

New approaches for decision making in information systems via decision diagrams

A. S. Salama

Mathematic Department, Faculty of Science, Tanta University, Tanta, Egypt
Mathematic Department, Faculty of Science and Human Studies, Shaqra University, Dawadmi, KSA
dr_salama75@yahoo.com

Abstract: In this paper we study new approach for data representation using decision diagrams in information systems. Firstly we signify data using binary decision diagrams. Also, we used ID3-Decision tree learning algorithm to reduct information systems. Some algorithms are built and developed to refinement the process of decision making in information systems. Illustrative figures and examples are presented and real life application examples are given.

Keywords: ID3 learning algorithm; Data representations ; Decision making; Decision diagram; Information system.

1. Introduction

We often get together decision making problems in our daily life or working surroundings. From time to time it is very difficult for us to make good decisions. In observe , we usually use our past understanding as a form of performing experiments costs time and money. Providentially, the developments of computer technologies and automatic learning techniques can make this easier and more resourceful. There exist many approaches of decision making techniques, such as decision trees or decision diagrams , artificial neural networks and Bayesian learning. In this paper , we focused on the decision diagrams especially on Binary Decision Diagram (BDD), Binary Decision Diagram Information System (BDDIS) and Decision Tree Learning Algorithm (DTLA) [14, 29, 31, 32, 36].

There exist many methods to do decision analysis. Each method has its own advantages and disadvantages [11, 13, 24-26].

The approach of BDD is used for binary data representation and it is very efficient for Boolean function representation such that relations, relational algebra, midterms, cubes, sets, sets of sets , state machines , partitions, set systems , graphs, covering, tables and matrices, but it is not understand for non-technical people.

For data in an information system, the acquisition of knowledge and reasoning may involve vagueness, incompleteness, and granularity. In order to deal with the incomplete and vague information in classification, concept formulation, and data analysis, researchers have proposed many methods other than classical logic, for example, rough fuzzy sets[17,18,19], rough set theory and its generalizations [2,7,8,9-11,14,15,20], computing with words, granular computing, formal concept analysis, quotient space theory , and computational theory for linguistic dynamic systems. The advantage of the rough set method is that it does not need any additional information about the data, like probability in statistics or membership in fuzzy set theory. The main idea of applications of the rough theory in knowledge discovery comes from W. Ziarko work [40-43]. Many researchers made contributions to this theory. Applications of the rough set and fuzzy set theories can be found in [11,14-17].

2. Decision making using Binary Decision Diagrams

The main aim of this section is to describe how data represents by Binary Decision Diagrams (BDD) and how to obtain reducts using it.

A BDD is a rooted , directed a cyclic graph, with two sink nodes labeled by constants 0 and 1, and whose internal nodes has two outs. Each internal node v is labeled by a binary variable $\text{var}(v) = x_i$, $i = 1, \dots, n$. It is two outgoing edges which are labeled by 0

and 1, and are usually depicted as dotted and solid lines, respectively. Let the two corresponding successor nodes of v be denoted by $low(v)$ and $high(v)$, respectively.

A BDD is ordered (OBDD) if, on all paths through the graph, the variables respect a given linear order $x_1 \prec x_2 \prec \dots \prec x_n$ i.e., for each edge leading from a node labeled by x_i to a node labeled by x_j , it holds that $x_i \prec x_j$. We can say that variable must appear in ascending order along all paths (see Figure 1).

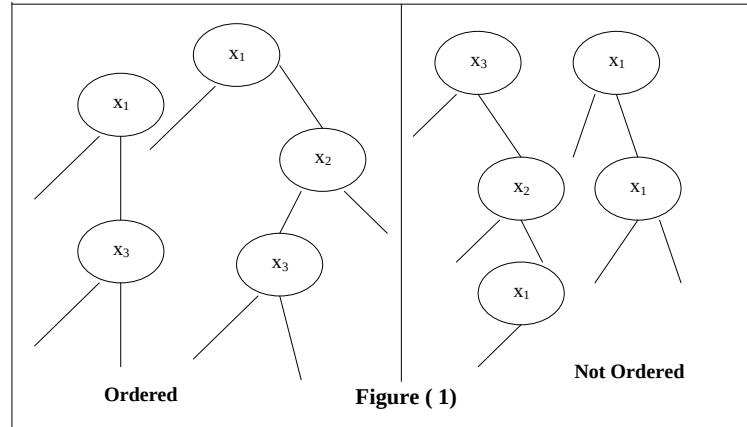


Figure (2) displays an example of an OBDD for a three variables function given by the Boolean formula:

$f(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge x_3$, with respect to the variable order $x_1 \prec x_2 \prec x_3$. Table 1 shows the values of this function.

x_1	x_2	x_3	$(x_1 \vee x_2) \wedge x_3$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Table 1

