International Journal of Engineering Innovation & Research
Volume 2, Issue 3, ISSN: 2277 – 5668

A Simple Dichotomizer

Dr. K. Veerabhadra Rao, Adeeba Riaz
Department of Computer Science & Engineering;
Muffakam Jah College of Engineering & Technology,
Banjara-Hills, Hyderabad-500034, India

***Abstract* – In most of the papers written on ID3 and C4.5 classifiers the example taken is the weather dataset of 14 days related with the playing of 'Base-Ball'. The authors have taken these 14 samples for training the system of the classifier .We will show in this paper that only 5 samples are sufficient for training the classifier with the four weather-attributes viz. Outlook, Temperature, Humidity & Wind-speed. Actually, there will be a total of 36 combinations with these 4 attributes with 3 level values in Outlook (namely Sunny, Overcast &Rain), 3 level values in Temperature (namely Hot, Mild & Cool), 2 level values in Humidity (namely High &Normal) and 2 level values in Wind-Speed (namely Strong & Weak). (i.e. 3x3x2x2=36 combinations) .The sufficiency requirement of 5 samples is proved from Multi-level Boolean functions of switching theory**.

***Keywords* – Machine Learning, ID3 Algorithm, Entropy, Information Gain, Decision-Tree.**

I. INTRODUCTION

In this paper we explain about a new type of dichotomizer, developed by us, which uses a least number of observations (of a data-set) to train this dichotomizer. We name this dichotomizer as 'ASD-1' (A Simple Dichotomizer-1).

A dichotomizer[1] is a classifier tool to classify the given set of objects into 2 separate sets with certain common features, say, to classify whether the currently received email is a phishing email or not. Thus, a dichotomizer is also used as a predictor.

A dichotomizer is developed through machine learning (un-supervised) with certain training data-set. The popular dichotomizer is ID3 **(Iterative Dichotomiser 3)** developed by Ross Quinlan [1,2] in 1979. In this method he develops a Decision Tree obtained through the Entropy & Information Gains of the parameters, which is used for the classification. ID3 method is explained, below, with his own training set of data.

1. Take all unused attributes and count their entropy concerning test samples
2. Choose attribute for which entropy is minimum (or, equivalently, information gain is maximum)
3. Make node containing that attribute

We take the same example taken by Ross Quinlan to predict "should we play Base-ball?" on that day, for which answer shall be in 'YES' or 'NO' terms.

To predict whether to play or not, four weather attributes viz Outlook, Temperature, Humidity, and Wind-speed are used. They can have the following values:

Outlook = { Sunny, Overcast, Rain }
Temperature = {Hot, Mild, Cool}
Humidity = {High, Normal}
Wind-speed = {Weak, Strong}

The training set consists of the observations made on 14 days. This data set consists of the values of 4 weather attributes with the final result whether the 'Base-Ball' was played or not.

Observation set S used for training

Table1: Training-Set for Machine Learning for Play Base-ball problem

Day	Outlook	Tempe-rature	Humidity	Wind-Speed	Play-ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

In ID3 algorithm we compute the Entropy [3] and Information Gain [1] of each attribute & its values. After computing the information gains, the attributes are arranged in descending order and a decision tree is formed.

II. DECISION-TREE GENERATION

Following is the Decision-Tree for the above training set, obtained with the ID3 algorithm [1], with target output values of "YES" or "NO" "Yes' for 'Play' and "No' for 'Do Not Play'.

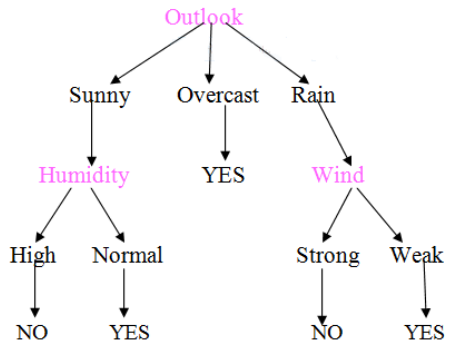

Fig.1. Decision-Tree diagram

International Journal of Engineering Innovation & Research
Volume 2, Issue 3, ISSN: 2277 – 5668

III. Formation of Rules from Decision-Tree

For programming the above Decision Tree with a computer language, such as C-language or Matlab, the decision tree is translated into a set of rules containing AND & OR binary operations which can be easily implemented in any computer language.

This Decision-Tree is described as five rules, one rule for each path of the decision-tree (There are only 5 paths in the above decision-tree.)

For framing the Rule1, the path taken is Sunny (Outlook) AND High (Humidity). Do not Play.

Similarly, the remaining 4 paths can be encoded into 4 rules.

All the 5 rules are listed below.

