DECISION TREE METHOD APPLIED IN ECONOMICS AND STATISTICS

Rocsana BUCEA-MANEA-ȚONIȘ, Radu BUCEA-MANEA-ȚONIȘ

„Spiru Haret“ University

REZUMAT. Arborii de decizie permit reprezentarea sub formă de diagramă a unor evenimente viitoare previzionate care condiționează decizia. Practic se reprezintă grafic toate combinațiile posibile ale variantelor decizionale şi ale stărilor sistemului în fiecare moment de timp. Deciziile sunt influențate de evenimente aleatoare a căror probabilitate poate fi anticipată. Reprezentarea vizuală a tuturor variantelor posibile facilitează luarea deciziilor, în timp real şi foarte uşor. În acest articol demonstrăm cum se implementează un arbore de decizie bazat pe algoritmul ID3. Deasemenea am implementat o simulare privind determinarea strategiei optime de distribuție a unei companii pe baza unui arbore de decizie.

Cuvinte cheie: entropie, câştig de informație, ID3, simulare.

ABSTRACT. Decision trees allow graphic representation of expected future events that condition the decision. In fact we plot all possible combinations of decision alternatives and states of the system each time. Decisions are influenced by random events whose probability can be predicted. The visual representation of all possible variants facilitate decision-making, in real time and very easily. In this paper we show how to implement a decision tree based on ID3 Algorithm. We also made a simulation in order to determine *the optimal distribution strategy* based on a decision tree.

Keywords: entropy, information gain, ID3, simulation.

1. INTRODUCTION

Decision trees are found to be excellent tools for making financial decisions related to numbers in cases where a large amount of complex information needs to be taken into account. They provide an effective structure in which alternative decisions and the implications of their choice can be assessed, and helps the formation of a balanced vision regarding risks and rewards arising from certain choices.

Decision trees provide unique capabilities to complement and substitute:

• traditional forms of statistical analysis (such as multiple linear regression);

• a variety of data mining tools and techniques (such as neural networks);

• newly developed forms of reporting and multi-dimensional analysis of the business intelligence.

Decision trees are based on algorithms that identify different ways of segmenting a data series into a tree. The tree consists of decision nodes, event/uncertainty node or end-nodes and system states branches and decision options branches. Any node of the tree, except the root node, have only one parent and one or more children. To calculate the values associated with each node the "rollback" procedure is used maximizing expected value, calculating the end-node values, then the values associated with intermediate nodes and initial ones respectively. The variant which corresponds to the largest expected profit or smallest loss possible with the highest expected value will be chosen.

The steps needed to create a decision tree are the following:

1. Describe the problem, possible events that influence the decision alternatives;

2. Decision nodes, variants that influence decision-making and their consequences events are represented;

3. Determine the consequences of each option;

4. Determine the possibilities of occurrence of events and how an event occurs.

Therefore the expected value and variance for each decision is calculated using formula:

$$S_m = \sum_{i=1}^{n} p_i R_i$$

where: S_m is the mathematical expectation; p_i – probability of events occurrence; R_i – the result of each variant.

The version with the best mathematical expectation – in case of positive values – is chosen. The accuracy of estimating the likelihood of events depends on making the right decision. The decision tree can be updated several times during the course of events depending on by the decision processes, the initial states of the system which may prove real or not.

2. IMPLEMENTATION OF ID3 DECISION TREE ALGORITHM

Most algorithms developed for decision trees successfully used in decision making are variations of the same algorithm involving a descending search space of possible decision trees. An example of this is the ID3 algorithm, developed by Australian Professor JR Quinlan, and the successor of this algorithm C4.5.

Principles of ID3 algorithm (Quinlan) are:

• apply to large data, each element having many attributes;

• recursively builds the tree from the initial node;

• selection criterion is maximum information gain, obtained by the difference between the entropies before and after the partitioning is the largest.

The set of standardized attributes include:

• numeric attributes - real numbers, *if* $n_i > V$;

• categorical attributes - belongs to a listed set *M*, *if* $n_i \in M$

• binary attributes — a threshold value distinguishes binary states, *if* $n_i = 1$

Attribute is identified by name and contains several values. Attributes are attached to decision tree nodes (each node corresponds to an attribute). From each node a sub-tree having nodes attached to its parent attribute values is generated. The tree is generated from the nodes with the lowest entropy – or with minimal uncertainty – and continuing with the variable that has less influence on the scope variable.