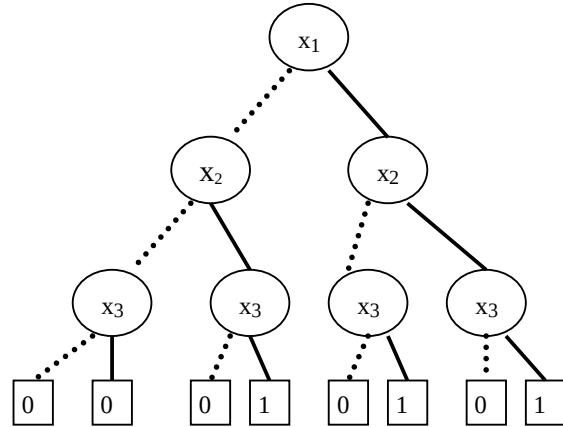
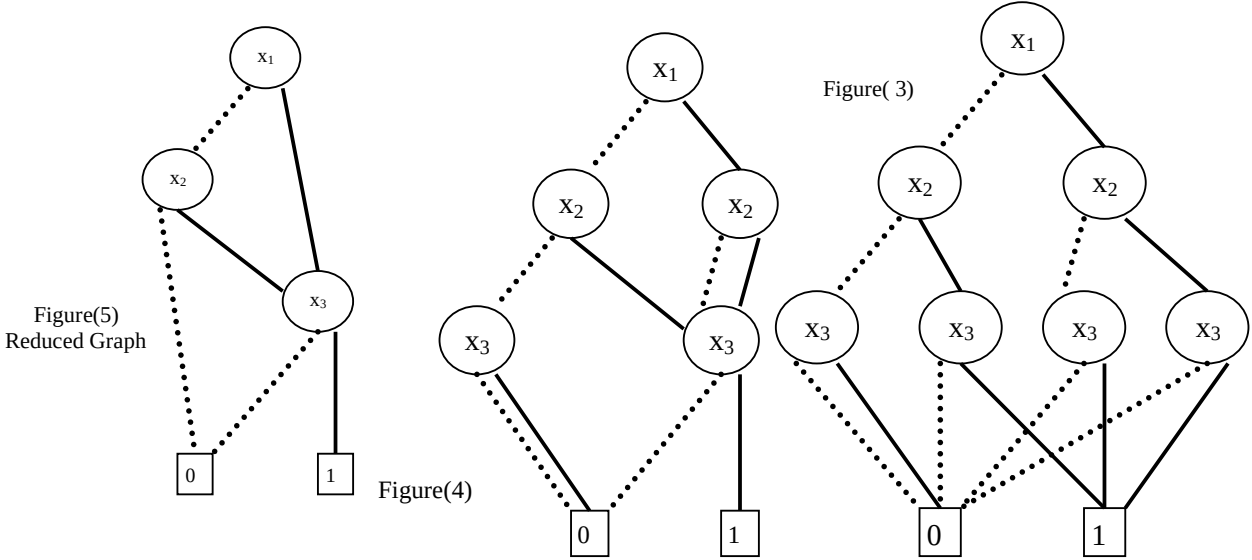


Figure (2)

Up to here, OBDD are not yet uniquely determined for each function. However, by further restricting the representation, a canonical form of BDDs. We observed that reduced OBDD (ROBDD) are a canonical representation of Boolean functions.

An OBDD is said to be reduced, if it does not contain a node v whose successor nodes are identical (i.e., $low(v) = high(v)$) and no two distinct nodes u and w exist such that the sub graphs rooted in u and w are isomorphic, i.e., for which $var(u) = var(w)$, $low(u) = low(w)$ and $high(u) = high(w)$.

The ROBDD representation of any function is uniquely determined (up to isomorphism), so that several properties (e.g. functional equivalence, constant functions, etc.) become easily testable. Conceptually, a reduced diagram can be interpreted as the result of the repeated application of three types of transformations. One reduction rule is merge equivalent leaves. This rule means use only one terminal label for 0 and one terminal label for 1 and redirect all lines from n to the respective node. Figure (3) represents this reduction rule for the same function given in Figure (2).



Second reduction rule is to merge isomorphic nodes. The second rule means if two terminal nodes x, y are on the same level and $low(x) = low(y)$ and $high(x) = high(y)$, then eliminate one of them, say x and redirect all lines into x to y . Figure(4) represents the second rule.

The last reduction rule is eliminate redundant tests. This means if $low(x) = high(x)$ for some one terminal x , then remove x and redirect all lines into x to $low(x)$. Figure(5) shows the last reduced graph.

The result of applying these rules to an OBDD until none of them can be applied any more is called a reduced OBDD.

Reduced Algorithm

- Visit OBDD bottom up and label each vertex with an identifier.
- If $id(low(v)) = id(high(v))$, then vertex v is redundant and set $id(v) = id(low(v))$.
- If $id(low(v)) = id(high(w))$, and $id(high(v)) = id(high(w))$ then set $id(v) = id(u)$.
- A different identifier is given to each vertex at level i .
- Terminated when root is reached.
- An ROBDD is identified by a subset of vertices with different identifiers.

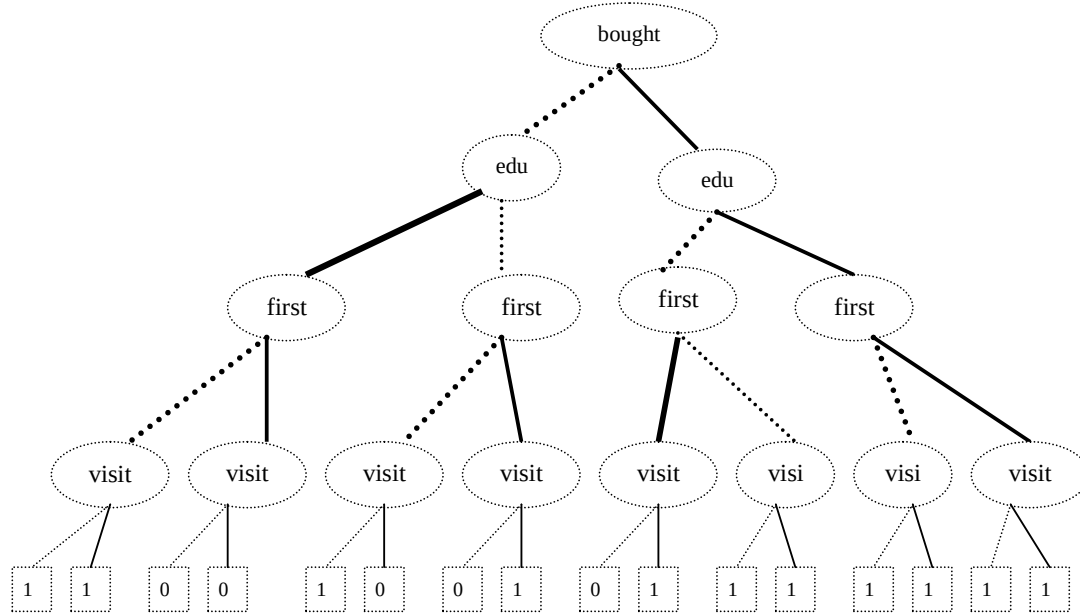
As we mentioned before the BDD is very efficient method for represents binary data, we will introduce an example from our daily life and we will try to make a reduction for container data using ROBDD.