Rule #1: if Outlook=Sunny AND Humidity=High then "Do Not Play"

Rule #2: if Outlook=Sunny AND Humidity=Normal then "Play"

Rule #3: if Outlook=Overcast then "Play"

Rule #4: if Outlook=Rain AND Wind=Strong then "Do Not Play"

Rule #5: if Outlook=Rain AND Wind=Weak then "Play"

IV. Classifying all Combinations of Weather Data-Set

We can see from the 4 weather attributes, there can be only 36 weather combinations (3 combinations of 'Outlook', 3 combinations of 'Temperature' 2 combinations of 'Humidity',2 combinations of 'Wind-speed' i.e 36=3x3x2x2) and the day under question , whether to play or not will fall under one of these 36 days.

If the above 5 rules can categorize all the 36 combinations into 'Play; and 'Do Not Play', the dichotomizer has achieved its objective. Actually, the above 5 rules categorize all the 36 days, properly, i.e not violating the results of observed data. The following Table-2 contains the categorization/classification of all the 36 weather-combination days.

Table 2: Classification of all the 36 weather combination days

Outlook	Temperature	Humidity	Wind-Speed	Play-ball
Sunny	Hot	High	Strong	No
Sunny	Hot	High	Weak	No
Sunny	Hot	Normal	Strong	Yes
Sunny	Hot	Normal	Weak	Yes
Sunny	Mild	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Mild	Normal	Strong	Yes
Sunny	Mild	Normal	Weak	Yes
Sunny	Cool	High	Strong	No
Sunny	Cool	High	Weak	No
Sunny	Cool	Normal	Strong	Yes
Sunny	Cool	Normal	Weak	Yes
Overcast	Hot	High	Strong	Yes
Overcast	Hot	High	Weak	Yes
Overcast	Hot	Normal	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Mild	High	Weak	Yes
Overcast	Mild	Normal	Strong	Yes
Overcast	Mild	Normal	Weak	Yes
Overcast	Cool	High	Strong	Yes
Overcast	Cool	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Cool	Normal	Weak	Yes
Rain	Hot	High	Strong	No
Rain	Hot	High	Weak	Yes
Rain	Hot	Normal	Strong	No
Rain	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No
Rain	Mild	High	Weak	Yes
Rain	Mild	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Cool	High	Strong	No
Rain	Cool	High	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Cool	Normal	Weak	Yes

Now, we will show that the Table-2 can also be obtained either taking the set of rules (Rule # 2, Rule # 3 & Rule #5) pertaining to 'Can be Played) alone or set of rules (Rule # 1 & Rule #4) pertaining to 'Do Not Play') alone.

While using the first set of rules alone (viz Rule2,3&5) to classify the 36 combinations, proceed with the assumptions that all the 36 days are 'Do Not Play' days and update this data with 'Play' which ever days meet these 3 rules.

If the second set of rules alone(viz Rule1&4) is used to classify the 36 combinations, proceed with the assumptions that all the 36 days are ' Play ' days and update this data with 'Do Not Play' which ever days meet these 2 rules. We have verified the generation of Table-2, separately, with these 2 sets of rules and found it to be correct.

As the classification can be done, properly, with either the 1st set of rules (viz Rule2,3&5) or with 2nd set of rules((viz Rule1&4), from the computation-time point of view it is better to use the set with less number of rules, in the present case it is the 2nd set which is preferred with only 2 rules.

Actually, we obtained the same Table2 by applying both the sets of rules for dichotomization, separately. We wrote two C-programmes, one for each rule-set.

V. Classifying all Combinations with Only One Set of Rules

Now our main proposition is that our classifier uses only the 2nd set of above rules (viz Rule1&4) pertaining to 'Do Not Play' days.

Even though, so far we have used the rules generated from ID3, now we will generate the above 2nd set of rules (viz Rule1&4) with a simple method using Boolean logic-with multi-levels[4] which is described below.

The data of observed 5 days of 'Did Not Play' from the 14 observed days (i.e Table-1) is reproduced below:

Table 3: List of 5 days on which Base-ball was Not-Played

Day	Outlook	Tempe-rature	Humidity	Wind-Speed	Play-ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D6	Rain	Cool	Normal	Strong	No
D8	Sunny	Mild	High	Weak	No
D14	Rain	Mild	High	Strong	No

VI. Generating Classification Rules from Multi-Level Boolean Algebra

In the following paragraphs, we explain a simple classification method, what we named as "A Simple Dicotomizer-1(ASD-1)" .

We put the above Table-3 contents of 'Did Not Play' cases in the following condensed way:

Row1: R1:	Sunny	Hot	High	Weak
Row2: R2:	Sunny	Hot	High	Strong
Row3: R3:	Rain	Cool	Normal	Strong
Row4: R4:	Sunny	Mild	High	Weak
Row5: R5:	Rain	Mild	High	Strong

The above last 4 columns pertain to the 4 weather attributes viz. Outlook, Temperature, Humidity, and Wind-speed, respectively.

A. Procedure

Let us apply the Boolean Logic (of multi-levels)[4] to the above 5 rows.

In R1& R2, only the last column differs, that is 'Did Not Play' when the first 3 values exist on that day, irrespective of whether the 'Wind-speed is "Strong or weak". That means, we can create a single row(R1,R2) by placing 'do not care', with the letter 'd' under the last column. The following Row6 replaces the two rows Row1&Row2.