ID3 algorithm iteratively tests the information gain for each attribute. The root node will contain the attribute with the lowest entropy (maximum gain of information) – and the frequency of values that satisfy best the criterion. Branches are added for each possible value of the root node with the leaf nodes whose value satisfy the criterion of maximum information gain; then repeats the previous step until leaf attributes are finished.

Metrics:

$$\boldsymbol{E(S) = -\sum_{j=1}^{n} f_S(j) \log_2 f_S(j)}$$

Entropy of values sets, where: *S* is the the set of possible values; *n* – the number of possible values for the attribute considered; $f_S(j)$ – the frequency of value ***j*** in the set ***S*** of possible values.

$$\boldsymbol{E(T) = \sum_{i=1}^{m} f_S(A_i)\, E\left(S_{A_i}\right)}$$

2. Test entropy – where: $f_S(A_i)$ is the frequency of possible values for node *A* of the set *S*; A_i – any possible discrete value for node *A*; $E\left(S_{A_i}\right)$ – entropy calculated from the node *A*.

The values of an attribute that influences the autcome positive or negative (rates of positive / negative samples) from all the values an attribute may have, are weighted by the gain of information provided, after Shannon's law, ie, how much the uncertainty diminishes as determined by the ratio positive / negative values in all possible values when the scope variable is binary (eg. 0-1, winner - loser). The difference is the final gain information or entropy.

3. The gain of information, ***C (S, A) = E (S) – – E (SA)***.

Stages of decision tree creation:

1. Loading associated attribute values from the database in internal data structures list (ex. Class Attribute);

```
Attribute ca = new Attribute("ca", new
   string[] { "<2mil", "2-10mil", "10-
   50mil"});
Attribute pn = new Attribute("pn", new
   string[] { "are_profit", "fără_profit"
   });
Attribute sal = new Attribute("sal", new
   string[] { "1-9", "10-49", "50-249"});
Attribute[] attributes = new Attribute[] {
   ca, sal };
```

2. Calculate the entropy at each node based on the possible values - determine achievements with the highest frequency within the set of values for each attribute;

```
mTotal = DW.Rows.Count;
mTargetAttribute = targetAttribute;
mTotalPositives = countTotalPositives(DW);
mEntropySet = calcEntropy(mTotalPositives,
   mTotal - mTotalPositives);
```

3. Choose root node that provides the maximum information gain – it is determined root attribute which provides biggest gain information;

```
Attribute bestAttribute =
   getBestAttribute(DW, attributes);
TreeNode root = new
   TreeNode(bestAttribute);
```

4. Branches are built for each possible value of the root node, the leaves being filled with the remaining nodes in descending order of information gain;

```
if (aSample.Rows.Count == 0)
{
return new TreeNode(new
   Attribute(getMostCommonValue(aSample,
   targetAttribute)));
}
else
{
DecisionTreeID3 dc3 = new
   DecisionTreeID3();
TreeNode ChildNode = dc3.mountTree(aSample,
   targetAttribute,
```

```
   (Attribute[])aAttributes.ToArray(typeof
   (Attribute)));
root.AddTreeNode(ChildNode, value);
}
```

5. For each node algorithm resumes at step 4.

This way, we can build more decision trees from the set of tuples and the decision tree depth necessary to classify a lot of fields may vary depending on the order in which attributes are tested.

We consider each attribute (in our example: turnover, number of employees) as having a particular information contribution to the classification of that field.

```
SqlConnection conectare = new
   SqlConnection(constr);
SqlCommand cmd = new SqlCommand(sqlstr,
   conectare);
conectare.Open();
SqlDataReader dr = cmd.ExecuteReader();
DataTable result = new DataTable("DW");
DataColumn column =
   result.Columns.Add("ca");
column.DataType = typeof(string);
column = result.Columns.Add("sal");
column.DataType = typeof(string);
column = result.Columns.Add("result");
column.DataType = typeof(bool);
while (dr.Read())
{
Double d1 = Convert.ToDouble(dr["CA"]);
Double d2 = Convert.ToDouble(dr["NrSal"]);
Boolean b =
   ((Convert.ToDouble(dr["Re"])>0.1)&&((Co
   nvert.ToDouble(dr["Rf"]))>Convert.ToDou
   ble(dr["Re"]))) ? true : false;
result.Rows.Add(new object[] {
   Convert.ToDouble(dr["CA"]) < 2000000 ?
   "<2mil" : Convert.ToDouble(dr["CA"]) <
   10000000 ? "2-10mil" : "10-50mil",
   Convert.ToDouble(dr["NrSal"]) < 10 ?
   "1-9" : Convert.ToDouble(dr["NrSal"]) <
   50 ? "10-49" : "50-249", b });
}
```