Example 2.1 According to Table 2 in electric commerce applications we want to make predictions about what a user will do. Consider the following made-up data used to predict whether someone will ask for more information (More-info) based on whether they accessed from an educational domain (Edu.), whether this is a first visit (First), whether they have bought goods from an affiliated company (Bought), and whether they have visited a famous online information store (Visited).

<i>Examples</i>	<i>Bought</i>	<i>Edu.</i>	<i>First</i>	<i>Visited</i>	<i>More-info</i>
e ₁	false	true	false	false	true
e ₂	true	false	true	false	false
e ₃	false	false	true	true	true
e ₄	false	false	true	false	false
e ₅	false	false	false	true	false
e ₆	true	false	false	true	true
e ₇	true	false	false	false	true
e ₈	false	true	true	true	false
e ₉	false	true	true	false	false
e ₁₀	true	true	true	false	true
e ₁₁	true	true	false	true	true
e ₁₂	false	false	false	false	true
e ₁₃	true	true	false	false	true
e ₁₄	false	true	false	true	true
e ₁₅	true	false	true	true	true
e ₁₆	true	true	true	true	true

Table 2

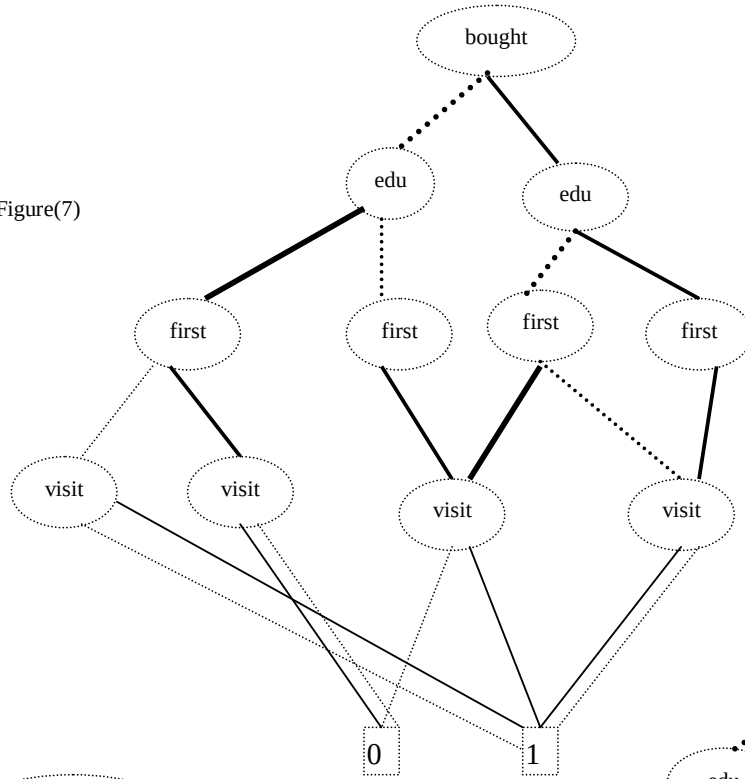
By using BDD we consider the attribute value true replaced by 1 and false by 0 , Figure(6) is the BDD for the data given in Table(2) using the order Bought $\prec$ Edu $\prec$ First $\prec$ Visited.



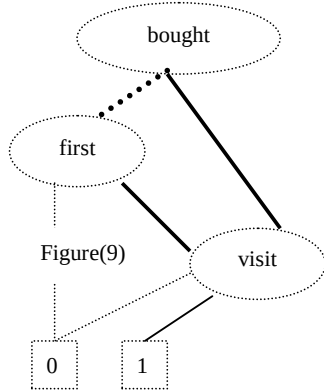
Figure(6)

According to reduction algorithm we have the following reduct figures.

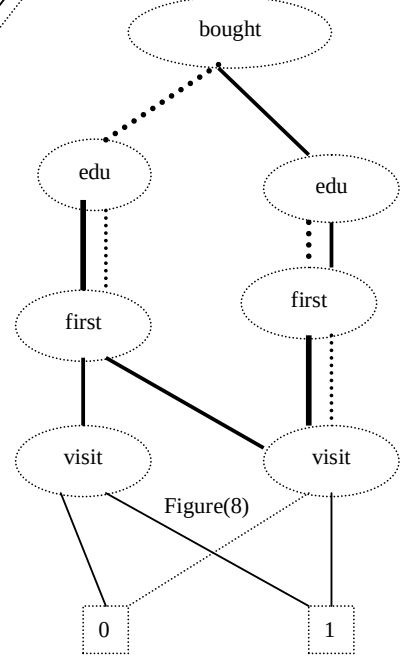
Figure(7)



Figure(9)



Figure(8)



A BDD is an efficient data structure for represents binary data (Boolean function). We use it to obtain reducts of our daily life binary data. But for the normal user, he cannot understand what is the meant by the final graph. So we can delete the redundant attributes from the table and rewrite it. By looking at the final graph the attribute Edu. can eliminated.