Row6: (R1, R2) Sunny Hot High 'd'

From, rows R1&R4, 'Did Not Play' when the temperature is 'Mild' or 'Cool' when the remaining 3 values are: Sunny, High & Weak .

As 'Mild' & 'Cool' temperatures are lower than 'Hot", we can extrapolate that 'Did Not Play' when the temperature is 'Hot' or 'Mild' or 'Cool'. That is 'Temperature' is not a deciding factor at all to play or not to play. Therefore, we can place 'd' in 'Temperature" columns of rows(1&4). Now rewrite R1&R4 as Row7, where

Row7: (R1&R4) Sunny 'd' High Weak

From Row6 &Row7, only 4th column from right & 2nd column from right are deciding, as the 3rd column from right and 1st column from right are having 'do not care' symbol 'd'.

Therefore, Row6& Row7 can be combined as

Row8: (R1, R2,R4) Sunny 'd' High 'd'

Thus, 3 observations R1, R2 & R4 can be represented by a single function given by Row8.

From R3&R5, we can see, 'Did Not play', whether the 'Humidity' is 'High' or 'Normal', when the other 3 values are: Rain, Cool /Mild & Strong.

As it was explained earlier, 'Hot' temperature is much higher than 'mild' & 'Cool', thus, the 'Temperature is not a deciding factor.

Now combine R3&R5 in the light of above argument and create Row9 as

Row9: (R3, R5) Rain 'd' 'd' strong.

Thus, the 5 observations of 'Did Not Play' from Table-1 can be represented with the above Row8 &Row9, which are listed below:

Sunny 'd' High 'd'

Rain 'd' 'd' strong.

If we state the above 2 rows in rule forms;

If Outlook=Sunny AND Humidity=High , 'Do not play'

and also

If Outlook=Rain AND Wind-speed=Strong, 'Do not play'

Notice, that these are the same rules derived from the Decision-Tree developed with ID3 algorithm, for 'Do Not Play' condition.

VII. Conclusion

Actually, we have written a programme in Matlab to simplify the multi-level Boolean functions and obtained the above results. Quine-McCluskey method [5] can also be used by encoding the multi-values of the 2 features, viz. Outlook & Temperature which are having 3 values each. With this encoding the problem in Quine-McCluskey method becomes a 2-level problem of six variables. But, we preferred the multi-level (3-level) Boolean simplification as we can handle the problem without an increase in number of variables i.e with only 4 variables. More over the multi-level Boolean simplification method will not have the limitation on the number of levels of a single feature.

Thus, our dichotomizer (ASD1) is much simpler than ID3 and it requires only a small amount of data for training the classifier.

Acknowledgements

We thank Dr. A. A. Qyser, the Head of the Department of Computer Science & Engineering, for his continuous support in providing the facilities for conducting the Research. We also express our thanks to the Principal, Dr. Basheer Ahmed for encouraging the staff & students to conduct research and publish papers.

References

[1] J.R.Quinlan," Induction of decision trees " in Machine Learning, Vol. 1, No. 1. (1 March 1986), pp. 81-106

[2] J.Ross Quinlan, "C4.5: Programs for Machine Learning", Morgan Kaufmann Publishers, Inc., 1993.

[3] Robert G. Gallager, "Information Theory and Reliable Communication ",John Wiley & Sons Inc. ,1968.
[4] Mitchell P. Marcus, "Switching Circuits for Engineers", 3rdEdition, Prentice-Hall,1975.
[5] Zvi Kohavi, "Switching and Finite Automata Theory", McGraw-Hill Companies Inc., 1970.

AUTHORS' PROFILE

Dr. K. Veerabhadra Rao

Professor, has completed B.E (E.C.E), M.E (E.C.E), Ph.D. (Eng'g) in 1965, 1967 &1986, respectively. All these three degrees were obtained by him from Osmania University, Hyderabad.

After completing his M.E (E.C.E), he joined in DLRL, Hyderabad in June, 1967 and worked for 25 years in DLRL. In 1992, when he was Sc'F', shifted to missiles lab Research Centre Imarat (RCI) to head the Signal Processing Group. Later on he became the Director of Seeker-Head Laboratory of RCI in Nov, 1995. After his retirement in Sepeber, 2001, he is working in private industries and & private engineering colleges. For the last 6 years, he is working as a professor in Computer Science & Engineering Department of Muffakham Jah College of Engineering & Technology, Banjara-hills, Hyderabad,

During his active service, he got 3 national awards. He published 2 papers in IEEE Transactions and nearly 15 articles in national journals. He visited 16 countries, so far. Email: drkv_rao@yahoo.com

Adeeba Riaz

M.Tech. 4th Semester Student, has completed her Bachelor's degree in Computer Science & Engineering from Muffakham Jah College of Engineering & Technology, Hyderabad in 2011 and she is currently in her final year of M.Tech. (Computer Science & Engineering) at the same college. Email: adeeba.mtech@mjcollege.ac.in

.