
Rainy	65	70	TRUE	No
overcast	sixty four	65	TRUE	Yes
Sunny	72	95	FALSE	No
Sunny	69	70	FALSE	Yes
Rainy	75	80	FALSE	Yes
Sunny	75	70	TRUE	Yes
overcast	72	90	TRUE	Yes
overcast	81	75	FALSE	Yes
Rainy	71	91	TRUE	No

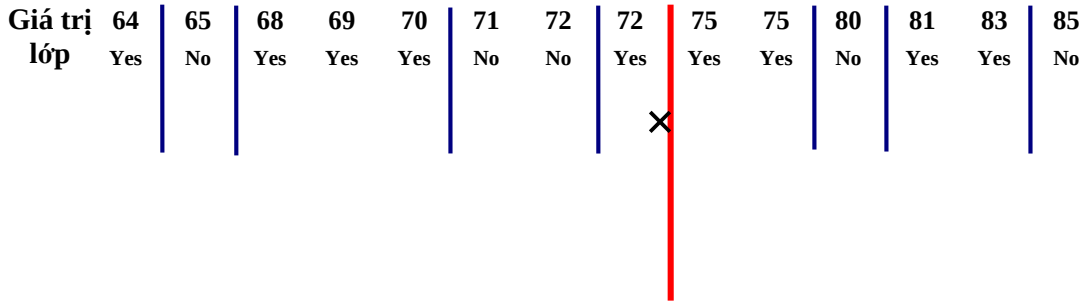

Figure 6: Illustration of data sorting and class change points on numeric attributes temperature, the long cutting line is the best cutting position when calculating Entropy or Gini.

In addition, to avoid "overfitting", it is necessary to combine the branch pruning technique after building the tree (postpruning) to increase the ability to classify or stop the branching process early (prepruning) if the branching does not occur. brings better results. Besides, considering solutions for handling data with missing values is also proposed by Breiman in CART and Quinlan in C4.5.

4. Conclusion and future directions according to

Advantage

Decision trees are simple and popular. This algorithm is commonly employed because of its advantages:

- The model produces rules that are simple to comprehend for the reader, each branch of the tree having a rule associated with it.
- Input data can be missing data, this is not necessary for normalizing or creating empty variables.
- Capable of working with both numerical and categorical information
- The model's validity can be determined using statistical procedures.
- Will likely collaborate with large data sets

Defect

Along with that, the decider tree also has specific disadvantages:

- The decision tree depends heavily on our input data. Even with a small change in the dataset, the structure of the decision tree model can change completely.
- Decision trees often have excessive problems

Acknowledgment

The authors would like to thank Van Lang University, the Faculty of Basic Sciences, and the Department of Informatics for supporting and facilitating this research.

REFERENCES

[1] David S. Evans, "The Economics of the Online Advertising Industry", *Review of Network Economics* , Vol.7, Issue 3, 2008.

[2] eMarketer Report, "US Ad Spending: eMarketer's Updated Estimates and Forecast for 2017", 2017.

[3] Robert B. Cleveland, Willam S. Cleveland, Jean E. McRae, and Irma Terpenning, "STL: A Seasonal-Trend Decomposition Procedure Based on Loess", Journal of Official Statistics , Vol.6, No. 1, 1990, pp. 3-73.

[4] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou , Tony Xing, Mao Yang, Jie Tong, Qi Zhang, "Time-Series Anomaly Detection Service at Microsoft", Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining , 2019.

[5] Luis M. de Alfonso, "Quantitative Methods 3 Level II 2020", DBF Finance , 2020.

[6] Sepp Hochreiter, Jurgen Schmidhuber, "Long Short-term memory ", Neural Computation , Vol.9, 1997, pp. 1735-1780.

[7] Sima Siami-Namini, Neda Tavakoli, Akbar Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series", 17th IEEE International Conference on Machine Learning and Applications, 2018.

[8] Vladik Kreinovich, Hung T. Nguyen, Rujira Ouncharoen, "How to Estimate Forecasting Quality: A SystemMotivated Derivation of Symmetric Mean Absolute Percentage Error (SMAPE) and Other Similar Characteristics", Technical Report:

[9] Ta Anh Son, Nguyen Thi Thuy Linh, Nguyen Ngoc Dang, "Solving Resource Forecasting in WiFi Networks by Hybrid AR-LSTM Model", The International Conference on Intelligent Systems & Networks - ICISN , 2021.

[10] Brownlee J., "Deep Learning for Time Series", 2019.

[11] Zhang GP, "Time series forecasting using a hybrid ARIMA and neural network model". Neurocomputing, Vol. 50, 2003, pp. 159–175.

[12] Mushtaq R., "Augmented dickey fuller test", SSRN Electronic Journal , 2011.

[13] Jenkins, GM, Box, GEP, "Time Series Analysis, Forecasting and Control", 3rd edn, 1970.

[14] Ta Older brother Son, Nguyen Exam Phuong, "Long-short term memory. memory networks for resource allocation forecasting in Wifi networks", 6th NAFOSTED Conference on Information and Computer Science (NICS) , 2019, pp. 348–351.

[15] Khashei, M., Bijari, M., "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting", Applied Soft Computing , Vol. 11, Issue 2, 2011, pp. 2664–2675.