3. Decision making using BDDIS

A BDDIS has one or more root nodes each unique root node has a sub-tree that may share nodes with other objects and represents one or more indiscernible objects. The level root

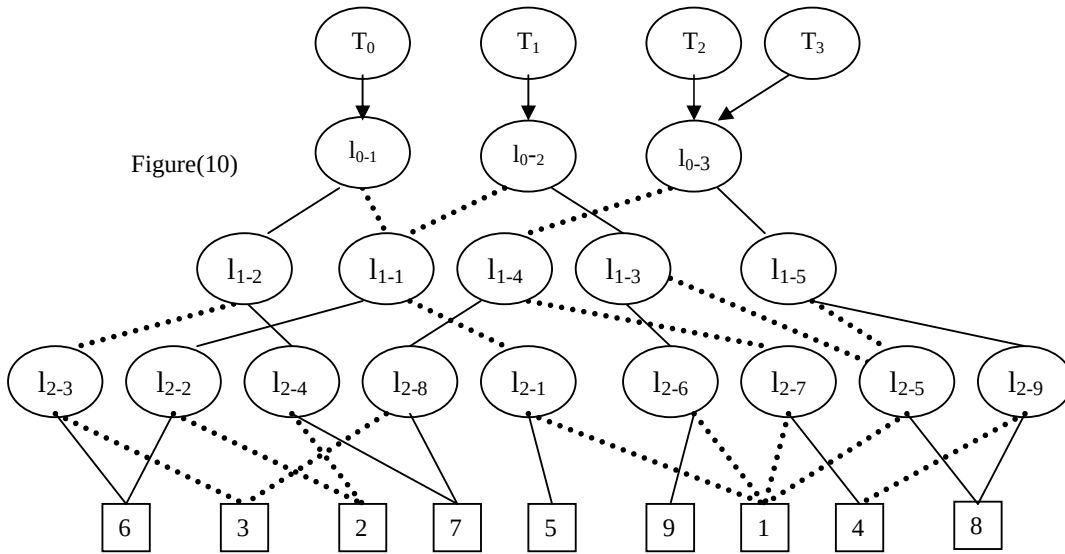
containing nodes which is defined as the object-top level ; a top level element is referenced by $T(x)$, $x \in U$.

The following information system can be represented by BDDIS using object-top level.

U	a	b	c	d
x_1	15	26	36	27
x_2	15	26	18	19
x_3	14	37	18	48
x_4	14	37	18	48

Table 3

The following diagram represents the data in Table 3 using BDDIS.



Each column within the information system is represented by a path composed of a unique series of l_o and/or h_i branches arriving at attribute top level node.

In Figure(9) the first column of Table 3 is represented by the path $l_o - l_o$, the second column is represented by the path $l_o - h_i$ and the third is represented by the path $h_i - l_o$. Each node is labeled with $l_n - x$ for level number n and node number x .

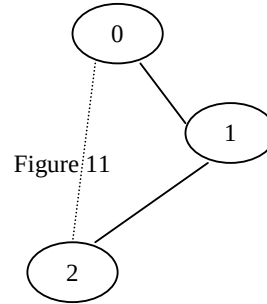
Two methods can be taken to reduce the size of the BDDIS. One can reduce the BDDIS by applying the ROBDD rules described before. Alternatively, one can create the BDDIS using a coded information system in which each attribute value V_a of the original information system is coded to value $0, 1, 2, \dots, k-1$, $k = |V_a|$; here each unique coded value represents a unique attribute values.

We can reduce the data of Table 3 by using the coded method. The corresponding coded data is given in Table 4.

U	a	b	c	d
x_1	0	0	0	0
x_2	0	0	1	1
x_3	2	2	1	2
x_4	2	2	1	2

Table 4

We can represent the coded table by attribute top level using BDDIS. By make a coded table we observe that the attribute values of attribute a is the same for the attribute value for attribute b, then we can represent BDDIS without attribute b. the reduced graph of the coded data is shown in Figure 10.



4. New Approach for decision making using decision trees

A decision tree (DT) is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision [12,13]. Decision tree is commonly used for gaining information for the purpose of decision making. DT starts with a root node on which it is for users to take actions. From this node, users split each node recursively according to DT in which each branch represents a possible scenario of decision and its outcome.

Decision Tree Learning Algorithm:

```

function DECISION –TREE-LEARNING (examples, attributes, default)
    returns a decision tree
    inputs: examples, set of examples, attributes, set of attributes
           Default, default value for the goal predicate
    If (examples is empty) then return default
    else if (all examples have the same classification ) then return the classification.
    else if (attributes is empty) then return MAJORITY-VALUE(examples)
    else
        best ← CHOOSE-ATTRIBUTE( attributes, examples)
        tree ← a new decision tree with root test best
        for each value  $v_i$  of best do
            example  $s_i$  ← {elements of examples with best =  $v_i$  }
            subtree ← DECISION-TREE-LEARNING(example  $s_i$ , attributes- best,

```

MAJORITY-VALUE(examples) add a branch to tree with label v_i and subtree subtree.
end

5. Remarks about ID3-decision tree learning algorithm

J. R. Quinlan in [31-34] have been developed Interactive Dichotomize 3 (ID3) algorithm for a simple decision tree learning by based on the concept learning system (CLS) algorithm. It builds the decision tree from some fixed or famous figurative data in order to classify them and foresee the classification of new data. The data must have more than a few attributes with different values, this data also has to belong to varied predefined separate classes (i.e., yes/no).