The resulting tree stands for emphasizing the relation between turnover/number of employees and the financial performance for SMEs, as may be seen in Figure 1.

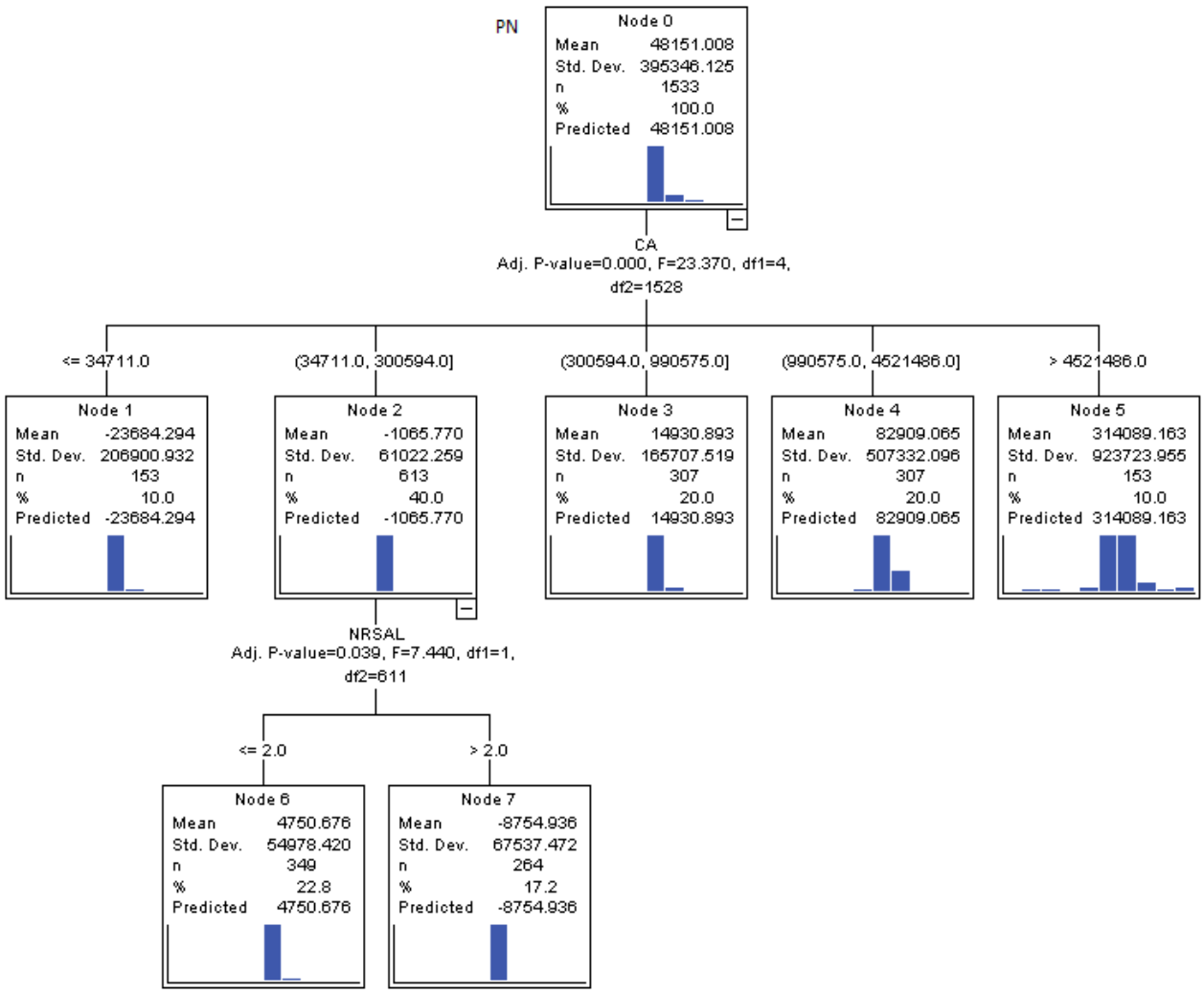

Fig. 1. Decision tree – attributes: turnover and number of employees.