The entropy measure is defined as follows:

$$H(S) = \sum_{i=1}^N -p_i^+ \log_2(p_i^+) - p_i^- \log_2(p_i^-) \quad \text{where } p_i^+ \text{ is the ratio of the positive examples}$$

S^+ with respect to the set of all examples S i.e., $p_i^+ = |S^+|/|S|$. And p_i^- is the ratio of the negative examples S^- with respect to the set of all examples S i.e., $p_i^- = |S^-|/|S|$, where $|S|$ is the cardinality of S .

For example, we assume that a sample set S has 14 members altogether ($|S|=14$), including 9 positive ($|S^+|=9$) and 5 negative examples. Then the entropy of S is :

$$H(S) = -(9/14)\log_2(9/14) - 5/14\log_2(5/14) = 0.94.$$

Below we discuss the entropy in the special case of the Boolean classification.

- If all the members of a set S belong to the identical kind (have the same decision value), then the entropy is null. That means that there is no classification uncertainty.
- If the number of the positive examples equals to the number of the negative examples, then entropy takes its maximum value. In this case, the classification uncertainty is maximum.

These results express separately that the same set has no uncertainty (the decision is clear); or it is 100% uncertain for decision making. If the number of positive examples is not equal to the number of the negative examples, then the value of entropy is measured between 0 and 1.

To carry on the attribute expansion, which is based on the data of this sample set, we must define a standard measure called information gain. An information gain of an attribute is the final information content, which is a result of the reduction of the sample set entropy after using this attribute to divide the sample set. The definition of the information gain of an attribute a relates to the sample set S is :

$$G(S, a) = H(S) - \sum_{v \in V_a} |S_v|/|S| \cdot H(v), \quad \text{where } S_v = \{x \in S : V_a(x) = v, v \in V_a\} \text{ and } H(v) \text{ is}$$

the entropy of the a -attribute value v .

Example 5.1 In this example, people decide to go to city patch or stay at home according to the weather ($w = \{\text{Hot (w1), Mild (w2), Cool (w3)}\}$), rain ($r = \{\text{Strong (r1), Weak (r2)}\}$) and traffic ($t = \{\text{Long (t1), Short (t2)}\}$) situation as shown in Table 5. Just like Table 5, the set $S = \{s_1, \dots, s_{14}\}$ contains in total 14 examples (9 positive decisions and 5 negative ones). We can calculate the information gain of the attribute r as follows:

$$G(S, r) = H(S) - 8/14H(r2) - 6/14H(r1), \text{ where}$$

$$H(r2) = -6/8\log_2(6/8) - 2/8\log_2(2/8) \text{ and } H(r1) = -3/6\log_2(3/6) - 3/6\log_2(3/6).$$

Then $G(S, r) = 0.048$.

S	w	r	t	Decision
s ₁	w1	r2	t1	No
s ₂	w1	r1	t1	No
s ₃	w1	r2	t1	Yes
s ₄	w2	r2	t1	Yes
s ₅	w3	r2	t2	Yes
s ₆	w3	r1	t2	No
s ₇	w3	r1	t2	Yes
s ₈	w2	r2	t1	No
s ₉	w3	r2	t2	Yes
s ₁₀	w2	r2	t2	Yes
s ₁₁	w2	r1	t2	Yes
s ₁₂	w2	r1	t1	Yes
s ₁₃	w1	r2	t2	Yes
s ₁₄	w2	r1	t1	No

Table 5

Using the same principle, we can calculate the information gain to the remaining attributes and we have: $G(S, w) = 0.029$ and $G(S, t) = 0.94$.

As a result, we may choose the attribute of highest information gain to used to expand the tree. We consider the attribute traffic-jam as the root for the decision tree (Figure 12).

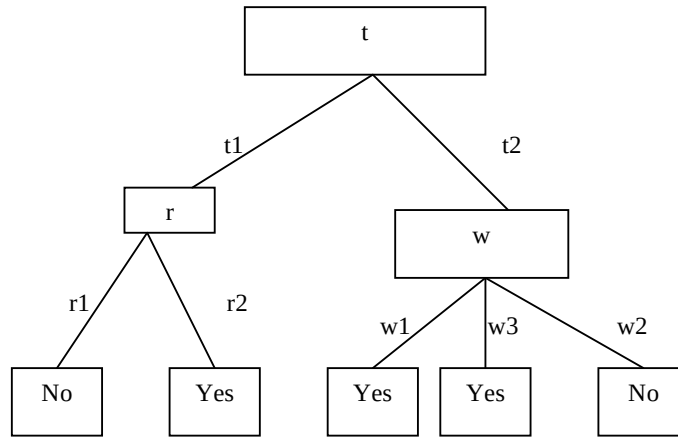


Figure 12

From the above figure as a result of using ID3 algorithm the following rules will be generated :

- If t is long and r is strong then people will choose to stay at home.
- If t is long and r is weak then people will choose to go to city patch. In logical form $(t, t1) \wedge (r, r2) \Rightarrow (go\ to\ city\ patch, No)$.
- If t is short and w is hot then people will choose to go to city patch. In logical form $(t, t2) \wedge (w, w1) \Rightarrow (go\ to\ city\ patch, Yes)$.
- If t is short and w is cool then people will choose to go to city patch. In logical form $(t, t2) \wedge (w, w3) \Rightarrow (go\ to\ city\ patch, Yes)$.

- If t is short and w is mild then people will choose to stay at home. In logical form $(t, t2) \wedge (w, w2) \Rightarrow (go\ to\ city\ patch, No)$.