3. SIMULATION DETERMINING THE OPTIMAL DISTRIBUTION STRATEGY

A firm wishes to market new IT gadgets on Romanian market. The company considers two ways of selling: selling through their own retail stores which will be located in the main cities or sale by CORA hypermarket network, which would generate a steady income of 378,000 EUR. However, top management of the company is considering paying a market survey, conducted by a leading consulting firm in Bucharest, whose price would be EUR 4,000 and would provide crucial data on the success of products on the market. Specialists have determined three scenarios for products, their associated probabilities and expected income in each case (Table 1).

Table 1. **Distribution Strategy - show possibilities for products**

Possible market situations	Probability of occurrence	Revenue estimated (thousand EUR)
SP1	0.6	898
SP2	0.25	578
SP3	0.15	415

SP1 – rapid acceptance WISEGADGET products on the market;
SP2 – average sales quantities WISEGADGET products on the market;
SP3 – poor sales WISEGADGET products under aggressive competition.

It identifies the following types of strategies available:

- prior market testing using the services of the consulting firm and then choosing either sale by CORA network either through their own retail stores;
- sale by CORA network without prior testing market, which brings secure income 378,000 EUR, commercialization efforts assuming their network CORA hypermarkets;
- selling through their own retail stores without a prior study of the market.

Study consulting firm determines the probability of 0.6 that the market launch of new products to be favorable and unfavorable 0.4 (that it is bringing the WISEGADGET products on Romanian market). It also provides top management's likelihood of achieving strategic alternatives simultaneously. (Table 2.)

Table 2. **Possibilities for simultaneous strategic alternatives**

Possible market situations	Favorable market	Unfavorable market	Absolute probabilities
SP1	0.4	0.15	**0.55**
SP2	0.14	0.14	**0.28**
SP3	0.06	0.11	**0.17**
Absolute probabilities	**0.6**	**0.4**	**1**

Based on the table above, we can calculate the probabilities of the three situations that may occur across the market, subject to a favorable or unfavorable market as follows:

$$P(SP1/p.f.) = 0.40 / 0.60 = 0.67$$
$$P(SP2/p.f.) = 0.14 / 0.60 = 0.23$$
$$P(SP3/p.f.) = 0.06 / 0.60 = 0.10$$

$$P(SP1/p.nef.) = 0.15 / 0.40 = 0.38$$
$$P(SP2/p.nef.) = 0.14 / 0.40 = 0.35$$
$$P(SP3/p.nef.) = 0.11 / 0.40 = 0.28$$

Furthermore there are three branches of the decision tree corresponding to the three strategic options for WISEGADGET product selling (Figure 2).

In 2, 4, 8 and 13 nodes we've calculated the **expected average income** indicator (VMA) where: P_i is the probability associated branch; V_{ij} – estimated income node *ij*.

$$VMA8 = 898 \times 0.67 + 578 \times 0.23\ 415 \times 0.1 = 775.03 \text{ (thousand EUR)}$$
$$VMA13 = 898 \times 0.38 + 578 \times 0.35\ 415 \times 0.28 = 653.2 \text{ (thousand EUR)}$$

Nodes 5 and 6 are picked up between the values of two adjacent branches representing strategic marketing options.

$$VMA2 = 653\ 175 \times 775\ 033 \times 0.6 + 0.4 = 726.29 \text{ (thousand EUR)}$$
$$VMA4 = 898 \times 0.6 + 578 \times 0.25\ 415 \times 0.15 = 745.55 \text{ (thousand EUR)}$$

Optimal strategy is testing the market prior to selling through their own retail stores generating an estimated average income of 775 000 EUR (VMA8), but must take into account the cost of consulting services 4,000 EUR. Selling by CORA network would provide a steady income of 378 000 EUR, much smaller than in the other two cases. Estimated income gap between the first two policy options is insignificant and, in the case of a market-oriented manager, customer will definitely choose the first strategic alternative (market testing and choosing the sales through his own stores) basing their decisions on the market study.