In order to extend decision tree induction to a wider variety of problems, a number of issues must be addressed such as:

- Missing data: In many domains not all the attribute values will be known for every example. The values may not have been recorded, or they may be too expensive to obtain. First, given a complete decision tree, how should one classify an object that is missing one of the test attributes?
- Multi-valued attributes: When an attribute has a large number of possible values, the information gain measure gives an inappropriate indication of the attribute's usefulness.
- Multi-valued data: The multi-valued data is a generalization for a single valued one, when an attribute value has a number of values. How we can represent this problem by the decision tree and how the information gain measure in this case is an open problem.
- Continuous valued attributes: Attributes such as height and weight have a large or infinite set of possible values. They are therefore not well-suited for decision tree learning algorithms.

We introduce the following example which studies the above cases.

Example 5.2 In this example, a company needs some employees. It decided to accept the person if it has some languages, some scientific degrees, experience and background knowledge according to Table 6.

Person	Language	Degree	Experience	Background	Decision
P ₁	{F,D}	{Bs,Ms,PhD}	Medium	Excellent	Accept
P ₂	{H,R}	{Bs}	Low	Neutral	Reject
P ₃	{F,D,S}	{Bs, Ms}	Low	Good	Reject
P ₄	{F}	{Bs, Ms}	High	Neutral	Accept
P ₅	{D}	{Bs}	Medium	Neutral	Reject
P ₆	{F,D}	{Bs, Ms}	High	Excellent	Accept
P ₇	{H,R}	{Bs}	High	Good	Accept

Table 6

From the above table, $S = \{P_1, P_2, P_3, P_4, P_5, P_6, P_7\}$ is the set of examples. The set {Language, Degree, Experience, Background} is the condition attributes of the company and Decision is the decision attribute. The attribute Language has the value set {F,D, H, R, S} which is some languages in symbolic $V_{language} = \{F, D, H, R, S\}$. Also $V_{degree} = \{Bs, Ms, PhD\}$,

$V_{experience} = \{low, medium, high\}$ and $V_{background} = \{neutral, good, excellent\}$.

Now, we need to represent this problem by using DTLA. If we consider the attribute values {F,D} of the attribute Language as a separately attribute value we can apply DTLA easily but the result is not good where we take more than one attribute value in the same class.

The entropy for the set S of all persons (S contains 4 positive examples and 3 negative examples) is:

$$H(S) = -4/7 \log_2(4/7) - 3/7 \log_2(3/7) = 0.985$$

Then the information gain of each attribute is:

$$G(S, language) = H(S) - 2/7 H(\{F, D\}) - 2/7 H(\{H, R\}) - 1/7 H(\{F, D, S\}) - 1/7 H(\{F\}) - 1/7 H(\{D\}) = 0.985 - 0.2857 = 0.69928$$

$$G(S, degree) = H(S) - 1/7 H(\{Bs, Ms, PhD\})$$

$$- 3/7 H(\{Bs\}) - 3/7 H(\{Bs, Ms\}) = 0.985 - 3/7 [-2/3 \log_2(2/3) - 1/3 \log_2(1/3)] = 0.19789$$

$$G(S, experience) = H(S) - 2/7 H(medium) - 2/7 H(low) - 3/7 H(high) = 0.69928$$

$$G(S, background) = H(S) - 2/7 H(excellent) - 3/7 H(neutral) - 3/7 H(good) = 0.3.$$

MID3 Algorithm:

```

function MID3 (examples, attributes, default)
    returns a decision tree
inputs: examples, set of examples, attributes, set of attributes
    Default, default value for the goal predicate
if (examples is empty) then return default
else if (all examples have the same classification ) then return the classification.
else if (attributes is empty) then return MAJORITY-VALUE(examples)
else
do{
if (the attribute values have multi-valued)then
{ classify the attribute values into sub-classes}
best ← CHOOSE-ATTRIBUTE( attributes, examples)
tree ← a new decision tree with root test best
for each value  $v_i$  of best do
example  $s_i$  ← {elements of examples with best =  $v_i$  }
subtree ← MID3(examples  $s_i$  , attributes-best, MAJORITY-VALUE(examples) add a
    branch to tree with label  $v_i$  and subtree sub-tree} //end do
end
return tree

```

According to this algorithm Table 6 can then transformed into Table 7.

Person	Classes	Language	Degree	Experience	Background	Decision
P ₁	P ₁₁	F	Bs	Medium	Excellent	Accept
	P ₁₂	F	Ms	Medium	Excellent	Accept
	P ₁₃	F	PhD	Medium	Excellent	Accept
	P ₁₄	D	Bs	Medium	Excellent	Accept
	P ₁₅	D	Ms	Medium	Excellent	Accept
	P ₁₆	D	PhD	Medium	Excellent	Accept
P ₂	P ₂₁	H	Bs	Low	Neutral	Reject
	P ₂₂	R	Bs	Low	Neutral	Reject
P ₃	P ₃₁	F	Bs	Low	Good	Reject
	P ₃₂	F	Ms	Low	Good	Reject
	P ₃₃	D	Bs	Low	Good	Reject
	P ₃₄	D	Ms	Low	Good	Reject
	P ₃₅	S	Bs	Low	Good	Reject
	P ₃₆	S	Ms	Low	Good	Reject
P ₄	P ₄₁	F	Bs	High	Neutral	Accept
	P ₄₂	F	Ms	High	Neutral	Accept
P ₅	P ₅₁	D	Bs	Medium	Neutral	Reject
P ₆	P ₆₁	F	Bs	High	Excellent	Accept
	P ₆₂	F	Ms	High	Excellent	Accept
	P ₆₃	D	Bs	High	Excellent	Accept
	P ₆₄	D	Ms	High	Excellent	Accept
P ₇	P ₇₁	R	Bs	High	Good	Accept
	P ₇₂	H	Bs	High	Good	Accept

Table 7

By applying the Multi-valued ID3 (MID3) algorithm for the above table then the information

gain is given as: $G(S, \text{language}) = H(\text{persons}) - 9/23 H(F) - 8/23 H(D)$
 $- 2/23 H(H) - 2/23 H(R) - 2/23 H(S) = 0.985 - 0.299 - 0.331977 - 4/23 = 0.1801$

$G(S, \text{degree}) = H(\text{persons}) - 13/23 H(Bs) - 8/23 H(Ms)$
 $- 2/23 H(PhD) = 0.985 - 13/23 [-7/13 \log_2(7/13)] - 6/13 \log_2(6/13) = 0.09$

$$G(S, experience) = H(persons) - 7/23H(medium) - 8/23H(low) - 8/23H(high) = 0.805$$

$$G(S, background) = H(persons) - 10/23H(excellent) - 5/23H(neutral) - 8/23H(good) = 0.49$$

The attribute experience is of highest value of information gain. Then we take it as a root of our tree as shown in Figure (13).

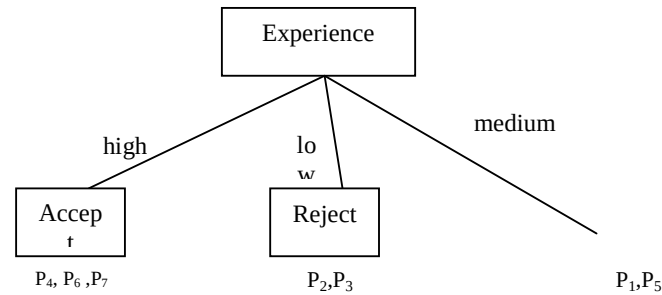


Figure 13

From the last decision tree we can deduce the following decision rules for the company example :

- The persons with high experience are accepted to work in the company.
- The persons with low experience are rejected to work in the company.
- The persons with medium experience and with excellent background are accepted to work in the company.
- The persons with medium experience and with neutral background are rejected to work in the company.

•

6. Conclusion and Future works

Decision trees are also a popular form of classification models. It is well known that classical trees lack the ability of modeling vagueness. By connecting fuzzy systems [38,39] and classical decision trees, we try to achieve classifiers that can model vagueness and are comprehensible. A popular and particularly efficient method for making a decision tree for classification from symbolic data is ID3 algorithm. Revised algorithms for numerical data have been proposed, some of which divide a numerical range into several intervals or fuzzy intervals. Their decision trees, however, are not easy to understand. In this work we proposed a new version of ID3 algorithm to generate a understandable fuzzy decision tree using fuzzy sets is fuzzy ID3.

In the near future we will generalize the approach of decision tree using topological rough sets [45,46] and topological fuzzy sets [44,47]. These approaches will discover new approximation class in information trees.

References

- [1] J. Atkinson-Abutridy, C. Mellish and S. Aitken, Combining information extraction with genetic algorithms for text mining. IEEE Intelligent System, vol. 19, No. 3, pp. 22-30. 2004.
- [2] W. H. Au and K. C. C Chan, An Effective Alogrithm for Discovering Fuzzy Rules in Relation Databases. In proceedings of the 7th IEEE International Conference on fuzzy system, pp. 1314-1319. 1998.
- [3] B. Babcock, S. Babu, M. Datar, R. Motwani and J. Widom. Models and Issues Data Stream Systems. In proceedings of the 2002 ACM Symposiumon Principles of Database System (PODS 2002) (Invited Paper). Acm Press, June 2002.
- [4] R. Berghammer, B. Leoniuk and V. Milanese, Implementation of relational algebra using binary decision diagrams, In de swart, H., editor, 6th International Conference Relmics 2001 and 1st workshop of cost Action 274 TARSKI, Oisterwijk, The Netherlands,