CONCLUSIONS

Decision trees represent a simple, but relevant method for the analysis of multiple variables that apply to very complex decision situations, involving successive random events.

Our case study is relevant for decision tree implementation in the economic field.

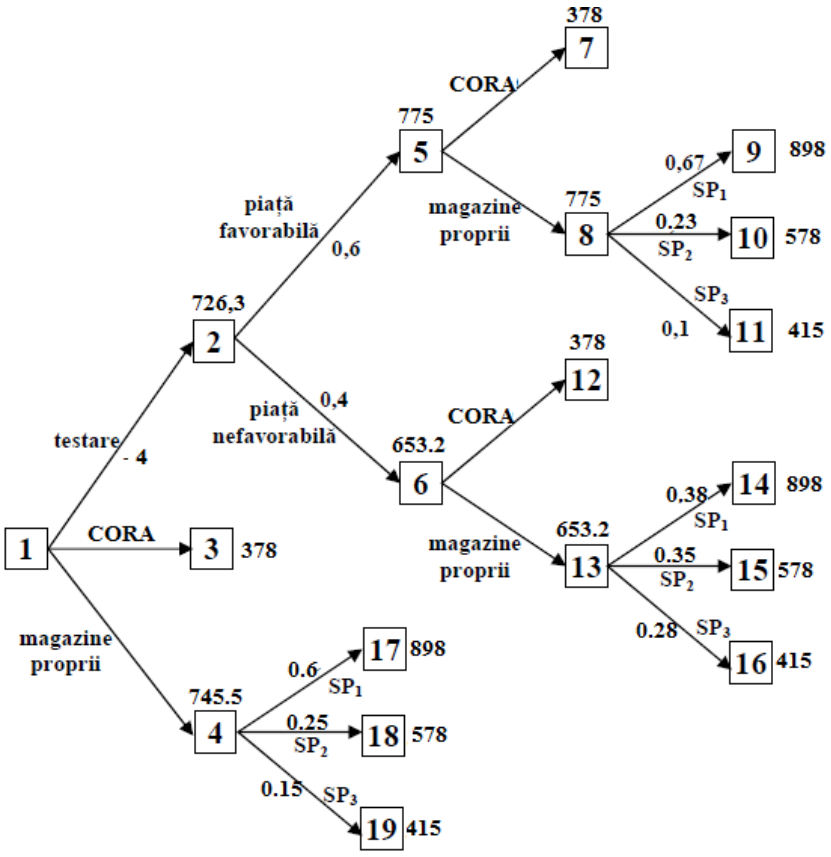

Fig. 2. The decision tree corresponding to the three proposed strategies.

BIBLIOGARPHY

[1] Golovin, D., Krause, A.: *Adaptive Submodularity: A New Approach to Active Learning and Stochastic Optimization* (2010), http://arxiv.org/abs/1003.3967

[2] Chakaravarthy, V., Pandit, V., Roy, S., Sabharwal, Y.: *Approximating Decision Trees with Multiway Branches.* In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009. LNCS, vol. 5555, pp. 210–221. Springer, Heidelberg (2009)

[3] Adler, M., Heeringa, B.: *Approximating Optimal Binary Decision Trees*. In: Goel, A., Jansen, K., Rolim, J.D.P., Rubinfeld, R. (eds.) APPROX and RANDOM 2008. LNCS, vol. 5171, pp. 1–9. Springer, Heidelberg (2008)

[4] Chakravarthy, V.T., Pandit, V., Roy, S., Awasthi, P., Mohania, M.: *Decision Trees for Entity Identification: Approximation Algorithms and Hardness Results.* In: PODS (2007)

[5] ID3 Decision Tree Algorithm in C# - http://www.codeproject.com/Articles/5276/ID3-Decision-Tree-Algorithm-in-C

[6] Algoritm ID3 - http://en.wikipedia.org/wiki/ID3_algorithm

[7] Depatment of Computer Science –San Jose State University College of Science - http://www.sjsu.edu/cs/

[8] Strategies in Decision Trees - http://www.treeplan.com/chapters/strategies-in-decision-trees.pdf