- October 16-21, Volume 2561 of lecture notes in computer science, pages 241-257, Berlin – Heidelberg – New York. Springer ,2002.
- [5] C. Bregler and S.M Omoundro. Non Linear Image Interpolation Using Manifold Learning. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), Advances in Neural Information Processing System 7, pp. 973-980. 1995.
 - [6] R. E. Bryant, Graph based algorithms for Boolean function manipulation, IEEE Trans on Computers, V.C 35, No. 8, pp. 677-691, 1986.
 - [7] W. Buntine, Learning Classification Trees, Statistics and Computing, 2:63-73, 1992.
 - [8] M. A. Carreira, Continuous latent variable models for dimensionality reduction and sequential data reconstruction. Phd thesis, University of Sheffield, UK. 2001.
 - [9] R. L. Chang P. T. Pavlidis, Fuzzy decision tree algorithms. IEEE Transactions on systems, Man, and cybernetics, vol. 7 (1), pp. 28-35, 1997.
 - [10] S. Chen, S.L. Lee and C. Lee, A new method for generating fuzzy rules from numerical data for handling classification problems. Applied Artificial Intelligence, vol. 15, No. 7, pp. 645-664. 2001.
 - [11] I-Jen Chiang, Jane Yung-jen Hsu. Fuzzy classification trees for data analysis. Fuzzy Sets and Systems , vol.130 ,pp.87 – 99. 2002.
 - [12] M. Dash, K. Choi, P. Scheuermann and H. Liu. Feature selection for clustering – A Filter Solution In Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 155-122. 2002.
 - [13] M. Dash and H. Liu. Feature Selection for classification. Intelligent Data Analysis, vol. 1, No 3, pp. 131-156. 1997.
 - [14] M. David, G. Squire, "The ID3 Decision tree Induction Algorithm", MONASH University Faculty of Information Technology. 2004.
 - [15] P. Devijver and J. Kittler. Pattern Recognition: A Statistical Approach. Prentice hall. 1982.
 - [16] A. Dobra, J. Gehrke, M. Garofalakis, and R. Rastogi. Processing Complex Aggregate Queries Over Data Streams. In proc. of the 2002 ACM SIGMOD Intel. Conf. On Management of Data, June 2002.
 - [17] U. M. Fayyad, and B. I. Keki, on the handling of continuous-Valued Attributes in Decision Tree Generation, Machine Learning, 8 87-102. 1992
 - [18] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From data mining to knowledge discovery in databases. Artificial Intelligence Magazine, vol. 17, No. 3, pp. 37-54. 1996.
 - [19] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. IEEE Transaction on Computers, vol. C-23, No. 9, pp. 881-890. 1974.
 - [20] J. H. Friedman and Wstuetzle. Projection pursuit regression. Journal of the American Statistics Association, vol. 76, pp. 817-823. 1981.
 - [21] Harish Garg , An approach for analyzing the reliability of industrial system using fuzzy Kolmogorov's differential equations, The Arabian Journal for Science and Engineering, 40(3), 975 – 987, 2015
 - [22] Harish Garg , Analyzing the behavior of an industrial system using fuzzy confidence interval based methodology, National Academy Science Letters, 37(4), 359 – 370, 2014
 - [23] Harish Garg , A novel approach for analyzing the behavior of industrial systems using weakest t-norm and intuitionistic fuzzy set theory, ISA Transaction, 53, 1199 – 1208, 2014
 - [24] Harish Garg ,Reliability, availability and maintainability analysis of industrial systems using PSO and fuzzy approach, MAPAN – Journal of Metrology Society of India, 29(2), 115 – 129, 2014
 - [25] Harish Garg, Preventive maintenance scheduling of pulping unit in a paper mill, Japan Journal of Industrial and Applied Mathematics, 30(2), 397 – 414, 2013
 - [26] Harish Garg, Performance analysis of complex repairable industrial systems using PSO and fuzzy confidence interval based lambda-tau methodology, ISA Transactions, 52(2), 171 – 183, 2013

- [27] Harish Garg, Performance and behavior analysis of repairable industrial systems using Vague Lambda-Tau methodology, *Applied Soft Computing*, 22, 323 – 338, 2014,
- [28] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, vol. 29, No. 1, pp. 1-27. 1964.
- [29] A. Muir, Lvo Dünstsch and Günther Gediga. Rough Set Data Representation Using Decision Diagram. *RACSAM, Rev. R. Acad. Cien. Serie A. Mat.*vol. 98 (1), pp. 197-211. 2004.
- [30] Y. Peng, P.A.Flach, Soft Discretization to enhance the continuous decision tree induction. *Integrating Aspects of Data Mining, Decision, Support and Metalearning*, pp. 109-118, 2001.
- [31] J. R. Quinlan, Induction of Decision Trees. *Machine learning*, vol. 1, pp. 81-106, 1986.
- [32] J. R. Quinlan, Decision trees and decision making, *IEEE transaction on system, Man and Cybernetic*, 20 - 339-346,1990.
- [33] J.R. Quinlan. C4.5: Programs for Machine Learning. The Morgan Kaufmann series in machine learning. Morgan kaufmann publishers, San Mateo, CA. 1993.
- [34] J. R. Quinlan, Improved use of continuous attributes in C4.5, *Journal of Artificial Intelligence Research*, 4- 77-90,1996.
- [35] B. Raman and T.R. Ioerger. Instance-Based Filter for Feature Selection. *Journal of Machine Learning Research*, vol. 1, pp. 1-23. 2002.
- [36] M. Sahinoglu and Security Meter. A Practical Decision Tree Model to Quantify Risk. *IEEE Security and privacy*, vol. 03 (3), pp. 18-24, May/June, 2005.
- [37] R. Slowinski, Intelligent decision support , handbook of applications and advances of rough set theory, Kluwer Academic Publishers, ISBN, 0-7923-1923-0, 1992.
- [38] L. A. Zadeh, Fuzzy Sets. *Information and Control*, vol. 8, pp. 338-353. 1965.
- [39] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning. *Information Sciences*, vol. 8, pp. 199-249, 301-357; vol 9, pp. 43-80. 1975.
- [40] W. Ziarko, The discovery, analysis and representation of data dependencies in databases, In: *Piatetsky G. – Shapiro, W, Frawley J. (Eds.), Knowledge Discovery in Databases*, AAAI Press / MIT press, pp. 177 – 195. 1991.
- [41] W. Ziarko, Variable Precision Rough Set Model. *Journal of Computer and System Sciences*, vol. 46, No. 1, pp. 39-59. 1993.
- [42] W. Ziarko, Rough sets and knowledge discovery, An overview. In: *Ziarko* , pp. 11 – 15. 1994.
- [43] W. Ziarko, Rough Sets, Fuzzy Sets and Knowledge Discovery (RSKD'93) Workshops in Computing, Springer–Verlag & British Computer Society, London, Berlin. (1994).
- [44] K. Y. Qin, Z. Pei, On the topological properties of fuzzy rough sets, *Fuzzy Sets and Systems* 151 (3) (2005) 601-613.
- [45] K. Y. Qin, J. L. Yang, Z. Pei, Generalized rough sets based on reflexive and transitive relations, *Information Sciences* 178 (2008) 4138-4141.
- [46] A. S. Salama, Topological solution of missing attribute values problem in incomplete information tables, *Information Sciences* 180 (2010) 631-639.
- [47] W. Z. Wu, A study on relationship between fuzzy rough approximation operator and fuzzy topological spaces, in: *L. Wang, Y. Jin (Eds.), FSKD 2005, LNAI*, vol. 3613, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 167-174.